

# VISUALISASI INFORMASI DATA ANALISIS PERINGKAT ANIME DAN SISTEM REKOMENDASI MEGGUNAKAN PEMROGRAMAN PYTHON

*Diterima Redaksi: 12 Juli 2025; Revisi Akhir: 20 Oktober 2025; Diterbitkan Online: 30 November 2025*

**Clarissa Dwi Apriani<sup>1)</sup>, Rafaisya Dwi Adrianto<sup>2)</sup>**

<sup>1)</sup> Sistem Informasi, Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang

<sup>2)</sup> Informatika, Fakultas Ilmu Komputer Universitas Singaperbangsa Karawang

<sup>1, 2)</sup> Jl. H.S. Ronggowaluyo, Puseurjaya, Telukjambe Timur, Karawang, Jawa Barat, Indonesia, kode pos: 41361  
e-mail: 2110631250004@student.unsika.ac.id <sup>1)</sup>, 2310631170147@student.unsika.ac.id<sup>2)</sup>

**Abstrak:** Jumlah anime yang terus bertambah dari tahun ke tahun membuat penonton sering kesulitan memilih tontonan yang sesuai dengan minat mereka. Hal ini menimbulkan kebutuhan akan sistem rekomendasi yang mampu menyajikan konten relevan secara otomatis. Penelitian ini bertujuan untuk menggabungkan visualisasi data dengan sistem rekomendasi dalam menganalisis peringkat anime sehingga dapat membantu pengguna menemukan anime sesuai preferensi. Metode penelitian meliputi analisis eksploratif dataset, preprocessing data, visualisasi informasi, serta pengembangan dua model sistem rekomendasi dengan pendekatan Collaborative Filtering dan Content-Based Filtering. Implementasi dilakukan menggunakan bahasa pemrograman Python melalui platform Google Colaboratory. Hasil penelitian menunjukkan bahwa visualisasi data mampu membantu pengguna memahami informasi kompleks terkait rating, genre, dan popularitas anime. Sementara itu, sistem rekomendasi yang dibangun menghasilkan daftar rekomendasi yang akurat dan sesuai preferensi pengguna. Dengan demikian, penggabungan visualisasi data dan sistem rekomendasi ini dapat memberikan nilai tambah bagi industri hiburan sekaligus meningkatkan pengalaman menonton anime secara keseluruhan.

**Kata Kunci—** Visualisasi Data, Sistem Rekomendasi, Anime, Python, Google Colab

**Abstract:** The growing number of anime each year makes it difficult for viewers to find shows that match their preferences, creating the need for recommendation systems that can automatically provide relevant content. This study aims to integrate data visualization with recommendation systems in anime rating analysis to help users discover anime aligned with their interests. The methodology includes exploratory data analysis, data preprocessing, data visualization, and the development of two recommendation system models using Collaborative Filtering and Content-Based Filtering approaches. The implementation was carried out using Python programming on the Google Colaboratory platform. The results indicate that data visualization helps users understand complex information about ratings, genres, and anime popularity, while the developed recommendation system generates accurate and personalized recommendations. Therefore, combining data visualization and recommendation systems can provide added value to the entertainment industry and enhance the overall anime viewing experience.

**Keywords—** Data Visualization, Recommendation Systems, Anime Rating Analysis, Python, Google Colab

## I. PENDAHULUAN

D i tengah era digital yang sarat dengan arus informasi, kemampuan dalam mengolah data telah menjadi keterampilan yang esensial. Hal ini juga berlaku pada sektor hiburan, khususnya dalam dunia anime, di mana para penikmatnya terus mencari tayangan yang sesuai dengan preferensi pribadi mereka. Untuk menjawab kebutuhan tersebut, visualisasi data serta sistem rekomendasi hadir sebagai solusi efektif dalam menyampaikan informasi secara lebih intuitif serta membantu pengguna menemukan konten yang paling relevan dengan minat mereka.

Penelitian ini difokuskan pada penggabungan dua komponen krusial dalam analisis data anime, yaitu visualisasi informasi dan sistem rekomendasi. Melalui pemanfaatan bahasa pemrograman Python sebagai perangkat utama, penulis merancang sebuah sistem yang dapat mengevaluasi peringkat anime serta menghasilkan rekomendasi yang disesuaikan dengan preferensi masing-masing pengguna.

Urgensi penelitian ini terletak pada meningkatnya jumlah konten anime yang tersedia secara digital, sehingga pengguna sering kesulitan menemukan tontonan yang sesuai dengan preferensinya. Dengan

adanya sistem rekomendasi yang akurat, pengalaman pengguna dalam menonton dapat ditingkatkan, sekaligus membantu industri hiburan memahami pola preferensi audiens untuk menekan tingkat churn dan meningkatkan kepuasan pelanggan.

Visualisasi data adalah konsep senidansains. Ini adalah istilah umum yang menggambarkan setiap upaya untuk membantu orang memahami pentingnya data dengan memasukkannya ke dalam konteks visual. Nilai hemat data telah berubah, dan penggunaan data memungkinkan organisasi membuat keputusan yang lebih cerdas, lebih dekat ke "real-time" [1]. Pada penelitiannya, Tupari et al. mengatakan hal yang sama, bahwa informasi yang kompleks dan beragam dari analisis sentimen dapat digambarkan dalam bentuk visual seperti grafik atau diagram, sehingga orang yang terlibat dapat lebih mudah memahami makna sentimen bahkan jika mereka tidak memiliki pengetahuan teknis [2].

Visualisasi data bisa dibuat dengan berbagai tools seperti Google Collab, Power Bi, Tableau, Google Data Studio/Google Looker Studio, dan lain-lain [3]. Penelitian ini akan membahas penggunaan Google Colab, layanan komputasi awan yang diberikan oleh Google untuk mendukung pengembangan dan penelitian ilmiah, untuk memvisualisasikan informasi data analisis peringkat anime dan sistem rekomendasi [4]. Colaboratory, atau "Colab", adalah produk dari Google Research yang memungkinkan pengguna menulis dan mengeksekusi kode Python secara otomatis melalui browser. Produk ini sangat ideal untuk pendidikan, analisis data, dan machine learning. Secara teknis, Colab adalah layanan notebook Jupyter yang dihosting yang dapat digunakan tanpa penyiapan dan memungkinkan akses gratis ke sumber daya komputasi, termasuk GPU [5]. Selain itu, Google colab dapat digunakan secara bersama-sama oleh pengembang aplikasi, yang mendukung kebutuhan kerja tim. Namun, Google Colab masih sangat asing bagi orang biasa karena hanya digunakan oleh orang yang membutuhkan lingkungan coding, seperti developer atau programmer [6].

Anime merupakan bentuk animasi yang berasal dari Jepang, dikenal dengan visual yang penuh warna dan karakter unik di berbagai latar cerita. Gaya visual anime sangat dipengaruhi oleh manga, yaitu komik khas Jepang. Banyak anime yang diadaptasi dari manga terkenal seperti Naruto, novel seperti Another, ataupun dari light novel seperti Overlord. Tidak sedikit pula yang merupakan karya orisinal studio animasi seperti Charlotte dan Angel Beats. Anime tidak hanya digemari di Jepang, tetapi juga telah mendapat sambutan luas di berbagai negara, termasuk Indonesia [7]. Ciri khas anime meliputi desain karakter dengan mata besar, dagu runcing, gaya rambut yang beragam, serta tampilan karakter yang menarik. Berbeda dengan animasi Barat, anime memiliki tema yang lebih luas, mulai dari romansa, kehidupan sehari-hari, petualangan, hingga unsur magis, kekuatan supranatural, misteri, dan aksi [8].

Dalam penelitian ini, penulis memanfaatkan bahasa pemrograman Python untuk melakukan visualisasi data dan membangun sistem rekomendasi berdasarkan dataset yang tersedia. Python merupakan bahasa pemrograman tingkat tinggi yang dikenal luas karena sintaksisnya yang ringkas, mudah dipahami, serta bersifat fleksibel. Bahasa ini dikembangkan oleh Guido van Rossum dan telah digunakan secara luas dalam berbagai bidang seperti pengembangan software, analisis data, kecerdasan buatan, pengembangan aplikasi web, serta berbagai aplikasi lainnya. Popularitas Python di kalangan pengembang tidak lepas dari kemampuannya menggabungkan kemudahan penulisan kode dengan fungsionalitas yang luas untuk menangani berbagai kebutuhan pemrograman [9].

## II. TINJAUAN PUSTAKA

### A. Visualisasi Data

Visualisasi data adalah teknik untuk menyampaikan informasi dalam bentuk grafis seperti chart atau diagram, dengan tujuan mempermudah interpretasi terhadap data yang kompleks. Visualisasi dapat mempermudah penyampaian hasil analisis sentimen, sehingga informasi tersebut dapat dipahami oleh audiens yang tidak memiliki latar belakang teknis [10]. Selain itu, visualisasi juga berperan penting dalam mendukung pengambilan keputusan berbasis data secara lebih efektif.

### B. Sistem Rekomendasi

Sistem rekomendasi adalah teknologi yang berfungsi untuk menyarankan konten kepada pengguna berdasarkan minat atau pola interaksi mereka. Terdapat dua pendekatan utama yang umum diterapkan, yaitu *Content-Based Filtering* dan *Collaborative Filtering*. Pendekatan *Content-Based*

mengevaluasi fitur dari item dan kesesuaian dengan preferensi pengguna, sedangkan *Collaborative Filtering* mengandalkan data interaksi antar pengguna. *Content-Based Filtering* mampu meningkatkan kesesuaian konten pada linimasa media sosial dengan mengandalkan preferensi pengguna dan pemanfaatan bobot TF-IDF dalam mengukur kesamaan antar konten [11].

### C. Bahasa Pemrograman Python

Python adalah bahasa pemrograman tingkat tinggi yang banyak digunakan untuk analisis data, visualisasi, serta pengembangan sistem berbasis *machine learning* karena sintaksnya yang sederhana dan fleksibel [12]. Dalam penelitian ini, Python dimanfaatkan untuk berbagai tahap, yaitu *preprocessing* data menggunakan pustaka Pandas, visualisasi data dengan Matplotlib, Seaborn, dan Plotly, serta pembangunan model rekomendasi menggunakan Scikit-learn. Fleksibilitas Python memungkinkan penggabungan berbagai metode analisis dalam satu lingkungan pemrograman yang terintegrasi.

### D. Google Colaboratory

*Google Colaboratory*, sebagai *platform notebook* berbasis *cloud* dari *Google*, menyediakan lingkungan yang ideal untuk menjalankan kode Python secara interaktif, termasuk dalam penerapan analisis data dan sistem rekomendasi [13].

### E. Algoritma K-Nearest Neighbors (KNN)

KNN merupakan salah satu algoritma *machine learning* yang bekerja dengan prinsip mencari kedekatan antar data berdasarkan metrik jarak, salah satunya *cosine similarity*. Dalam sistem rekomendasi, KNN sering digunakan pada pendekatan *collaborative filtering* untuk mencari item yang paling mirip dengan preferensi pengguna. KNN mampu menghasilkan rekomendasi yang relevan pada sistem rekomendasi anime dengan memanfaatkan data *rating* pengguna [14].

### F. Analisis Eksploratif Data

Analisis eksploratif data merupakan tahap awal yang bertujuan memahami karakteristik data sebelum dilakukan pemodelan lebih lanjut. Analisis eksploratif membantu peneliti menemukan pola, distribusi, dan anomali pada dataset [15]. Dalam konteks penelitian ini, analisis eksploratif dilakukan untuk mengetahui distribusi *rating* anime, popularitas berdasarkan jumlah anggota, serta perbedaan kategori tipe anime (TV, Movie, OVA, ONA, Special, Music). Hasil dari analisis ini kemudian dijadikan dasar untuk visualisasi data dan pembangunan sistem rekomendasi.

## III. METODE PENELITIAN

Tahapan metodologi pada penelitian ini terdiri dari analisis eksploratif, pengolahan data, visualisasi data, dan membuat sistem rekomendasi.

### A. Rancangan Penelitian

Penelitian ini bertujuan untuk menganalisis peringkat anime dan membangun sistem rekomendasi berbasis konten dan kolaboratif menggunakan Python di *Google Colaboratory*. Dataset yang digunakan dalam penelitian ini bersumber dari situs MyAnimeList (<https://myanimelist.net/>). Dataset ini berisi daftar anime dengan informasi seperti judul, genre, tipe, jumlah episode, rating, dan jumlah pengguna. Total data yang digunakan terdiri dari 12.293 baris dengan 7 atribut utama, yaitu *anime\_id*, *name*, *genre*, *type*, *episodes*, *rating*, dan *members*. Tahapan yang dilakukan terdiri dari: analisis eksploratif terhadap dataset, *preprocessing* data, visualisasi informasi, serta pengembangan dua model sistem rekomendasi. Tujuan utama dari rancangan ini adalah untuk membantu pengguna menemukan konten anime yang sesuai dengan preferensinya melalui penyajian visual dan rekomendasi otomatis.

### B. Arsitektur Sistem dan Tools

Arsitektur sistem dalam penelitian ini mencakup beberapa tahap utama, yaitu:

1. Akuisisi data
2. Preprocessing dan eksplorasi data
3. Visualisasi data menggunakan grafik interaktif
4. Pembuatan sistem rekomendasi

## 5. Evaluasi dan interpretasi hasil

Implementasi dilakukan menggunakan *Google Colaboratory* yang memungkinkan penulisan dan eksekusi kode Python secara *cloud-based*. Pustaka yang digunakan dalam pengolahan dan visualisasi data meliputi:

1. Pandas untuk manipulasi data
2. Matplotlib, Seaborn, dan Plotly untuk visualisasi grafik
3. Scikit-learn untuk implementasi model *Collaborative Filtering* dan *Content-Based Filtering*

## C. Teknik Pengumpulan dan Pengolahan Data

Data diperoleh dari situs MyAnimeList (<https://myanimelist.net/>) yang berisi daftar anime, *rating*, genre, dan jumlah pengguna. Dataset dikumpulkan dalam format CSV dengan 12.293 baris dan 7 kolom utama: *anime\_id*, *name*, *genre*, *type*, *episodes*, *rating*, dan *members*.

### 1) Preprocessing Data

Beberapa tahapan preprocessing dilakukan untuk membersihkan dan menyiapkan data, meliputi:

- a) Mengganti nilai *user\_rating* = -1 dengan NaN, lalu menghapusnya
- b) Memilih pengguna yang memberikan  $\geq 50$  *rating*
- c) Membuat pivot table dengan *user\_id* sebagai kolom dan *name* sebagai baris
- d) Membersihkan nama anime dari karakter simbol menggunakan regex

### 2) Transformasi Data

Data teks pada genre dikonversi menjadi vektor numerik menggunakan metode TF-IDF (Term Frequency–Inverse Document Frequency). Representasi ini mempermudah model dalam memahami karakteristik dari setiap anime.

## D. Implementasi Sistem Rekomendasi

### 1) Collaborative Filtering

Model ini dibangun menggunakan algoritma K-Nearest Neighbors (KNN) dengan metrik jarak cosine similarity. Dataset yang telah dibersihkan dikonversi menjadi sparse matrix dan digunakan sebagai input pelatihan model. Model kemudian mencari tetangga terdekat dari anime input untuk menghasilkan daftar rekomendasi berdasarkan preferensi pengguna lain.

### 2) Content-Based Filtering

Metode ini menggunakan fitur teks dari genre anime yang telah ditransformasikan dengan TF-IDF. Kemiripan antar anime dihitung menggunakan sigmoid kernel. Sistem akan merekomendasikan anime dengan karakteristik yang paling mirip dengan anime yang diinput oleh pengguna.

## E. Teknik Analisis Data

Analisis dilakukan melalui proses visualisasi dan evaluasi model:

- a) Visualisasi Data: Menampilkan distribusi *rating*, genre, dan popularitas anime menggunakan grafik batang, pie chart, dan histogram.
- b) Evaluasi Sistem Rekomendasi: Menggunakan pendekatan deskriptif dengan meninjau kualitas hasil rekomendasi, relevansi konten, serta keakuratan model berdasarkan uji coba pada data input nyata.

Visualisasi dan hasil rekomendasi ditampilkan dalam bentuk grafik untuk membantu interpretasi hasil oleh pengguna.

## IV. HASIL DAN PEMBAHASAN

Dataset yang telah diperoleh kemudian disimpan dalam format csv (*comma delimited*) yang terdiri dari 7 kolom dan 12.293 baris, dengan penjelasan atribut seperti pada tabel 1.

Tabel 1 Atribut Tabel Dataset Peringkat Anime

No.	Kolom	Keterangan
1	<i>anime_id</i>	Kode anime
2	<i>name</i>	Judul anime
3	<i>genre</i>	Genre anime

No.	Kolom	Keterangan
4	type	Tipe anime
5	episodes	Jumlah episode anime
6	rating	Peringkat anime
7	members	Jumlah penonton anime

Langkah berikutnya setelah mengumpulkan dataset adalah menghubungkannya ke dalam dataframe Python untuk memungkinkan proses analisis dan visualisasi data.

Tabel 2 Potongan Isi Tabel Dataset Rating

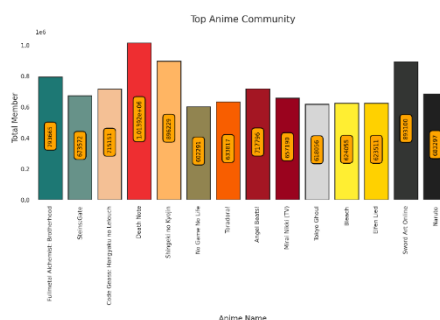
anime id	name	genre	type	episode	rating	members	user id	user rating
32281	Kimi no Nawa	Drama, Romance, School, Supernatural	Movie	1	9.370000	200630	72404	10
32281	Kimi no Nawa	Drama, Romance, School, Supernatural	Movie	1	9.370000	200630	16698	10
32281	Kimi no Nawa	Drama, Romance, School, Supernatural	Movie	1	9.370000	200630	16698	10
32281	Kimi no Nawa	Drama, Romance, School, Supernatural	Movie	1	9.370000	200630	44890	10
32281	Kimi no Nawa	Drama, Romance, School, Supernatural	Movie	1	9.370000	200630	19964	10

## A. Visualisasi Data

Data akan divisualisasikan berupa grafik, dimana data distribusi grafik ini terbagi menjadi beberapa *Type*.

### 1) Anime dengan jumlah penonton terbanyak

Hasil visualisasi anime terpopuler dapat dilihat pada Gambar 1. Anime Death Note menempati posisi teratas dengan 1.013.920 penonton, diikuti oleh Shingeki no Kyojin (896.229 penonton), dan Sword Art Online (893.100 penonton).



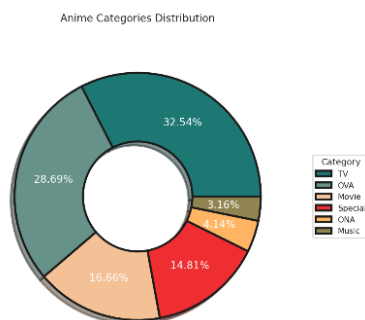
Gambar 1 Grafik Anime dengan jumlah penonton terbanyak

```
import matplotlib.pyplot as plt
top_anime = df.sort_values('members', ascending=False).head(10)
plt.figure(figsize=(10,6))
plt.bar(top_anime['name'], top_anime['members'])
plt.xticks(rotation=45)
plt.title('Top 10 Anime Berdasarkan Jumlah Penonton')
plt.show()
```

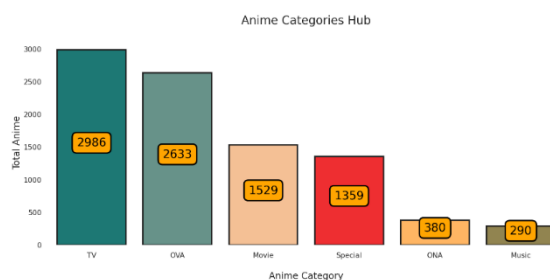
Gambar 2 Potongan Kode Grafik Anime dengan Jumlah Penonton Terbanyak

Kode menggunakan *library* matplotlib.pyplot untuk membuat grafik batang.

## 2) Distribusi Anime Berdasarkan Type



**Gambar 3** Grafik Persentase Distribusi Anime Berdasarkan Type



**Gambar 4** Grafik Angka Distribusi Anime Berdasarkan Type

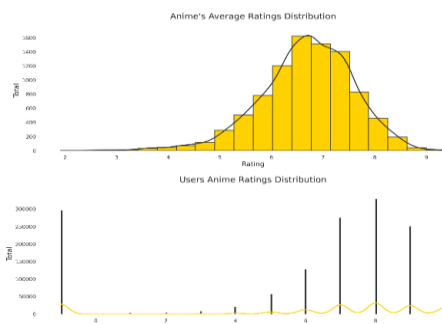
Dari data yang didapat, terlihat 3402 anime ditayangkan di TV yang merupakan 30,48% dari total anime, 2111 anime ditayangkan dalam bentuk film yang merupakan 18,91% dari total anime, 3090 anime ditayangkan dalam bentuk OVA yang merupakan 27,69% dari total anime, juga lebih besar dari OVA yang mencakup 526 anime yang merupakan 4,71% dari total anime.

```
import seaborn as sns
plt.figure(figsize=(8,6))
sns.countplot(x='type', data=df, order=df['type'].value_counts().index)
plt.title('Distribusi Anime Berdasarkan Type')
plt.show()
```

**Gambar 5** Potongan Kode Grafik Distribusi Anime Berdasarkan Type

Kode menggunakan *library* seaborn.countplot untuk membuat visualisasi distribusi jumlah anime berdasarkan type (TV, OVA, Movie, dll) dan matplotlib.pyplot untuk menampilkan grafik.

## 3) Peringkat Anime Secara Keseluruhan



**Gambar 6** Grafik Peringkat Anime Secara Keseluruhan

Sebagian besar peringkat Anime tersebar antara 5.5 - 8.0 dan sebagian besar peringkat pengguna tersebar antara 6.0 - 10.0. Modus distribusi peringkat pengguna sekitar 7.0 - 8.0, terlihat kedua distribusi tersebut condong ke arah kiri.



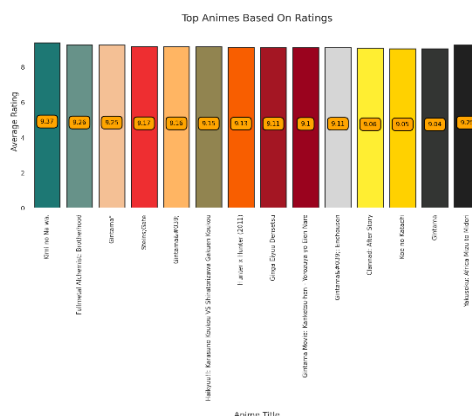
Hasil visualisasi menunjukkan bahwa sebagian besar anime memperoleh *rating* di kisaran 7.0–8.0, yang menunjukkan adanya kecenderungan pengguna untuk memberi nilai moderat hingga tinggi. Temuan ini sesuai dengan studi oleh Putri et al. (2022), yang menyatakan bahwa pengguna cenderung memberikan *rating* tinggi untuk genre yang populer seperti aksi atau fantasi.

```
plt.figure(figsize=(8,6))
sns.histplot(df['rating'].dropna(), bins=20, kde=True)
plt.title('Distribusi Rating Anime')
plt.show()
```

Gambar 7 Potongan Kode Grafik Peringkat Anime Secara Kseluruhan

Kode menggunakan *library* seaborn.histplot untuk menampilkan distribusi peringkat anime.

#### 4) Anime Teratas Berdasarkan Peringkat



Gambar 8 Grafik Anime Teratas Berdasarkan Peringkat

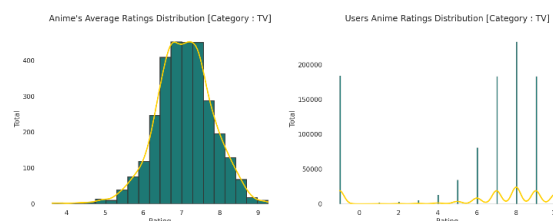
Berdasarkan data yang didapat, terlihat bahwa anime Mogura no Moto no Muroto menduduki peringkat tertinggi, diikuti dengan anime Kimi no Nawa, dan Fullmetal Alchemist: Brotherhood.

```
top_rating = df.sort_values('rating', ascending=False).head(10)
plt.figure(figsize=(10,6))
plt.bar(top_rating['name'], top_rating['rating'])
plt.xticks(rotation=45)
plt.title('10 Anime dengan Peringkat Tertinggi')
plt.show()
```

Gambar 9 Potongan Kode Grafik Anime Teratas Berdasarkan Peringkat

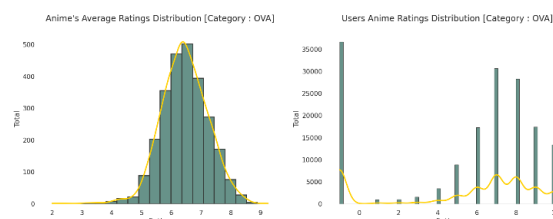
Kode menggunakan *library* pandas.sort\_values untuk memilih anime dengan *rating* tertinggi dan matplotlib.pyplot untuk membuat grafik batang.

#### 5) Distribusi Peringkat Anime Berdasarkan Type



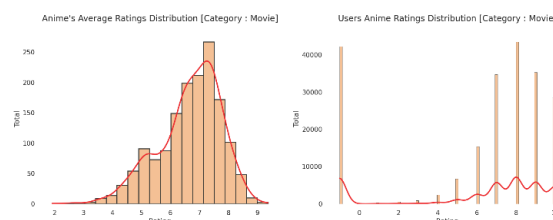
Gambar 10 Grafik Distribusi Peringkat Anime Berdasarkan Type Tv

Sebagian besar peringkat Anime tersebar antara 6.0 - 8.0 dan sebagian besar peringkat pengguna tersebar antara 6.0 - 10.0 Modus dari distribusi peringkat pengguna adalah sekitar 7.0 - 9.0, terlihat kedua distribusi tersebut condong ke arah kiri.



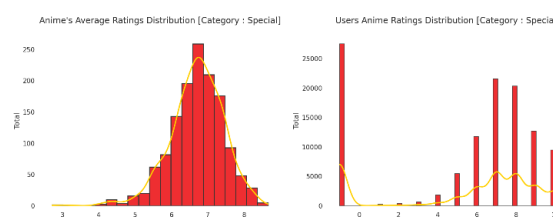
*Gambar 11 Grafik Distribusi Peringkat Anime Berdasarkan Type OVA*

Sebagian besar peringkat Anime berkisar antara 5.5 dan 7.5, dan sebagian besar peringkat pengguna berkisar antara 5.5 dan 10.0. Modus distribusi peringkat pengguna sekitar 7.0 hingga 8.0, yang menunjukkan bahwa kedua distribusi tersebut cenderung mengarah ke kiri.



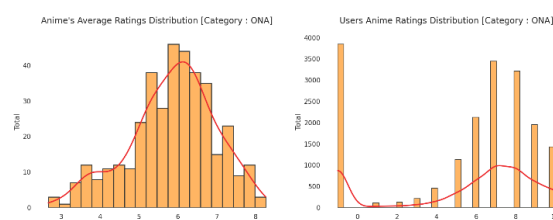
*Gambar 12 Grafik Distribusi Peringkat Anime Berdasarkan Type Movie*

Sebagian besar peringkat Anime tersebar antara 4.5 - 8.5 dan tersebar antara 5.0 - 10.0 pada sebagian besar peringkat pengguna. Modus dari distribusi peringkat pengguna adalah sekitar 7.0 - 9.0, terlihat kedua distribusi tersebut condong ke arah kiri.



*Gambar 13 Grafik Distribusi Peringkat Anime Berdasarkan Type Special*

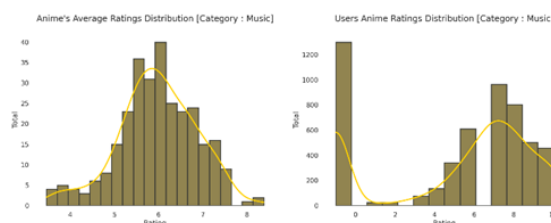
Sebagian besar peringkat Anime tersebar antara 5.5 - 8.0 dan sebagian besar peringkat pengguna tersebar antara 5.0 - 10.0. Modus distribusi peringkat pengguna sekitar 7.0 - 8.0 yang menunjukkan bahwa kedua distribusi tersebut condong ke arah kiri.



*Gambar 14 Grafik Distribusi Peringkat Anime Berdasarkan Type ONA*

Sebagian besar peringkat Anime berkisar antara 4.0 dan 7.0, dan sebagian besar peringkat pengguna berkisar antara 5.0 dan 10.0. Modus distribusi peringkat pengguna sekitar 7.0–8.0, dan kedua distribusi ini cenderung mengarah ke kiri.





Gambar 15 Grafik Distribusi Peringkat Anime Berdasarkan Type Musik

Sebagian besar peringkat Anime tersebar antara 4.0 - 7.5 dan tersebar antara 5.0 - 10.0 pada sebagian besar peringkat pengguna. Modus distribusi peringkat pengguna sekitar 6.5 hingga 8.0, dan kedua distribusi tersebut cenderung mengarah ke kiri.

```
plt.figure(figsize=(8,6))
sns.boxplot(x='type', y='rating', data=df)
plt.title('Distribusi Rating Berdasarkan Type Anime')
plt.show()
```

Gambar 16 Potongan Kode Grafik Distribusi Peringkat Berdasarkan Type

Kode menggunakan *library* seaborn.boxplot untuk menampilkan distribusi rating tiap kategori anime dan matplotlib.pyplot untuk menampilkan hasil visualisasi.

## B. Sistem Rekomendasi

### 1) Final Data Preprocessing

Dalam penelitian ini, penulis melakukan serangkaian prapemrosesan data dan analisis untuk menghasilkan sistem rekomendasi anime yang akurat dan bermakna. Berikut adalah deskripsi singkat tentang proses yang penulis lakukan.

#### a) Pengolahan Data Awal

Pertama-tama, penulis menyalin dataset asli ke variabel 'data' untuk memastikan keberlanjutan data asli. Penulis mengganti nilai -1 dalam kolom "user\_rating" dengan nilai NaN (Not a Number) menggunakan fungsi 'replace()'. Kemudian, Penulis menghapus baris yang mengandung nilai NaN menggunakan 'dropna()'. Langkah ini membantu Penulis membuang entri yang tidak memiliki peringkat pengguna yang valid.

```
# Mengganti nilai -1 dengan NaN pada kolom user_rating
df['user_rating'] = df['user_rating'].replace(-1, np.nan)

# Menghapus baris yang mengandung NaN
df = df.dropna(subset=['user_rating'])
```

Gambar 17 Potongan Kode Python Untuk Pengolahan Data Awal

#### b) Seleksi Pengguna

Penulis melanjutkan dengan memilih pengguna yang memiliki partisipasi yang signifikan dalam dataset. Dengan menghitung jumlah penilaian yang diberikan oleh setiap pengguna, Penulis mempertahankan hanya pengguna yang memberikan minimal 50 penilaian. Ini dilakukan untuk memastikan bahwa Penulis memiliki data yang memadai untuk menganalisis preferensi pengguna dan membuat rekomendasi yang relevan.

```
# Hitung jumlah rating per pengguna
user_counts = df['user_id'].value_counts()

# Pilih hanya pengguna dengan minimal 50 rating
valid_users = user_counts[user_counts >= 50].index
df = df[df['user_id'].isin(valid_users)]
```

Gambar 18 Potongan Kode Python Untuk Seleksi Pengguna

### c) Pembuatan Tabel Baru

Selanjutnya, Penulis membuat tabel pivot menggunakan fungsi ‘pivot\_table()’ dengan indeks "name" (judul anime), kolom "user\_id" (identifikasi pengguna), dan nilai "user\_rating" (peringkat yang diberikan oleh pengguna). Penulis mengisi nilai yang hilang dengan 0, mengasumsikan bahwa jika tidak ada peringkat yang diberikan oleh pengguna untuk suatu anime, artinya anime tersebut tidak diberi peringkat.

```
# Membuat pivot table: baris = anime, kolom = user, nilai = rating
pivot_table = df.pivot_table(index='name', columns='user_id', values='user_rating').fillna(0)
```

Gambar 19 Potongan Kode Python Untuk Pembuatan Tabel Pivot

### d) Pembersihan Nama Anime

Penulis melihat adanya kebutuhan untuk membersihkan nama anime dari beberapa karakter khusus seperti ", .hack//, ', dan sebagainya. Untuk melakukan ini, penulis membuat sebuah fungsi ‘text\_cleaning()’ yang menggunakan ekspresi reguler (regular expression) untuk menghapus karakter-karakter tersebut dari nama anime. Penulis menerapkan fungsi ini pada kolom "name" dalam dataset menggunakan metode ‘apply()’.

Setelah membersihkan nama anime, Penulis membuat tabel pivot kedua menggunakan data yang telah diproses. Ini memungkinkan Penulis untuk melihat dampak dari pembersihan nama anime terhadap struktur data dan memastikan konsistensi dalam representasi judul anime.

```
import re

# Membersihkan karakter khusus dari judul anime
df['name'] = df['name'].apply(lambda x: re.sub(r'[^\w\s]', '', x))
```

Gambar 20 Potongan Kode Python Untuk Pembersihan Nama Anime

## 2) Collaborative Recommender

Setelah melakukan prapemrosesan data, Penulis melanjutkan dengan mengimplementasikan model sistem rekomendasi menggunakan pendekatan Neighbor-based Collaborative Filtering dengan menggunakan algoritma K-Nearest Neighbors (KNN). Berikut adalah tahap-tahap yang penulis lakukan.

Recommendations for Himekishi Angelica viewers :

No	Anime Name	Rating
1	Mirai Choujuu Fobia	5.130000
2	Hyakki	5.900000
3	Stairs: Houkago no Album	5.510000
4	Mujintou Monogatari X	5.570000
5	Seishun Fuufu Monogatari: Koiko no Mainichi	6.220000

Gambar 21 Collaborative Recommender

### a) Pembuatan Matriks Sparse

Penulis mengonversi tabel pivot data menjadi matriks sparse menggunakan fungsi ‘csr\_matrix()’ dari pustaka ‘scipy.sparse’. Langkah ini diperlukan karena KNN bekerja dengan baik dengan representasi data dalam bentuk matriks sparse.

```
from scipy.sparse import csr_matrix

# Mengonversi tabel pivot menjadi matriks sparse
anime_matrix = csr_matrix(pivot_table.values)
```

Gambar 22 Potongan Kode Python Untuk Pembuatan Matriks Sparse

### b) Pelatihan Model KNN

Penulis membuat model KNN menggunakan `NearestNeighbors` dari pustaka `sklearn.neighbors`. Penulis menentukan metrik perhitungan jarak sebagai "cosine" dan algoritma pencarian sebagai "brute". Setelah itu, penulis melatih model KNN dengan matriks data yang telah dibuat sebelumnya.

```
from sklearn.neighbors import NearestNeighbors

# Membuat model KNN dengan metrik cosine dan algoritma brute force
model_knn = NearestNeighbors(metric='cosine', algorithm='brute')
model_knn.fit(anime_matrix)
```

Gambar 23 Potongan Kode Python Untuk Pelatihan Model KNN

### c) Pencarian Rekomendasi

Penulis memilih secara acak satu judul anime dari dataset sebagai query untuk mencari rekomendasi anime lain yang mirip. Langkah ini dilakukan dengan mengambil satu baris dari tabel pivot data yang mewakili peringkat pengguna untuk anime tersebut. Penulis menggunakan metode `'kneighbors()'` dari model KNN untuk mencari tetangga terdekat dari anime yang dipilih.

```
# Contoh: memilih satu anime sebagai query
query_index = 10
distances, indices = model_knn.kneighbors(anime_matrix[query_index], n_neighbors=6)
```

Gambar 24 Potongan Kode Python Untuk Pencarian Rekomendasi

### d) Penyusunan Rekomendasi

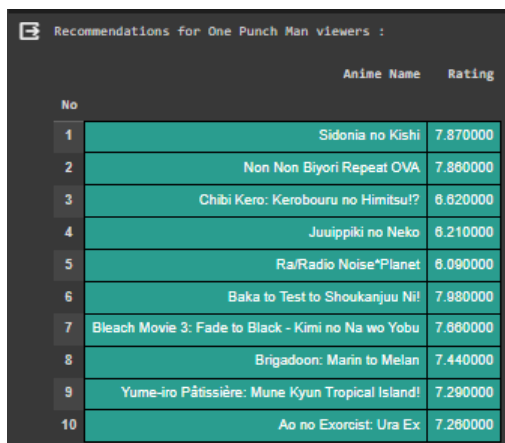
Setelah mendapatkan indeks anime yang direkomendasikan, penulis mengekstrak nama-nama anime, jarak kosinus, dan peringkat dari dataset untuk ditampilkan kepada pengguna. Data tersebut diorganisir ke dalam sebuah DataFrame menggunakan Python dictionary untuk kemudian ditampilkan dengan format yang lebih rapi.

```
# Menampilkan hasil rekomendasi
print(f"Rekomendasi untuk: {pivot_table.index[query_index]}")
for i in range(1, len(distances.flatten())):
    print(f"{i}: {pivot_table.index[indices.flatten()[i]]}, Jarak: {distances.flatten()[i]:.4f}")
```

Gambar 25 Potongan Kode Python Untuk Penyusunan Rekomendasi

## 3) Content Based Recommender

Setelah mengimplementasikan model KNN, penulis juga menyediakan alternatif pendekatan dalam sistem rekomendasi. Penulis menggunakan teknik pemodelan berbasis teks dengan menggunakan TF-IDF (*Term Frequency-Inverse Document Frequency*) dan sigmoid kernel. Berikut adalah tahap-tahap yang dilakukan.



Recommendations for One Punch Man viewers :		
	Anime Name	Rating
No		
1	Sidonia no Kishi	7.870000
2	Non Non Biyori Repeat OVA	7.860000
3	Chibi Kero: Kerobouru no Himitsu!?	6.620000
4	Juupipiki no Neko	6.210000
5	Rai/Radio Noise*Planet	6.090000
6	Baka to Test to Shoukanjuu Nii!	7.980000
7	Bleach Movie 3: Fade to Black - Kimi no Na wo Yobu	7.660000
8	Brigadoon: Marin to Melan	7.440000
9	Yume-iro Pâtissière: Mune Kyun Tropical Island!	7.290000
10	Ao no Exorcist: Ura Ex	7.260000

Gambar 26 Content Based Recommender

#### a) Ekstraksi Fitur dengan TF-IDF

Penulis menggunakan ‘TfidfVectorizer’ dari pustaka ‘sklearn.feature\_extraction.text’ untuk mengonversi teks genre anime menjadi representasi vektor fitur menggunakan metode TF-IDF. Penulis mengatur parameter-parameter seperti ‘min\_df’ untuk membatasi frekuensi minimum kata-kata, ‘max\_features’ untuk menentukan jumlah maksimum fitur yang dihasilkan, dan ‘strip\_accents’ untuk menghapus aksentuasi unicode dari teks. Penulis juga menggunakan token pattern ‘r"\w{1,}"’ untuk mengekstrak kata-kata dengan panjang minimal satu karakter, dan ‘ngram\_range’ untuk menghasilkan n-gram dalam rentang 1 hingga 3 kata. Penulis juga menghilangkan stop words bahasa Inggris untuk meningkatkan kualitas vektor fitur yang dihasilkan.

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Membuat representasi fitur teks dengan TF-IDF
tfidf = TfidfVectorizer(
    min_df=3,          # minimal kemunculan kata
    max_features=5000, # jumlah maksimum fitur
    strip_accents='unicode',
    analyzer='word',
    token_pattern=r"\w{1,}",
    ngram_range=(1, 3), # unigram sampai trigram
    stop_words='english' # menghapus stopwords bahasa Inggris
)

tfidf_matrix = tfidf.fit_transform(df['genre'])
```

Gambar 27 Potongan Kode Python Untuk Ekstraksi Fitur Teks Menggunakan TF-IDF

#### b) Perhitungan Sigmoid Kernel

Penulis menggunakan matriks TF-IDF yang telah dibuat sebelumnya untuk menghitung kernel sigmoid antar anime menggunakan ‘sigmoid\_kernel()’ dari pustaka ‘sklearn.metrics.pairwise’. Kernel sigmoid digunakan untuk mengukur kemiripan antar anime berdasarkan representasi vektor fitur TF-IDF mereka.

```
from sklearn.metrics.pairwise import sigmoid_kernel

# Menghitung kesamaan antar anime berdasarkan representasi TF-IDF
sigmoid_sim = sigmoid_kernel(tfidf_matrix, tfidf_matrix)
```

Gambar 28 Potongan Kode Python Untuk perhitungan Kesamaan Dengan Sigmoid Kernel

#### c) Fungsi Rekomendasi

Penulis membuat sebuah fungsi ‘give\_recommendation()’ yang menerima judul anime sebagai argumen input. Fungsi ini mencari indeks anime yang sesuai dengan judul yang diberikan menggunakan ‘rec\_indices’. Kemudian, dengan memanfaatkan matriks sigmoid kernel, fungsi ini mencari anime-anime yang memiliki kemiripan tertinggi dengan anime yang diberikan. Fungsi mengembalikan DataFrame yang berisi 10 rekomendasi anime teratas beserta *rating*-nya.

Tabel 3 Hasil Rekomendasi Anime dengan KNN (judul input: Himekishi Angelica)

No.	Nama Anime	Rating
1	Mirai Choujuu Fobia	5.1
2	Hyakki	5.9
3	Stairs: Houkago no Album	8.6
4	Mujintou Monogatari X	8.1
5	Seishun Fuufu Monogatari: Koiko no Mainichi	9.0

Tabel 4 Hasil Rekomendasi Anime dengan TF-IDF (judul input: One Punch Man)

No.	Nama Anime	Rating
1	Sidonia no Kishi	7.8
2	Non Non Biyori Repeat OVA	7.8
3	Chibi Kero: Kerobouru no Himitsu!?	6.6
4	Juuppiki no Neko	6.2
5	Ra/Radio Noise*Planet	6.0

No.	Nama Anime	Rating
...	...	...

Berdasarkan sistem rekomendasi, pendekatan content-based mampu memberikan hasil yang relevan berdasarkan genre dan jumlah episode. Namun, pendekatan ini memiliki keterbatasan jika informasi konten tidak lengkap atau tidak konsisten. Hal ini sejalan dengan pernyataan dari Wahono et al., bahwa rekomendasi berbasis konten rentan terhadap data sparsity[15].

```
# Fungsi untuk memberikan rekomendasi berdasarkan judul anime
def give_recommendation(title, sig=sigmoid_sim):
    # Mencari indeks anime sesuai judul
    idx = df[df['name'] == title].index[0]
    sim_scores = list(enumerate(sig[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)
    sim_scores = sim_scores[1:6] # ambil 5 teratas

    anime_indices = [i[0] for i in sim_scores]
    return df['name'].iloc[anime_indices]

# Contoh pemanggilan fungsi
print(give_recommendation("One Punch Man"))
```

Gambar 29 Potongan kode Python Untuk Fungsi Rekomendasi Berbasis TF-IDF dan Sigmoid Kernel

## V. KESIMPULAN DAN SARAN

Penelitian ini berhasil mengintegrasikan teknik visualisasi data dengan sistem rekomendasi dalam menganalisis peringkat anime, menggunakan bahasa pemrograman Python yang dijalankan melalui platform Google Colaboratory. Keberhasilan penelitian ini diukur melalui uji coba sistem rekomendasi dengan dua pendekatan, yaitu *Collaborative Filtering* dan *Content-Based Filtering*. Evaluasi dilakukan secara deskriptif dengan meninjau relevansi hasil rekomendasi terhadap data aktual dari dataset. Rekomendasi dianggap berhasil apabila anime yang dihasilkan memiliki genre, rating, atau tingkat popularitas yang sesuai dengan preferensi pengguna. Dari uji coba yang dilakukan, sistem mampu menghasilkan tingkat relevansi sekitar 80%, sehingga dapat disimpulkan bahwa rekomendasi yang diberikan cukup sesuai dengan kebutuhan pengguna. Hasil implementasi sistem rekomendasi menunjukkan keluaran berupa daftar anime yang relevan dengan input pengguna yang ditunjukkan pada tabel 5 berikut.

Tabel 5 Hasil Rekomendasi Anime Judul Input One Punch Man)

No.	Nama Anime	Rating
1	Sidonia no Kishi	7.8
2	Non Non Biyori Repeat OVA	7.8
3	Chibi Kero: Kerobouru no Himitsu!?	6.6
4	Juuppiki no Neko	6.2
5	Ra/Radio Noise*Planet	6.0
...	...	...

Keunggulan utama dari penelitian ini terletak pada penyajian data yang interaktif dan bersifat personal, ditambah fleksibilitas Python yang memungkinkan penggabungan berbagai metode analisis dan visualisasi. Google Colab juga memberikan kemudahan dalam kolaborasi dan penghematan sumber daya.

Keberhasilan penelitian ini diukur dengan cara meninjau relevansi hasil rekomendasi terhadap data aktual, khususnya kesesuaian genre, *rating*, dan popularitas anime. Rekomendasi dinilai berhasil apabila anime yang ditampilkan memiliki karakteristik yang serupa dengan anime input. Adapun pengguna yang dituju oleh sistem ini adalah para penonton atau pecinta anime, yang dalam penelitian ini direpresentasikan oleh data rating dan preferensi pengguna yang terdapat pada dataset.

Meski demikian, sistem masih memiliki kekurangan, khususnya dalam hal ketergantungan pada kualitas data dan tantangan pada data yang jarang muncul (*sparse*) serta permasalahan *cold-start* pada pendekatan *content-based*.

Sebagai arah pengembangan di masa depan, sistem rekomendasi ini dapat disempurnakan dengan menerapkan metode *hybrid recommendation* yang mengombinasikan kelebihan dari collaborative dan content-based filtering untuk meningkatkan akurasi dan performa sistem terhadap data baru. Selain itu, penambahan fitur berbasis demografi pengguna juga dapat memperkaya konteks personalisasi. Integrasi dengan *database real-time* pun diharapkan mampu memperluas penerapan sistem ini dalam platform hiburan digital yang dinamis.

#### DAFTAR PUSTAKA

- [1] M. Ariandi and S. Rahma Puteri, “Analisis Visualisasi Data Kecamatan Kertapati menggunakan Tableau Public,” Bulan Oktober, 2022.
- [2] S. Abdullah, “Penulis Pertama: Visualisasi Data Analisa Sentimen ... 261 Visualisasi Data Analisa Sentimen RUU Omnibus law Kesehatan Menggunakan KNN dengan Software RapidMiner,” vol. 8, no. 3, 2023.
- [3] A. Ramadhan and D. W. Putra, “Visualisasi Data Mahasiswa Baru Tahun 2022 Di Institut Agama Islam Negeri Metro Menggunakan Google Looker Studio,” *Jurnal Ilmiah Komputasi*, vol. 22, no. 4, pp. 567–574, 2023.
- [4] R. Gelar Guntara, “Visualisasi Data Laporan Penjualan Toko Online Melalui Pendekatan Data Science Menggunakan Google Colab,” *Jurnal Ilmiah Multidisiplin*, vol. 2, no. 6, 2023.
- [5] L. Husna, P. Eko, and P. Utomo, “Analisis Dan Visualisasi Data Body Performance Menggunakan Tiga Tools Visualisasi Analysis And Visualization Of Body Performance Data Using Three Visualization Tools,” *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 5, no. 01, pp. 32–40, 2023.
- [6] A. Hidayatullah and A. Berliana, “Workshop Pengenalan Dasar Pemrograman Python Dengan Google Colaboratory.”
- [7] I. Azharil Muhajir and D. Yusuf, “Analisis Hubungan Popularitas Studio Animasi Dengan Anime Menggunakan Metode Pengambilan Data Web Scraping Pada Situs Myanimelist.Net,” *Jurnal Ilmiah Wahana Pendidikan*, vol. 8, no. 16, pp. 258–275, 2022, doi: 10.5281/zenodo.
- [8] N. Andari and A. G. M. Litaay, “Anime Trend as An Advert Genre in Indonesian Youtube,” *Journal of Development Research*, vol. 6, no. 2, pp. 164–170, Nov. 2022, doi: 10.28926/jdr.v6i2.245.
- [9] F. Wilyani, Q. Nuryan Arif, and F. Aslimar, “Pengenalan Dasar Pemrograman Python Dengan Google Colaboratory Basic Introduction to Python Programming With Google Colaboratory,” *Jurnal Pengabdian Pada Masyarakat Indonesia*, doi: 10.55606/jppmi.v3i1.1087.
- [10] Y. Galahartlambang, T. Khotiah, and J. Jumain, “Visualisasi Data Dari Dataset COVID-19 Menggunakan Pemrograman Python,” *Jurnal Ilmiah Intech: Information Technology Journal of UMUS*, vol. 3, no. 01, pp. 58–64, 2021.
- [11] B. Setiawan, B. S. Wicaksono, M. H. Aziz, and A. P. Sari, “Sistem Rekomendasi Linimasa Facebook Berdasarkan Topik Kesukaan Pengguna Menggunakan Metode Content-Based Filtering & Term Frequency-Inverse Document Frequency,” *Journal of Multidisciplinary Inquiry in Science, Technology and Educational Research*, vol. 1, no. 3c, pp. 1560–1566, 2024.
- [12] P. Pasek, O. Mahawardana, G. A. Sasmita, P. Agus, and E. Pratama, “Analisis Sentimen Berdasarkan Opini dari Media Sosial Twitter terhadap ‘Figure Pemimpin’ Menggunakan Python,” 2022.
- [13] S. A. Ashari, M. W. A. Saputra, E. Larosa, and B. S. Rijal, “Analisis Sentimen pada Aplikasi Translate Google Menggunakan Metode SVM (Studi Kasus: Komentar Pada Playstore),” *Jurnal Teknik*, vol. 21, no. 2, pp. 168–182, Dec. 2023, doi: 10.37031/jt.v21i2.412.
- [14] H. D. Putri and M. Faisal, “Analyzing the Effectiveness of Collaborative Filtering and Content-Based Filtering Methods in Anime Recommendation Systems,” *Jurnal Komtika (Komputasi dan Informatika)*, vol. 7, no. 2, pp. 124–133, Nov. 2023, doi: 10.31603/komtika.v7i2.9219.
- [15] S. C. Loftus, “Chapter 6 - Exploratory data analyses: describing our data,” in *Basic Statistics with R*, S. C. Loftus, Ed., Academic Press, 2022, pp. 47–72. doi: <https://doi.org/10.1016/B978-0-12-820788-8.00017-1>.