



SISTEM REKOMENDASI PRODUK E-COMMERCE MENGUNAKAN COLLABORATIVE FILTERING DAN CONTENT-BASED FILTERING

*(E-Commerce Product Recommendation System Using Collaborative
Filtering and Content-Based Filtering)*

Ani Rachmaniar¹, Susi Widayati², Kokoy Rokoyah³

Program Studi Manajemen Informatika¹, Program Studi Sistem Informasi²,
Program Studi Manajemen Informatika³
STMIK Jakarta STI&K^{1,2,3}, Jakarta

anistore700@gmail.com¹, widayatisusi@gmail.com²,
kokoyrokoyah@gmail.com³

Received: April 27, 2025. **Revised:** May 6, 2025. **Accepted:** May 7, 2025.
Issue Period: Vol.9 No.1 (2025), Pp. 40-54

Abstrak: Sistem rekomendasi merupakan komponen penting dalam platform e-commerce untuk meningkatkan pengalaman pengguna, mempercepat pencarian produk, dan mendorong transaksi penjualan. Dengan meningkatnya volume interaksi pengguna dan ragam produk, dibutuhkan sistem yang cerdas dan adaptif. Pendekatan tunggal seperti Collaborative Filtering atau Content-Based Filtering seringkali menghadapi kendala seperti masalah cold start dan data sparsity. Penelitian ini mengembangkan sistem rekomendasi hibrida dengan menggabungkan kedua pendekatan tersebut untuk mengatasi keterbatasan masing-masing. Collaborative Filtering bekerja dengan menganalisis pola interaksi pengguna terhadap produk, misalnya melalui rating, guna menemukan kesamaan preferensi antar pengguna, menggunakan teknik seperti cosine similarity. Di sisi lain, Content-Based Filtering merekomendasikan produk berdasarkan kemiripan fitur atau deskripsi produk dengan riwayat interaksi pengguna sebelumnya, dengan memanfaatkan teknik vektorisasi teks seperti TF-IDF. Sistem diuji menggunakan dataset e-commerce yang telah dikurasi, dan kinerjanya diukur menggunakan metrik evaluasi precision, recall, dan F1-score. Hasil evaluasi menunjukkan bahwa pendekatan hibrida menghasilkan rekomendasi yang lebih relevan dibandingkan metode tunggal. Sistem dikembangkan menggunakan Python di lingkungan Jupyter Notebook, dengan proses mencakup pengolahan dataset, implementasi algoritma rekomendasi berbasis kemiripan item dan konten, serta penggabungan skor rekomendasi melalui skema weighting yang dapat disesuaikan. Selain itu, visualisasi data turut disertakan untuk membantu pemahaman pengguna terhadap hasil dan distribusi rekomendasi. Sistem ini mampu menghasilkan rekomendasi yang personal dan kontekstual, serta fleksibel untuk diterapkan pada platform e-commerce berskala kecil hingga menengah.



DOI: 10.52362/jisicom.v9i1.1904

Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



Kata kunci: Sistem Rekomendasi, Collaborative Filtering, Content-Based Filtering, E-Commerce, Machine Learning.

Abstract: Recommendation systems are an essential component in e-commerce platforms to enhance user experience, accelerate product discovery, and drive sales transactions. With the increasing volume of user interactions and product variety, intelligent and adaptive systems are needed. Single approaches such as Collaborative Filtering or Content-Based Filtering often face constraints such as cold start and data sparsity issues. This study develops a hybrid recommendation system by combining both approaches to overcome their respective limitations. Collaborative Filtering works by analyzing user interaction patterns towards products, for example through ratings, to find similarities in preferences between users, using techniques such as cosine similarity. On the other hand, Content-Based Filtering recommends products based on the similarity of product features or descriptions with the user's previous interaction history, by utilizing text vectorization techniques such as TF-IDF. The system is tested using a curated e-commerce dataset, and its performance is measured using precision, recall, and F1-score evaluation metrics. The evaluation results show that the hybrid approach produces more relevant recommendations than either method alone. The system is developed using Python in a Jupyter Notebook environment, with the process including dataset processing, implementation of item and content similarity-based recommendation algorithms, and aggregation of recommendation scores through a customizable weighting scheme. In addition, data visualization is included to help users understand the results and distribution of recommendations. The system is capable of producing personalized and contextual recommendations, and is flexible to be applied to small to medium-sized e-commerce platforms..

Keywords: Recommendation Systems, Collaborative Filtering, Content-Based Filtering, E-Commerce, Machine Learning.

I. PENDAHULUAN

Perkembangan teknologi informasi dan komunikasi dalam beberapa dekade terakhir telah memberikan dampak signifikan terhadap berbagai sektor kehidupan, termasuk dalam bidang perdagangan. Salah satu bentuk transformasi paling nyata dari digitalisasi perdagangan adalah kemunculan dan pertumbuhan pesat platform e-commerce. E-commerce, atau perdagangan elektronik, memungkinkan transaksi jual beli barang dan jasa dilakukan secara daring melalui perangkat komputer maupun ponsel pintar. Fenomena ini memberikan kemudahan luar biasa bagi konsumen dalam mencari, memilih, dan membeli produk tanpa batasan geografis. E-commerce memberikan peluang besar bagi pelaku usaha untuk menjangkau konsumen lebih luas, mengoptimalkan biaya operasional, serta memanfaatkan data digital untuk meningkatkan strategi pemasaran. Seiring meningkatnya jumlah produk dan pengguna dalam platform e-commerce, tantangan baru pun muncul, yakni bagaimana memberikan pengalaman belanja yang efisien, personal, dan relevan bagi setiap pengguna. Salah satu solusi yang banyak digunakan untuk menjawab tantangan tersebut adalah dengan mengimplementasikan sistem rekomendasi. Sistem rekomendasi bertujuan untuk membantu pengguna menemukan produk yang sesuai dengan preferensi dan kebutuhan mereka, sekaligus meningkatkan potensi penjualan bagi penyedia layanan e-commerce.

Sistem rekomendasi telah diterapkan secara luas dalam berbagai platform e-commerce. Pendekatan tersebut tidak mempertimbangkan perilaku unik dari setiap pengguna maupun karakteristik mendalam dari produk. Bertambahnya jumlah data pengguna dan produk, sistem rekomendasi yang tidak adaptif menjadi sulit untuk mengelola informasi secara efisien, mengakibatkan rekomendasi yang bersifat umum dan kurang menarik bagi pengguna. Masalah cold start, yakni kondisi di mana sistem kesulitan memberikan rekomendasi untuk pengguna baru atau produk baru karena kurangnya data historis, juga menjadi tantangan tersendiri. Isu sparsity atau kelangkaan data dalam matriks interaksi pengguna dan produk yang membuat prediksi preferensi menjadi tidak akurat. Pendekatan seperti Collaborative Filtering dan Content-Based Filtering muncul sebagai dua metode utama yang dapat saling melengkapi dalam menciptakan sistem rekomendasi yang lebih efektif dan personal.





Collaborative Filtering (CF) adalah metode populer dalam sistem rekomendasi yang memanfaatkan data interaksi atau penilaian pengguna untuk memprediksi preferensi berdasarkan kesamaan antar pengguna (user-based) atau produk (item-based). CF efektif dalam e-commerce karena mampu mengungkap preferensi tersembunyi dari pola pembelian serupa, meski produk yang disukai berbeda. Tantangan utama metode ini adalah cold start dan data sparsity, terutama saat pengguna baru belum memiliki riwayat interaksi. Meski begitu, CF tetap efektif jika tersedia banyak data dan partisipasi aktif dari pengguna.

Content-Based Filtering (CBF) merekomendasikan produk berdasarkan kesamaan fitur dengan preferensi eksplisit pengguna, seperti kategori, merek, atau bahan. Metode ini efektif untuk pengguna baru karena tidak bergantung pada data interaksi. Namun, kualitas rekomendasi sangat bergantung pada kelengkapan data produk. Sementara itu, Collaborative Filtering (CF) mengenali pola antar pengguna. Penggabungan keduanya dalam sistem hybrid menjadi solusi ideal untuk mengatasi masalah cold start dan data sparsity. Pendekatan ini menghasilkan rekomendasi yang lebih akurat dan personal. Pada penelitian ini, sistem hybrid diimplementasikan menggunakan Python di Jupyter Notebook untuk eksplorasi data, pengujian model, dan visualisasi hasil.

Pengembangan sistem rekomendasi ini dilakukan menggunakan Python dan Jupyter Notebook. Python dipilih karena mendukung analisis data dan pembelajaran mesin melalui pustaka seperti Pandas, NumPy, Scikit-Learn, dan Surprise. Jupyter Notebook digunakan untuk memuat data, preprocessing, membangun model Collaborative dan Content-Based Filtering, serta menggabungkannya dalam sistem hybrid. Penelitian ini bertujuan meningkatkan relevansi rekomendasi, personalisasi, dan mengatasi masalah cold start serta sparsity. Sistem diuji menggunakan data e-commerce nyata, dan hasilnya menunjukkan potensi untuk diterapkan sebagai solusi rekomendasi yang adaptif, cerdas, serta bermanfaat bagi pelaku e-commerce dalam meningkatkan kepuasan dan daya saing bisnis.

II. METODE PENELITIAN

Penelitian ini mengadopsi pendekatan rekayasa sistem informasi melalui tahapan analisis, perancangan, implementasi, dan evaluasi terhadap sistem rekomendasi hibrida. Sistem rekomendasi dikembangkan dengan menggabungkan dua pendekatan utama, yaitu Collaborative Filtering (CF) dan Content-Based Filtering (CBF). Tujuan dari metode ini adalah untuk menghasilkan sistem rekomendasi yang lebih akurat, relevan, dan mampu mengatasi permasalahan cold start serta data sparsity yang umum terjadi pada sistem rekomendasi tradisional. Berikut ini adalah tahapan metode penelitian:

1. Desain Arsitektur Sistem

Sistem rekomendasi yang dikembangkan terdiri dari Empat komponen utama:

- Modul Pengolahan Data: untuk membaca, membersihkan, dan memformat data transaksi serta data produk.
- Modul Rekomendasi: yang terdiri dari algoritma CF dan CBF.
- Modul Integrasi Hibrida: yang menggabungkan hasil rekomendasi dari CF dan CBF menggunakan pendekatan weighted hybrid.
- Modul Visualisasi: untuk menampilkan hasil rekomendasi dan analisis data secara grafis.

2. Dataset dan Preprocessing

Dataset yang digunakan merupakan data transaksi e-commerce dan informasi produk yang telah dikurasi. Adapun langkah-langkah preprocessing meliputi:

- Normalisasi dan pembersihan data (missing values, duplikasi)
- Pembuatan matriks interaksi pengguna-produk
- Tokenisasi dan vektorisasi teks deskripsi produk menggunakan TF-IDF
- Penghapusan stopwords dan stemming (jika diperlukan)

3. Implementasi Collaborative Filtering

Metode CF diterapkan dengan pendekatan item-based, karena cocok untuk kondisi data yang sparse. Langkah-langkahnya adalah:





- Penyusunan matriks interaksi pengguna–produk berbasis rating atau jumlah pembelian.
- Perhitungan cosine similarity antar item untuk menentukan tingkat kemiripan produk.
- Estimasi skor rekomendasi bagi pengguna berdasarkan item serupa yang pernah dinilai tinggi.

4. Implementasi Content-Based Filtering

Metode CBF memanfaatkan fitur konten dari produk, khususnya deskripsi teks. Langkah-langkah utama:

- Representasi deskripsi produk dengan teknik TF-IDF vectorization.
- Perhitungan cosine similarity antar produk berdasarkan representasi TF-IDF.
- Rekomendasi produk diberikan kepada pengguna berdasarkan kemiripan produk baru dengan produk yang pernah berinteraksi sebelumnya.

5. Integrasi Hibrida

Sistem rekomendasi hibrida dibentuk dengan menggabungkan skor rekomendasi dari CF dan CBF menggunakan formula sebagai berikut:

$$\text{SkorHybrid} = \alpha \cdot \text{SkorCF} + (1 - \alpha) \cdot \text{SkorCBF}$$

Dengan parameter α sebagai bobot pembobotan yang dapat disesuaikan. Penelitian ini melakukan eksperimen dengan berbagai nilai α untuk mencari kombinasi optimal.

6. Evaluasi Sistem

Pengukuran performa sistem rekomendasi, digunakan tiga metrik evaluasi umum:

- Precision: rasio item relevan yang direkomendasikan dari total item yang direkomendasikan.
- Recall: rasio item relevan yang direkomendasikan dari total item relevan yang tersedia.
- F1-score: harmonisasi antara precision dan recall.
- Evaluasi dilakukan dengan membandingkan performa CF, CBF, dan Hybrid terhadap data uji.

7. Pengembangan dan Tools

Seluruh sistem dikembangkan menggunakan bahasa pemrograman Python dan dijalankan di lingkungan Jupyter Notebook. Adapun library yang digunakan meliputi:

- Pandas dan Numpy untuk manipulasi data
- Scikit-learn untuk TF-IDF dan evaluasi metrik
- Matplotlib dan Seaborn untuk visualisasi data dan hasil

8. Visualisasi dan Analisis Hasil

Visualisasi hasil digunakan untuk memudahkan interpretasi performa sistem rekomendasi. Bentuk visualisasi meliputi:

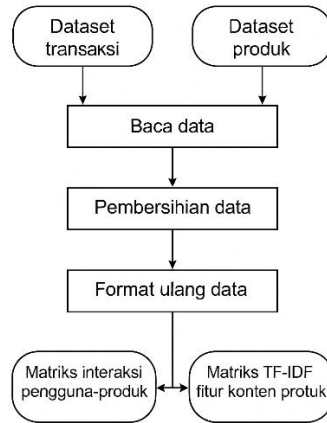
- Heatmap kesamaan antar item
- Grafik precision, recall, dan F1-score
- Distribusi skor rekomendasi yang dihasilkan sistem

III. PEMBAHASAN DAN HASIL

3.1. Proses Pengolahan Data

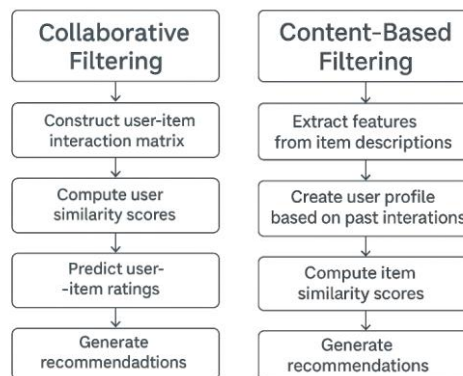
Modul Pengolahan Data berperan sebagai tahap awal dalam sistem rekomendasi yang bertugas membaca, membersihkan, dan memformat data transaksi pengguna serta data deskripsi produk. Proses ini memastikan data yang digunakan dalam sistem memiliki kualitas dan struktur yang sesuai untuk dianalisis lebih lanjut.





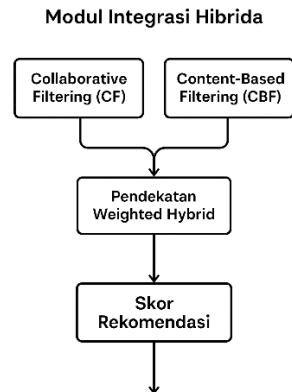
Gambar 1. Proses Pengolahan Data

Modul Rekomendasi terdiri dari dua algoritma utama, yaitu Collaborative Filtering (CF) dan Content-Based Filtering (CBF). CF bekerja dengan menganalisis pola interaksi antar pengguna, sementara CBF memanfaatkan kemiripan fitur konten produk untuk menghasilkan rekomendasi yang relevan.



Gambar 2. Algoritma CF dan CBF

Modul Integrasi Hibrida menggabungkan skor rekomendasi dari CF dan CBF menggunakan pendekatan weighted hybrid. Pendekatan ini memungkinkan sistem menghasilkan rekomendasi yang lebih akurat dengan mengombinasikan keunggulan kedua metode.



Gambar 3. Algoritma Hybrid

Modul Visualisasi berfungsi menampilkan hasil rekomendasi serta analisis data secara grafis. Modul ini membantu pengguna dan peneliti dalam memahami distribusi data, performa sistem, serta interpretasi hasil rekomendasi secara visual dan informatif.

3.2 Dataset dan Preprocessing

Pengembangan sistem rekomendasi ini, digunakan dataset yang terdiri dari dua sumber utama, yaitu data transaksi e-commerce dan data informasi produk. Dataset transaksi memuat informasi historis interaksi pengguna terhadap produk, seperti pemberian rating, pembelian, atau klik. Sementara itu, dataset produk mencakup detail atribut produk seperti nama, kategori, dan deskripsi produk. Data dapat digunakan secara optimal dalam proses rekomendasi, dilakukan beberapa tahap preprocessing sebagai berikut:

1. Normalisasi dan Pembersihan Data

Langkah pertama mencakup pembersihan dataset dari missing values (nilai kosong) dan duplikasi entri. Data yang tidak lengkap atau berulang dapat mempengaruhi akurasi sistem rekomendasi, sehingga perlu diidentifikasi dan dibersihkan. Selain itu, dilakukan konversi format data agar seragam, seperti standarisasi huruf, format tanggal, dan tipe data numerik.

```
python

import pandas as pd

df = pd.read_csv('transaksi.csv')
df = df.drop_duplicates()
df = df.dropna()
df['rating'] = df['rating'].astype(float)
```

Gambar 4. Program Python Normalisasi Data

2. Pembuatan Matriks Interaksi Pengguna–Produk

Pembersihan data transaksi kemudian akan dibentuk matriks interaksi pengguna–produk yang menjadi dasar untuk metode Collaborative Filtering. Matriks memiliki baris mewakili pengguna dan kolom mewakili produk dimana nilai berupa skor interaksi seperti rating atau frekuensi pembelian.

```
python Copy Edit  
  
import pandas as pd  
  
# Contoh data transaksi  
df = pd.read_csv('transaksi.csv') # berisi user_id, product_id, rating/frequency  
  
# Membuat matriks interaksi  
interaction_matrix = df.pivot_table(index='user_id', columns='product_id', values='rating', f
```

Gambar 5. Program Python membuat matrik pengguna-produk

4. Tokenisasi dan Vektorisasi Deskripsi Produk

Mendukung pendekatan Content-Based Filtering deskripsi produk diproses melalui tokenisasi, yaitu memecah teks menjadi kata-kata kunci. Vektorisasi menggunakan metode Term Frequency–Inverse Document Frequency (TF-IDF) dengan mengubah data teks menjadi representasi numerik yang menggambarkan pentingnya suatu kata dalam sebuah deskripsi produk relatif terhadap semua deskripsi dalam dataset.

```
python Copy Edit  
  
from sklearn.feature_extraction.text import TfidfVectorizer  
  
tfidf = TfidfVectorizer()  
tfidf_matrix = tfidf.fit_transform(df['cleaned_description'])  
  
print(tfidf_matrix.shape) # bentuk matriks: (jumlah produk, jumlah fitur/kata unik)
```

Gambar 6. Program Python Vektorisasi TF-IDF

5. Penghapusan Stopwords dan Stemming (Opsional)

Meningkatkan kualitas representasi teks dilakukan penghapusan stopwords (kata-kata umum seperti “dan”, “adalah”, dll.) yang tidak memberikan nilai semantik signifikan dalam proses analisis. Stemming dapat diterapkan untuk mengubah kata ke bentuk dasarnya guna menyederhanakan variasi kata yang memiliki makna serupa.

```
python Copy  
  
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory  
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory  
  
# Stopwords  
stop_factory = StopWordRemoverFactory()  
stopwords = stop_factory.get_stop_words()  
  
# Stemming  
stemmer_factory = StemmerFactory()  
stemmer = stemmer_factory.create_stemmer()  
  
text = "Produk ini sangat bagus dan berkualitas tinggi"  
clean_text = stemmer.stem(text)  
  
print(clean_text)
```

Gambar 7. Program Python Stopwords dan Stemming Library Sastrawi

3.3. Implementasi Collaborative Filtering dan Content-Based Filtering

Berikut ini adalah tahapan menggunakan kode program untuk proses filtering menggunakan algoritma Collaborative Filtering dan Content-Based Filtering

```
python Copy Edit  
  
import pandas as pd  
from sklearn.metrics.pairwise import cosine_similarity  
from sklearn.preprocessing import StandardScaler  
import numpy as np  
  
# -----  
# Step 1: Dataset simulasi  
# -----  
# Data transaksi pengguna-produk (rating atau jumlah pembelian)  
data = {  
    'user_id': [1, 1, 1, 2, 2, 3, 3, 4],  
    'product_id': ['A', 'B', 'C', 'A', 'D', 'B', 'E', 'C'],  
    'rating': [5, 4, 3, 4, 5, 5, 2, 4]  
}  
df = pd.DataFrame(data)
```

Gambar 8. Sampel Kode Python Algoritma CF


```
python

import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity

# 1. Contoh Dataset Produk (ID dan Deskripsi)
data_produk = pd.DataFrame({
    'product_id': [1, 2, 3, 4],
    'description': [
        'Kemeja pria lengan panjang bahan katun nyaman dipakai',
        'Celana jeans wanita slim fit biru muda',
        'Jaket kulit pria model terbaru stylish',
        'Kemeja pria slim fit putih bahan lembut'
    ]
})
```

Gambar 9. Sampel Kode Python Algoritma CBF

```
python

import numpy as np
import pandas as pd

# Contoh data skor hasil dari Collaborative Filtering (CF)
# Format: user_id, product_id, skor_cf
cf_scores = pd.DataFrame({
    'user_id': [1, 1, 1, 2, 2],
    'product_id': [101, 102, 103, 101, 104],
    'skor_cf': [0.9, 0.6, 0.4, 0.8, 0.5]
})

# Contoh data skor hasil dari Content-Based Filtering (CBF)
# Format: user_id, product_id, skor_cbf
cbf_scores = pd.DataFrame({
```

Gambar 10. Sampel Kode Python Hybrid CF-CBF

3.4. Visualisasi dan Analisis Hasil

Visualisasi hasil digunakan untuk memudahkan interpretasi performa sistem rekomendasi dengan menyajikan data dalam bentuk grafik yang informatif dan mudah dipahami. Melalui visualisasi seperti heatmap kesamaan antar item, grafik precision, recall, dan F1-score, serta distribusi skor rekomendasi, pengguna dapat melihat pola hubungan antar produk, mengevaluasi akurasi sistem dalam memberikan rekomendasi yang relevan, serta memahami sebaran nilai prediksi yang dihasilkan. Pendekatan ini membantu dalam proses analisis dan pengambilan keputusan terhadap kualitas dan efektivitas model rekomendasi yang dibangun.

```
python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import precision_score, recall_score, f1_score

# Simulasi data kesamaan antar item (item-item similarity matrix)
item_similarity = np.random.rand(10, 10) # 10 produk
np.fill_diagonal(item_similarity, 1)

# Heatmap kesamaan antar item
plt.figure(figsize=(8,6))
sns.heatmap(item_similarity, annot=True, cmap='coolwarm')
```

Gambar 11. Sampel Program Python untuk Visualisasi Grafik

3.5. Hasil Pengembangan

Pengembangan sistem rekomendasi dilakukan menggunakan bahasa pemrograman Python yang dijalankan di lingkungan Jupyter Notebook untuk memudahkan eksplorasi data dan dokumentasi proses. Beberapa library utama yang digunakan antara lain Pandas dan Numpy untuk keperluan manipulasi dan analisis data, Scikit-learn untuk penerapan metode TF-IDF serta perhitungan metrik evaluasi seperti precision, recall, dan F1-score, serta Matplotlib dan Seaborn yang berfungsi dalam visualisasi data dan hasil rekomendasi secara grafis. Kombinasi tools ini memungkinkan proses pengembangan yang efisien, fleksibel, dan mudah direproduksi. Berikut ini adalah tampilan Program:

1. Sampel Data Produk dan Interaksi.

Berikut ini adalah sampel data penjualan e-commerce dalam bentuk file CSV.

Tabel 1. Data Produk

produk_id	nama_produk	kategori	deskripsi
1	Kemeja Putih	Fashion Pria	Kemeja putih lengan panjang bahan katun
2	Celana Jeans	Fashion Pria	Celana jeans warna biru dengan potongan slim fit
3	Kaos Hitam	Fashion Pria	Kaos hitam polos bahan katun lembut
4	Jaket Kulit	Fashion Pria	Jaket kulit asli warna hitam dengan resleting
5	Sepatu Sneakers	Fashion Pria	Sepatu sneakers putih dengan sol karet

Tabel 2. Data Interaksi

user_id	produk_id	rating
1	1	5
1	2	4
2	2	5
2	3	3
3	3	4
3	4	5
4	5	4

2. Visualisasi Data berdasarkan Rating Produk

Berikut ini adalah pembuatan aplikasi dengan menggunakan Bahasa Pemrograman Python dengan perangkat bantu Jupyter Notebook.

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
import numpy as np

# Load dataset
produk_df = pd.read_csv('produk1.csv')
interaksi_df = pd.read_csv('interaksi1.csv')

# Content-Based Filtering
tfidf = TfidfVectorizer()
tfidf_matrix = tfidf.fit_transform(produk_df['deskripsi'])
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)

def content_based_recommendations(produk_id, top_n=3):
    idx = produk_df[produk_df['produk_id'] == produk_id].index[0]
    sim_scores = list(enumerate(cosine_sim[idx]))
    sim_scores = sorted(sim_scores, key=lambda x: x[1], reverse=True)[1:top_n+1]
    produk_indices = [i[0] for i in sim_scores]
    return produk_df.iloc[produk_indices][['produk_id', 'nama_produk']]

# Collaborative Filtering Manual
pivot_table = interaksi_df.pivot_table(index='user_id', columns='produk_id', values='rating').fillna(0)
produk_similarity = cosine_similarity(pivot_table.T)
produk_similarity_df = pd.DataFrame(produk_similarity, index=pivot_table.columns, columns=pivot_table.columns)

def collaborative_recommendations(user_id, top_n=3):
    user_ratings = pivot_table.loc[user_id]
    rated_produk = user_ratings[user_ratings > 0].index.tolist()
    score_series = pd.Series(0, index=pivot_table.columns)
    for pid in rated_produk:
        similarity_scores = produk_similarity_df[pid]
        score_series += similarity_scores * user_ratings[pid]
    for pid in rated_produk:
        score_series[pid] = 0
    recommended_ids = score_series.sort_values(ascending=False).head(top_n).index.tolist()
    return produk_df[produk_df['produk_id'].isin(recommended_ids)][['produk_id', 'nama_produk']]

# Hybrid Recommendation
def hybrid_recommendation(user_id, produk_id, alpha=0.5, top_n=3):
    idx = produk_df[produk_df['produk_id'] == produk_id].index[0]
    content_scores = list(enumerate(cosine_sim[idx]))
    user_ratings = pivot_table.loc[user_id]
    rated_produk = user_ratings[user_ratings > 0].index.tolist()
    score_series = pd.Series(0, index=pivot_table.columns)
    for pid in rated_produk:
        similarity_scores = produk_similarity_df[pid]
        score_series += similarity_scores * user_ratings[pid]
    for pid in rated_produk:
        score_series[pid] = 0
    hybrid_scores = []
    for i, sim in content_scores:
        pid = produk_df.iloc[i]['produk_id']
        if pid != produk_id:
            cf_score = score_series.get(pid, 0)
            combined_score = alpha * sim + (1 - alpha) * cf_score
            hybrid_scores.append((pid, combined_score))
    hybrid_scores.sort(key=lambda x: x[1], reverse=True)
    top_hybrid_ids = [pid for pid, _ in hybrid_scores[:top_n]]
    return produk_df[produk_df['produk_id'].isin(top_hybrid_ids)][['produk_id', 'nama_produk']]
```



```
# Contoh Eksekusi
print("Content-Based:")
print(content_based_recommendations(1))

print("\nCollaborative Filtering:")
print(collaborative_recommendations(1))

print("\nHybrid:")
print(hybrid_recommendation(1, 1))

import matplotlib.pyplot as plt
import seaborn as sns

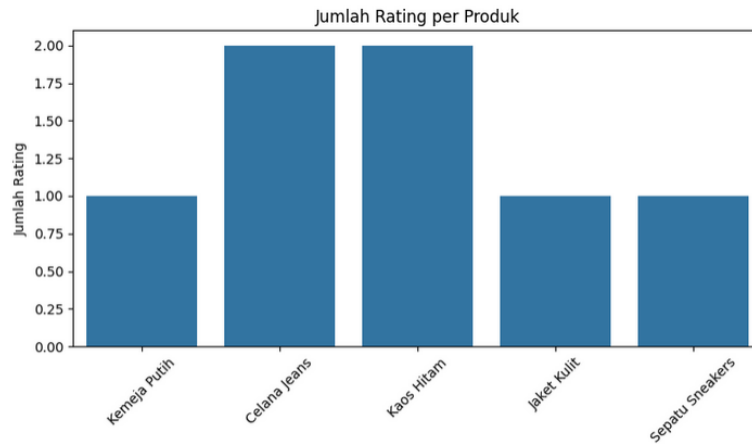
# =====
# Visualisasi 1: Jumlah Rating per Produk
# =====
plt.figure(figsize=(8,5))
rating_counts = interaksi_df.groupby('produk_id')['rating'].count()
produk_names = produk_df.set_index('produk_id').loc[rating_counts.index]['nama_produk']
sns.barplot(x=produk_names, y=rating_counts.values)
plt.title('Jumlah Rating per Produk')
plt.xlabel('Produk')
plt.ylabel('Jumlah Rating')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# =====
# Visualisasi 2: Rata-Rata Rating per Produk
# =====
plt.figure(figsize=(8,5))
avg_rating = interaksi_df.groupby('produk_id')['rating'].mean()
produk_names = produk_df.set_index('produk_id').loc[avg_rating.index]['nama_produk']
sns.barplot(x=produk_names, y=avg_rating.values)
plt.title('Rata-Rata Rating per Produk')
plt.xlabel('Produk')
plt.ylabel('Rata-Rata Rating')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()

# =====
# Visualisasi 3: Heatmap Similarity antar Produk (Content-Based)
# =====
plt.figure(figsize=(8,6))
sim_df = pd.DataFrame(cosine_sim, index=produk_df['nama_produk'], columns=produk_df['nama_produk'])
sns.heatmap(sim_df, annot=True, cmap='coolwarm')
plt.title('Content-Based Product Similarity')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()

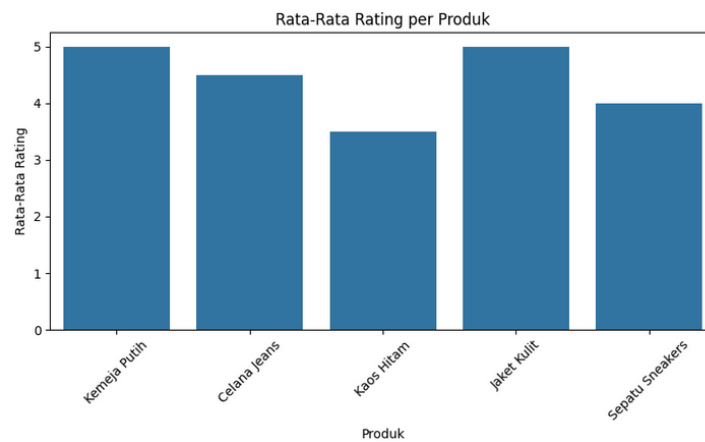
# =====
# Visualisasi 4: Heatmap Similarity antar Produk (Collaborative Filtering)
# =====
plt.figure(figsize=(8,6))
cf_sim_df = pd.DataFrame(produk_similarity, index=produk_df['nama_produk'], columns=produk_df['nama_produk'])
sns.heatmap(cf_sim_df, annot=True, cmap='YlGnBu')
plt.title('Collaborative Filtering Product Similarity')
plt.xticks(rotation=45)
plt.yticks(rotation=0)
plt.tight_layout()
plt.show()
```





Gambar 12. Jumlah Rating per Produk

Visualisasi ini menampilkan jumlah rating yang diterima oleh masing-masing produk dalam dataset. Setiap batang pada diagram batang mewakili satu produk, dengan tinggi batang menunjukkan berapa kali produk tersebut telah diberi rating oleh pengguna. Dari hasil ini, kita dapat mengidentifikasi produk mana yang paling sering berinteraksi dengan pengguna. Produk dengan jumlah rating tinggi biasanya lebih populer atau memiliki visibilitas lebih tinggi di platform e-commerce. Informasi ini penting dalam menganalisis bias popularitas dalam sistem rekomendasi.



Gambar 13. Rata-Rata Rating per Produk

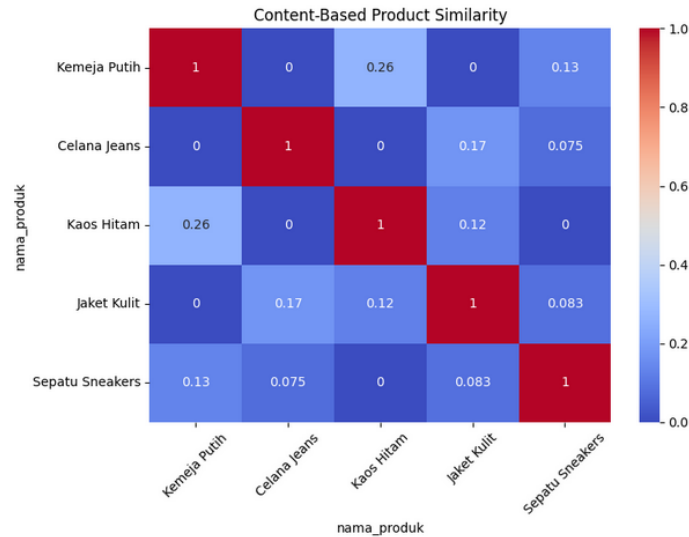
Diagram batang ini menggambarkan nilai rata-rata rating yang diberikan oleh pengguna untuk setiap produk. Rata-rata rating memberikan indikasi kualitas persepsi pengguna terhadap produk, terlepas dari seberapa sering produk tersebut dinilai. Produk dengan rata-rata rating tinggi dianggap disukai pengguna meskipun jumlah interaksinya sedikit. Kombinasi informasi dari visualisasi ini dan visualisasi pertama dapat membantu menentukan apakah sebuah produk layak dipromosikan lebih lanjut dalam rekomendasi.

3. Visualisasi Content-Based Filtering



DOI: 10.52362/jisicom.v9i1.1904

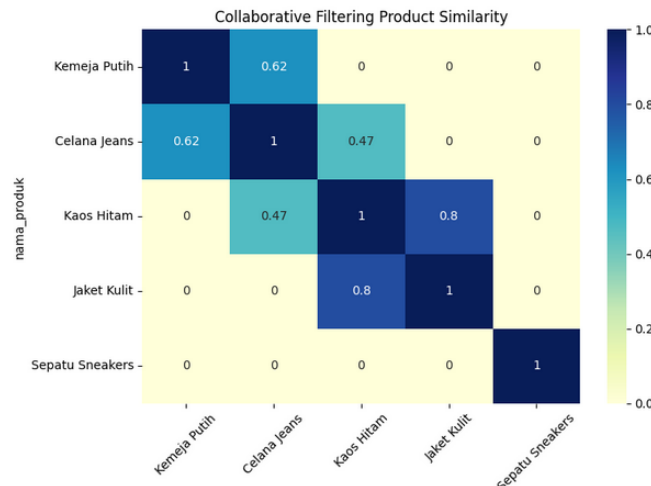
Ciptaan disebarluaskan di bawah [Lisensi Creative Commons Atribusi 4.0 Internasional](https://creativecommons.org/licenses/by/4.0/).



Gambar 14. Heatmap Similarity antar Produk (Content-Based)

Heatmap menggambarkan kemiripan antar produk berdasarkan deskripsi menggunakan Content-Based Filtering dengan TF-IDF dan cosine similarity. Nilai mendekati 1 menunjukkan tingkat kemiripan tinggi. Visualisasi ini membantu memahami cara sistem mengenali kesamaan konten antar produk. Sistem merekomendasikan produk seperti Kaos Hitam, Sepatu Sneakers, dan Celana Jeans karena memiliki karakteristik serupa dengan produk yang disukai pengguna, meskipun belum pernah diberi rating secara langsung, seperti dalam kasus preferensi terhadap pakaian kasual.

4. Visualisasi Collaborative Filtering



Gambar 16. Heatmap Similarity antar Produk (Collaborative Filtering)

Heatmap ini menampilkan kemiripan antar produk berdasarkan pola interaksi pengguna dengan pendekatan Collaborative Filtering. Produk dianggap mirip jika pengguna memberi rating serupa pada keduanya, dihitung melalui korelasi rating. Visualisasi ini berbeda dari pendekatan content-based, karena menekankan preferensi kolektif pengguna. Sistem merekomendasikan produk seperti Kemeja Putih, Celana Jeans, dan Kaos Hitam karena populer di





kalangan pengguna dengan preferensi serupa. Pendekatan ini membantu memahami pola perilaku pengguna dan mendukung rekomendasi yang lebih relevan secara sosial.

IV. KESIMPULAN

Berdasarkan hasil penelitian, sistem rekomendasi e-commerce berbasis pendekatan Collaborative Filtering dan Content-Based Filtering mampu menghasilkan rekomendasi produk yang relevan. Collaborative Filtering bekerja dengan mengidentifikasi kesamaan pola perilaku antar pengguna, sehingga produk yang disukai oleh pengguna serupa dapat direkomendasikan. Sementara itu, Content-Based Filtering menganalisis fitur produk menggunakan TF-IDF dan cosine similarity untuk menemukan kemiripan konten dengan produk yang telah disukai pengguna sebelumnya. Produk seperti Kaos Hitam, Celana Jeans, dan Sepatu Sneakers menjadi contoh yang sering muncul dalam rekomendasi, baik karena popularitas di kalangan pengguna serupa maupun karena kemiripan kontennya. Visualisasi seperti diagram batang dan heatmap turut membantu memperkuat pemahaman terhadap performa sistem dan persepsi pengguna terhadap produk. Pendekatan hybrid yang menggabungkan kedua metode menunjukkan hasil yang lebih seimbang dan efektif, khususnya dalam menangani tantangan cold start dan preferensi pengguna yang beragam. Sistem ini tidak hanya meningkatkan akurasi rekomendasi, tetapi juga mendukung pengalaman belanja yang lebih personal serta strategi pemasaran produk yang lebih tepat sasaran dalam platform e-commerce.

REFERENASI

- [1.] Sari, D. M. (2020). "Sistem Rekomendasi Produk E-Commerce Menggunakan Metode Collaborative Filtering". *Jurnal Informatika Universitas Pamulang*, 5(2), 89–96.
- [2.] Ramadhani, R., & Nugroho, A. S. (2022). "Implementasi Content-Based Filtering untuk Rekomendasi Produk Pakaian pada E-Commerce". *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 9(1), 45–53.
- [3.] Pratama, G., & Hidayat, A. R. (2019). "Sistem Rekomendasi pada Marketplace Menggunakan Metode Hybrid Filtering". *Jurnal RESTI (Re"Kayasa Sistem dan Teknologi Informasi)*, 3(2), 299–306.
- [4.] Hidayatullah, M., & Fauzi, A. (2018). "Pemanfaatan TF-IDF dan Cosine Similarity untuk Sistem Rekomendasi pada Situs E-Commerce". *Jurnal Ilmiah Informatika KOMPUTA*, 7(1), 12–18.
- [5.] Yuliana, S., & Firmansyah, D. (2016). "Sistem Rekomendasi Produk Fashion Berbasis Content-Based Filtering". *Jurnal Teknologi dan Sistem Komputer*, 4(3), 394–400.
- [6.] Setiawan, A. H., & Kurniawan, I. (2021). "Penerapan Collaborative Filtering untuk Rekomendasi Produk di Platform E-Commerce". *Jurnal Ilmu Komputer AMIKOM*, 7(1), 71–78.
- [7.] Kurniawan, R., & Andriani, D. (2020). "Analisis Sistem Rekomendasi dengan Metode Hybrid Filtering untuk E-Commerce Produk Elektronik". *Jurnal Sains dan Informatika*, 6(2), 112–119.
- [8.] Wulandari, F., & Hermawan, D. (2019). "Perbandingan Collaborative Filtering dan Content-Based Filtering untuk Sistem Rekomendasi Buku". *Jurnal Ilmiah Teknik Informatika Komputer (JITIK)*, 5(1), 23–30.
- [9.] Astuti, D., & Prasetyo, H. (2017). "Rancang Bangun Sistem Rekomendasi Menggunakan Hybrid Filtering pada Toko Online". *Jurnal Sisfokom*, 6(2), 151–158.
- [10.] Nugroho, B. S., & Sari, R. N. (2018). "Pengembangan Sistem Rekomendasi Produk Menggunakan Algoritma Content-Based Filtering". *Jurnal Ilmiah Matrik*, 20(2), 85–92.

