

LAPORAN PROYEK
Sistem Rekomendasi Film Menggunakan Pendekatan
Content Based Filtering



Disusun oleh:

1. 12S17004 – Fivin Sadesla Tambunan
2. 12S17026 – Mika Lestari Valentina Manurung
3. 12S17037 – Nita Sophia Winandi Sirait

11S4037 – PEMROSESAN BAHASA ALAMI
FAKULTAS INFORMATIKA DAN TEKNIK ELEKTRO
INSTITUT TEKNOLOGI DEL
2020

DAFTAR ISI

DAFTAR ISI	2
DAFTAR TABEL	3
DAFTAR GAMBAR	4
BAB 1 PENDAHULUAN	5
1.1 Latar Belakang.....	5
1.2 Tujuan.....	6
1.3 Manfaat.....	7
1.4 Ruang Lingkup.....	7
BAB 2 ISL.....	8
2.1 Analisis	8
2.1.1 Analisis Data	8
2.1.2 Analisis Metode.....	11
2.2 Desain	11
2.2.1 Desain Arsitektur Sistem	11
2.2.2 Desain Umum Sistem	12
2.3 Implementasi	13
2.3.1 <i>Data Preparation</i>	13
2.3.2 <i>Content Based Modeling</i>	14
2.4 Evaluasi dan Pembahasan Hasil Eksperimen.....	16
BAB 3 PENUTUP	18
3.1 Pembagian Tugas dan Tanggung Jawab.....	18
3.2 Kesimpulan dan Saran	18
3.2.1 Kesimpulan	19
3.2.2 Saran.....	19
DAFTAR PUSTAKA	20

DAFTAR TABEL

Tabel 3.1. Pembagian Tugas	18
----------------------------------	----

DAFTAR GAMBAR

Gambar 2.1. Movie dengan Rating Terbanyak.....	8
Gambar 2.2. Genre Movie Paling Populer	9
Gambar 2.3. Rata-Rata Rating Setiap Movie	9
Gambar 2.4. Movie dengan Rata-Rata Rating Tertinggi.....	10
Gambar 2.5. Movie dengan Rata-Rata Rating Terendah	10
Gambar 2.6. Desain Arsitektur Sistem	11
Gambar 2.7. Flowchart Desain Umum Sistem	12
Gambar 2.8. Kode Program Merge Dataset.....	13
Gambar 2.9. Dataset yang telah di Merge	13
Gambar 2.10. Kode Program Cek NaN	14
Gambar 2.11. Jumlah NaN Pada Dataset	14
Gambar 2.12. Kode Program Cek NaN	14
Gambar 2.13. Jumlah NaN pada Dataset	14
Gambar 2.14. Kode Program Menentukan TD-IDF	14
Gambar 2.15. Kode Program Membuat Matriks TD-IDF.....	15
Gambar 2.16. Kode Program Menghitung Cosine Similarity	15
Gambar 2.17. Kode Program Membuat Daftar Film Berdasarkan Kemiripan	15
Gambar 2.18. Kode Program Mendapatkan Rekomendasi Film.....	16
Gambar 2.19. Rekomendasi Film.....	16
Gambar 2.20. Kode Program Evaluasi Model	16
Gambar 2.21. Kode Program Menghitung Nilai Evaluasi	17
Gambar 2.22. Hasil Evaluasi	17

BAB 1

PENDAHULUAN

Bab ini menjelaskan latar belakang pemilihan topik proyek, tujuan yang ingin dicapai pada pengerjaan proyek, manfaat pengerjaan proyek, dan ruang lingkup dari proyek.

1.1 Latar Belakang

Natural Language Processing (NLP) merupakan bagian dari sub bidang linguistik, ilmu komputer teknik informasi, dan kecerdasan buatan yang berkaitan dengan interaksi antara komputer dan bahasa manusia (alami). NLP adalah pembuatan program yang memiliki kemampuan untuk memahami bahasa manusia. Tujuan dari pemrograman bahasa alami adalah melakukan proses pembuatan model komputasi dari bahasa sehingga dapat terjadi suatu interaksi antara manusia dengan komputer dengan perantara bahasa alami [1]. Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia. Salah satu topik yang berkaitan dengan NLP adalah Sistem Rekomendasi.

Semakin berkembangnya teknologi maka akan semakin banyaknya informasi yang tersedia. Informasi ini dapat diakses melalui pemanfaatan dari teknologi multifungsi yang dikaji agar lebih efisien dan optimal melalui internet. Saat ini dengan adanya teknologi, semua informasi yang dibutuhkan sudah tersedia di internet dengan berbagai versi. Pada era sekarang, banyak masyarakat yang membutuhkan hiburan untuk menghilangkan rasa penat dari kesibukan sehari-hari, hiburan tersebut dapat diperoleh melalui film. Menurut *British Film Institute* (BFI) film *box office* yang diproduksi terus meningkat setiap tahunnya mulai tahun 2009 hingga tahun 2015 (*British Film Institute*, 2016). Pada tahun 2009 terdapat 503 film yang diproduksi pada tahun tersebut dan sebanyak 759 film diproduksi pada tahun 2015 (*British Film Institute*, 2016). Berdasarkan hal tersebut maka dapat disimpulkan bahwa banyaknya film yang tersedia di internet. Oleh karena itu, dibutuhkan sebuah sistem yang dapat membantu memberikan informasi sesuai dengan keinginan pengguna. Sistem tersebut sering mengurangi jumlah informasi yang terlalu banyak. Sistem rekomendasi adalah suatu teknologi yang didesain bagi pengguna agar memudahkan dalam menemukan suatu data yang sesuai dengan profil pengguna secara cepat dan dapat disebut dengan sistem rekomendasi.

Sistem rekomendasi memiliki kemampuan untuk melakukan prediksi tentang sebuah item yang mungkin disukai pengguna berdasarkan informasi yang didapat dari pengguna tersebut. Item tersebut atau dalam hal ini film dapat disarankan berdasarkan rating film yang diberikan pengguna atau berdasarkan pengguna lain yang memiliki kebiasaan yang mirip. Terdapat beberapa algoritma yang dapat dipakai untuk membangun Sistem rekomendasi dengan kelebihan dan kekurangan masing-masing. *Content Based Recommendation* yang dikenal dengan cara memanfaatkan informasi item untuk direkomendasikan kepada *user* yang berkaitan dengan riwayat *user* sebelumnya, *Collaborative Filtering* dengan memanfaatkan transaksi dari suatu item yang didasarkan pada perilaku *user* dan *Hybrid Recommendation* yang merupakan gabungan dari *Content Based Filtering* dan *Collaborative Filtering*.

Pada tahun 2018, Imam melakukan penelitian Sistem rekomendasi film menggunakan *Collaborative Filtering*. Pada penelitian ini dapat diketahui bahwa metode *Collaborative Filtering* masih memiliki kekurangan yaitu item yang baru masuk kedalam sistem dan belum di-*rating* sama sekali oleh pengguna tidak akan pernah direkomendasikan kepada pengguna. Hal ini diakibatkan karena pendekatan *collaborative filtering* melakukan prediksi berdasarkan rating yang diberikan pengguna pada item [2].

Penelitian ini menggunakan pendekatan *content based filtering* karena pendekatan content based filtering memiliki kemampuan merekomendasikan item (contoh: film, lagu, artikel dll) yang sifatnya baru bagi *user*, karena prinsip kerjanya yaitu dengan melihat diskripsi *content* yang dikandung oleh item yang pernah diberi nilai rating tinggi sebelumnya oleh *user*. Berdasarkan data training, *user* model menginduksikan kemungkinan sistem penyaringan untuk mengklasifikasikan item yang tidak terlihat masuk ke dalam kelas positif c (relevan untuk *user*) atau ke dalam kelas negatif c (tidak relevan untuk *user*) [3].

1.2 Tujuan

Tujuan yang diharapkan dari pengerjaan proyek ini adalah menghasilkan sebuah Sistem rekomendasi yang dapat membantu masyarakat dapat menyaksikan film yang sama berdasarkan *genre* yang sesuai dengan *profile*.

1.3 Manfaat

Manfaat yang diharapkan dari penelitian ini yaitu: menambah pengetahuan tentang pembuatan Sistem rekomendasi yang menggunakan pendekatan *content based filtering*.

1.4 Ruang Lingkup

Ruang lingkup pada Sistem rekomendasi Film dengan pendekatan *Content Based Filtering* adalah sebagai berikut:

1. Menggunakan konsep bahasa pemrograman Python.
2. Rekomendasi yang diberi hanya berdasarkan penjelasan item dan profil dari penggunanya.

BAB 2

ISI

Bab ini menjelaskan isi meliputi meliputi analisis data dan metode, desain, implementasi dan hasil evaluasi dalam penyelesaian proyek.

2.1 Analisis

Pada sub bab ini dijelaskan analisis data dan metode yang digunakan untuk memberikan rekomendasi film bagi penonton film.

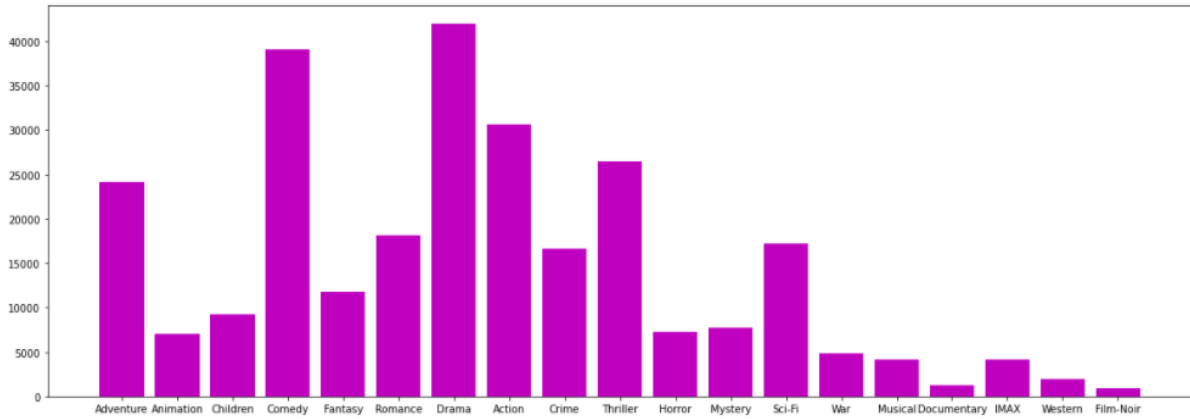
2.1.1 Analisis Data

Dataset yang digunakan pada proyek ini didapatkan dari URL <https://grouplens.org/datasets/movielens/>. *Dataset* terdiri dari 100836 baris data dan 6 atribut yaitu movieId, title, genres, *userId*, rating, dan timestamp. Pada *dataset*, 10 *movie* yang paling banyak di rating dapat dilihat pada Gambar 2.1.

title	
Forrest Gump (1994)	329
Shawshank Redemption, The (1994)	317
Pulp Fiction (1994)	307
Silence of the Lambs, The (1991)	279
Matrix, The (1999)	278
Star Wars: Episode IV - A New Hope (1977)	251
Jurassic Park (1993)	238
Braveheart (1995)	237
Terminator 2: Judgment Day (1991)	224
Schindler's List (1993)	220
dtype: int64	

Gambar 2.1. *Movie* dengan Rating Terbanyak

Genre yang paling populer berdasarkan jumlah *movie* yang dirilis dapat dilihat pada Gambar 2.2.



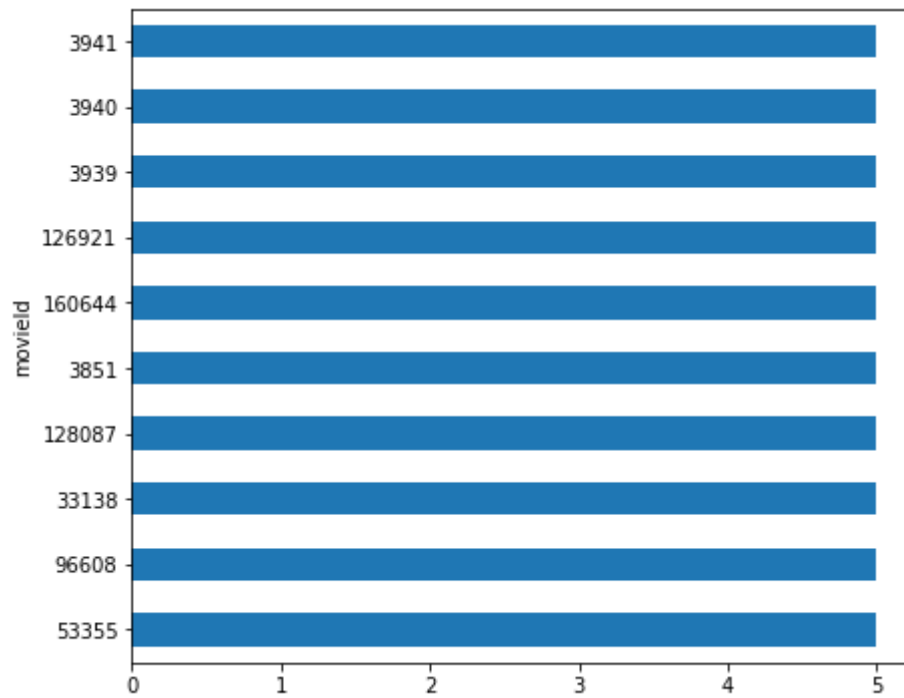
Gambar 2.2. Genre Movie Paling Populer

Kemudian, dilakukan analisis terhadap film dengan rating rata-rata tertinggi. Rata-rata rating untuk setiap film adalah sebagai berikut :

	userId	rating
	mean	mean
movieId		
1	306.530233	3.920930
2	329.554545	3.431818
3	283.596154	3.259615
4	219.857143	2.357143
5	299.571429	3.071429
6	309.647059	3.946078
7	279.000000	3.185185
8	258.250000	2.875000
9	362.937500	3.125000
10	315.159091	3.496212

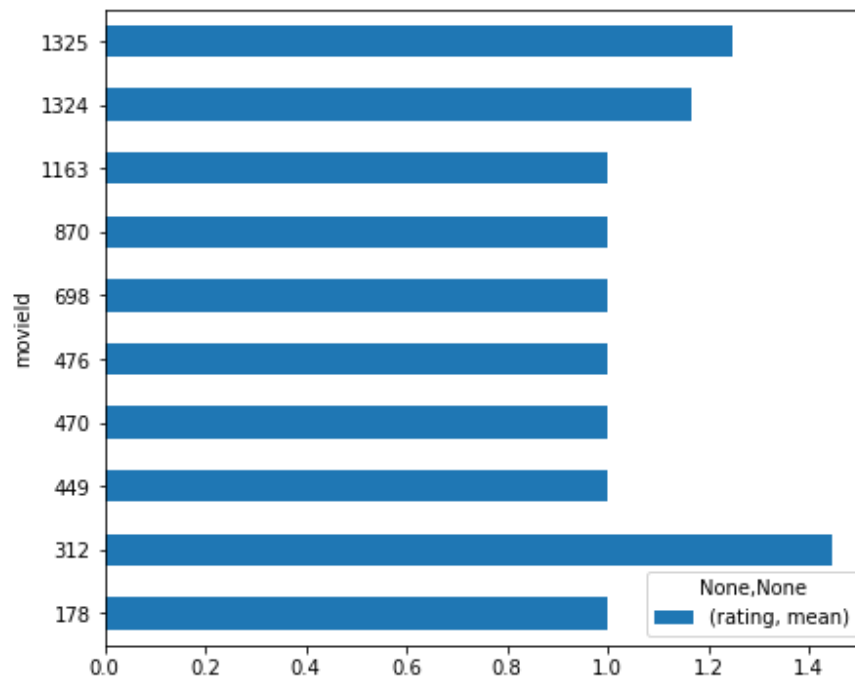
Gambar 2.3. Rata-Rata Rating Setiap *Movie*

Film dengan rata-rata rating tertinggi dapat dilihat pada Gambar 2.4.



Gambar 2.4. Movie dengan Rata-Rata Rating Tertinggi

Selanjutnya untuk film dengan rating rata-rata terendah dapat dilihat pada Gambar 2.5.



Gambar 2.5. Movie dengan Rata-Rata Rating Terendah

2.1.2 Analisis Metode

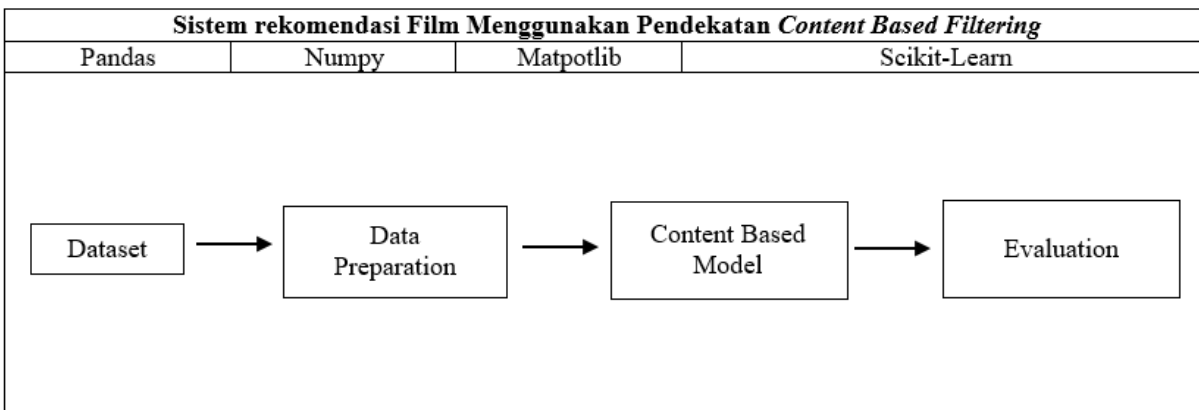
Sistem rekomendasi memiliki beberapa metode, seperti *Content Based Filtering*, *Collaborative Filtering* dan *Hybrid Recommendation*. Metode yang digunakan pada Sistem rekomendasi film ini adalah *Content-based recommendation*. *Content-based recommendation* adalah suatu sistem yang akan merekomendasikan sebuah item kepada *user* yang didasari oleh deskripsi dari suatu item dan profil dari ketertarikan *user*. *Content based filtering* bekerja menggunakan data yang disediakan oleh *user*, semakin banyak data yang di-input maka rekomendasi yang dihasilkan akan lebih akurat.

2.2 Desain

Pada subbab ini dijelaskan mengenai perancangan terhadap sistem yang akan diimplementasikan oleh penulis. Desain yang dimaksud terdiri dari desain arsitektur sistem yang dibangun, desain umum proyek, desain pre-pemrosesan data, desain model, dan desain eksperimen.

2.2.1 Desain Arsitektur Sistem

Pada bagian ini dijelaskan gambaran arsitektur sistem yang digunakan dalam pengerjaan proyek. Dalam melakukan tahapan pembangunan sistem digunakan arsitektur seperti pada Gambar 2.6.



Gambar 2.6. Desain Arsitektur Sistem

Berikut dijelaskan beberapa *library* yang akan digunakan pada pengerjaan proyek:

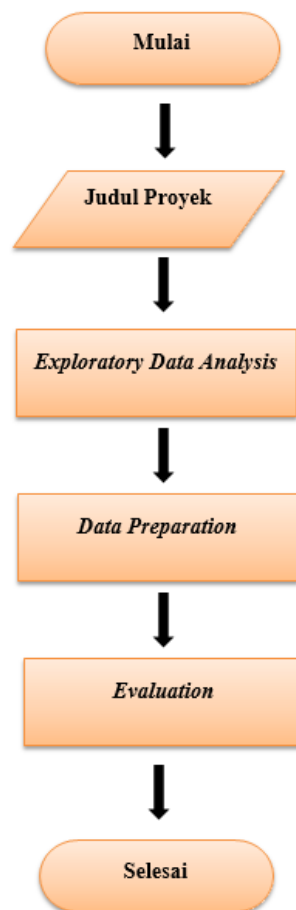
1. *Library* Pandas, merupakan *library* yang dirancang untuk melakukan analisis data dalam keuangan, statistik, ilmu sosial dan *engineering*. Pandas dapat bekerja dengan baik pada data

yang tidak lengkap, berantakan, dan tidak diberi label serta menyediakan *tools* untuk membentuk, menyatukan, dan memotong *dataset*.

2. *Library* Numpy, merupakan *library* yang digunakan untuk berbagai fungsi statistik dan matematika. Numpy merupakan *library* dasar untuk komputasi ilmiah pada Python.
3. *Library* Matplotlib, digunakan untuk menyediakan ekstensi matematika yang disediakan oleh *library* Numpy dan digunakan untuk visualisasi data.
4. *Library* Scikit-Learn, merupakan *library* machine learning yang digunakan dalam melakukan *data mining* atau analisis data, seperti *clustering*, *regression*, dan *classification*.

2.2.2 Desain Umum Sistem

Flowchart untuk desain umum sistem pada sistem rekomendasi film menggunakan *content based modeling* dapat dilihat pada Gambar 2.7.



Gambar 2.7. *Flowchart* Desain Umum Sistem

Sistem yang dibangun pada proyek ini adalah *movie recommendation* dengan menggunakan pendekatan *content based filtering*. Tahapan yang dilakukan setelah menganalisis data (*exploratory data analysis*) pada *movie recommendation* terbagi atas 3 fungsi utama, yaitu: (1) *data preprocessing*, (2) Membangun Model, dan (3) melakukan evaluasi terhadap hasil rekomendasi tersebut.

2.3 Implementasi

Pada subbab ini dijelaskan mengenai implementasi proyek berupa *data preparation* dan *content based modeling*.

2.3.1 Data Preparation

Pada *data preparation*, dilakukan penggabungan *dataset* yaitu *dataset* movies.csv yang berisi movieId, title, dan genres digabungkan dengan *dataset* ratings.csv yang berisi userId, movieId, dan rating. Program kode yang digunakan untuk menggabungkan dataset dapat dilihat pada Gambar 2.8.

```
data3 = pd.merge(data1, data2)
data3.head()
```

Gambar 2.8. Kode Program Merge Dataset

Data 1 berisi *dataset* movies.csv dan data2 berisi *dataset* ratings.csv. Hasil dari *dataset* yang telah digabungkan dapat dilihat pada Gambar 2.9.

	movieId	title	genres	userId	rating	timestamp
0	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	1	4.0	964982703
1	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	5	4.0	847434962
2	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	7	4.5	1106635946
3	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	15	2.5	1510577970
4	1	Toy Story (1995)	Adventure Animation Children Comedy Fantasy	17	4.5	1305696483

Gambar 2.9. Dataset yang telah di Merge

Untuk mengecek apakah terdapat NaN (nilai null) pada dataset digunakan kode pada Gambar 2.10.

```
print("Jumlah NaN pada data : ", sum(data3.isnull().any()))
```

Gambar 2.10. Kode Program Cek NaN

Hasil pada Gambar 2.10 adalah sebagai berikut :

```
Jumlah NaN pada data : 0
```

Gambar 2.11. Jumlah NaN Pada Dataset

Cara lain yang dapat digunakan untuk mengecek nilai null pada *dataset* adalah :

```
data3.isnull().sum()
```

Gambar 2.12. Kode Program Cek NaN

Hasil untuk Gambar 2.12 adalah sebagai berikut :

```
movieId      0
title        0
genres       0
userId       0
rating       0
timestamp    0
dtype: int64
```

Gambar 2.13. Jumlah NaN pada Dataset

Pada gambar diatas, untuk setiap atribut seperti *movieId*, *title*, *genres*, *userId*, *rating*, dan *timestamp* tidak terdapat nilai null (NaN).

2.3.2 Content Based Modeling

Konsep *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) digunakan dalam content based recommender. TF/IDF digunakan untuk menentukan kepentingan relatif dari film. Untuk menentukan TF-IDF dapat dilihat pada Gambar 2.14.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel
# Menentukan TF-IDF Vectorizer Object.
tfidf_data1_genres = TfidfVectorizer(token_pattern = '[a-zA-Z0-9\-\-]+')
# Mengganti NaN dengan string kosong
data1['genres'] = data1['genres'].replace(to_replace="(no genres listed)",
value="")
```

Gambar 2.14. Kode Program Menentukan TD-IDF

Kemudian, untuk membuat TF-IDF yang diperlukan dapat dengan menyesuaikan dan mengubah data. Program kode yang digunakan adalah :

```
# Membuat matriks TF-IDF yang diperlukan dengan menyesuaikan dan mengubah data

tfidf_data1_genres_matrix =
tfidf_data1_genres.fit_transform(data1['genres'])
print(tfidf_data1_genres.get_feature_names())
```

Gambar 2.15. Kode Program Membuat Matriks TD-IDF

Kemudian, dilakukan penghitungan *cosine similarity* yaitu :

```
# Menghitung cosine similarity
cosine_sim_movies = linear_kernel(tfidf_data1_genres_matrix,
tfidf_data1_genres_matrix)
print(cosine_sim_movies)
```

Gambar 2.16. Kode Program Menghitung Cosine Similarity

Selanjutnya, dibuat fungsi untuk membuat daftar film berdasarkan kemiripannya dengan film lain. Misalkan terdapat film A, maka akan direkomendasikan daftar 3 film yang memiliki kemiripan paling tinggi dengan film A berdasarkan *cosine similarity* matrix yang telah dibuat.

```
# fungsi untuk mengambil judul film sebagai input
def get_recommendations(movie_title,
cosine_sim_movies=cosine_sim_movies):

    # Mengambil indeks film yang sesuai dengan judul film
    idx_movie = data1.loc[data1['title'].isin([movie_title])]
    idx_movie = idx_movie.index

    # Mengambil similarity scores pasangan dari semua film dengan film itu
    sim_scores_movies = list(enumerate(cosine_sim_movies[idx_movie][0]))

    # Mengurutkan film berdasarkan similarity scores
    sim_scores_movies = sorted(sim_scores_movies, key=lambda x: x[1],
reverse=True)

    # Mengambil skor 10 film paling mirip
    sim_scores_movies = sim_scores_movies[1:3]

    # Mendapatkan indeks film
    movie_indices = [i[0] for i in sim_scores_movies]

    # Mendapatkan 2 film paling mirip
    return data1['title'].iloc[movie_indices]
```

Gambar 2.17. Kode Program Membuat Daftar Film Berdasarkan Kemiripan

Selanjutnya untuk mendapatkan rekomendasi film dapat dilihat pada Gambar 2.18.

```
get_recommendations("Flint (2017)")
```

Gambar 2.18. Kode Program Mendapatkan Rekomendasi Film

Pada kode program dapat dilihat rekomendasi film yang mirip dengan film Flint 2017. Rekomendasi yang dihasilkan adalah sebagai berikut :

```
25          Othello (1995)
30  Dangerous Minds (1995)
Name: title, dtype: object
```

Gambar 2.19. Rekomendasi Film

Rekomendasi film yang memiliki kemiripan dengan film Flint (2017) adalah film Othello (1995) dan film Dangerous Minds (1995).

2.4 Evaluasi dan Pembahasan Hasil Eksperimen

Algoritma KNN termasuk kedalam algoritma *lazy learning* yang sering diimplementasikan. Metode ini akan digunakan untuk mengukur performa sistem. Model ini dievaluasi untuk melihat kecocokan antara genre dengan genre film yang sudah ditonton oleh *user*. Langkah-langkah algoritma KNN:

1. Menentukan parameter *k* (jumlah tetangga terdekat), pada model evaluasi proyek yang digunakan *k*=5
2. Hitung jarak data latih dengan semua data uji
3. Urutkan jarak tersebut berdasarkan nilai yang terkecil sejumlah *k*.
4. Tentukan kelompok data uji berdasarkan label mayoritas pada *k*.

Kode Program untuk evaluasi model dapat dilihat pada Gambar 2.20.

```
from sklearn.neighbors import KNeighborsClassifier
def get_movie_label(movie_id):
    classifier = KNeighborsClassifier(n_neighbors=5)
    x= tfidf_data1_genres_matrix
    y = data1.iloc[:, -1]
    classifier.fit(x, y)
    y_pred = classifier.predict(tfidf_data1_genres_matrix[movie_id])
    return y_pred
```

Gambar 2.20. Kode Program Evaluasi Model

Untuk menghitung nilai evaluasi berupa *hit* ($true_count/total$) dan *fault* ($false_count/total$) adalah sebagai berikut :


```

true_count = 0
false_count = 0
def evaluate_content_based_model():
    """
    Evaluate content based model.
    """
    for key, columns in data1.iterrows():

        movies_recommended_by_model =
get_recommendations(columns["title"])
        predicted_genres =
get_movie_label(movies_recommended_by_model.index)
        for predicted_genre in predicted_genres:
            global true_count, false_count
            if predicted_genre == columns["genres"]:
                true_count = true_count+1
            else:
#                 print(columns["genres"])
#                 print(predicted_genre)
                false_count = false_count +1
evaluate_content_based_model()
total = true_count + false_count
print("Hit:" + str(true_count/total))
print("Fault:" + str(false_count/total))

```

Gambar 2.21. Kode Program Menghitung Nilai Evaluasi

Metode *Cosine Similarity* merupakan metode yang digunakan untuk menghitung *similarity* (tingkat kesamaan) antar dua buah objek. Secara umum penghitungan metode ini didasarkan pada vector space similarity measure. Metode *cosine similarity* ini menghitung *similarity* antara dua buah objek (misalkan D1 dan D2) yang dinyatakan dalam dua buah vektor dengan menggunakan *keywords* (kata kunci) dari sebuah dokumen sebagai ukuran [4].

$$CosSim \quad (d_i, q_i) = \frac{q_i \cdot d_i}{|q_i| |d_i|} = \frac{\sum_{j=1}^t (q_{ij} \cdot d_{ij})}{\sqrt{\sum_{j=1}^t (q_{ij})^2 \cdot \sum_{j=1}^t (d_{ij})^2}}$$

Keterangan :

q_{ij} = bobot istilah j pada dokumen i= $tf_{ij} \cdot idf_j$

d_{ij} = bobot istilah j pada dokumen i= $tf_{ij} \cdot idf_j$

Dari hasil evaluasi model *content based* diperoleh *output* berupa :

Hit:0.9325087251077807
Fault:0.06749127489221926

Gambar 2.22. Hasil Evaluasi

Dapat disimpulkan bahwa nilai *true* lebih besar daripada nilai *false* sehingga model yang digunakan cukup baik.

BAB 3

PENUTUP

Bab ini menjelaskan pembagian tugas kelompok serta kesimpulan dan saran selama pengerjaan proyek.

3.1 Pembagian Tugas dan Tanggung Jawab

Berikut merupakan tabel pembagian tugas anggota kelompok :

Tabel 3.1. Pembagian Tugas

No	Anggota kelompok	Peran	Tugas
1	Fivin Sadesla Tambunan	<i>Team leader</i>	<ul style="list-style-type: none">- Bertanggung jawab membuat <i>schedule</i> kegiatan atau jadwal kegiatan pekerjaan.- Memonitor atau memantau progress pekerjaan proyek- Bertanggung jawab terhadap Pemberi Pekerjaan yang berkaitan terhadap kegiatan tim pelaksana pekerjaan.
		<i>System Analist</i>	<ul style="list-style-type: none">- Mengidentifikasi permasalahan- Membuat diagram alir, dan spesifikasi yang akan digunakan oleh programmer- Menganalisis hasil yang sudah dikerjakan
2	Mika Lestari Valentina Manurung	<i>System Analist</i>	<ul style="list-style-type: none">- Mengidentifikasi permasalahan- Membuat diagram alir, dan spesifikasi yang akan digunakan oleh programmer- Menganalisis hasil yang sudah dikerjakan
		<i>Programmer</i>	<ul style="list-style-type: none">- Mengimplementasikan code untuk membangun sistem- Menguji sistem yang telah dibuat, agar aplikasi yang dibuat bisa bermanfaat untuk pengguna
3	Nita Sophia Winandi Sirait	<i>System Analist</i>	<ul style="list-style-type: none">- Mengidentifikasi permasalahan- Membuat diagram alir, dan spesifikasi yang akan digunakan oleh <i>programmer</i>- Menganalisis hasil yang sudah dikerjakan
		<i>Programmer</i>	<ul style="list-style-type: none">- Mengimplementasikan <i>code</i> untuk membangun sistem- Menguji sistem yang telah dibuat, agar aplikasi yang dibuat bisa bermanfaat untuk pengguna

3.2 Kesimpulan dan Saran

Subbab ini berisi penjelasan terkait kesimpulan dan saran terhadap proyek yang telah dikerjakan.

3.2.1 Kesimpulan

Penggunaan *recommendation system* pada *movie recommendation* dengan metode *content based filtering* dapat menemukan *movie* yang memiliki kesamaan dari sisi deskripsi yang ditampilkan pada *dataset*. Kemiripan ini diperoleh melalui konsep *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF) yang digunakan dalam *content based recommender* untuk menentukan kepentingan relatif dari film. Beberapa fungsi akan didefinisikan untuk membuat daftar film berdasarkan kemiripannya dengan film lain berdasarkan *cosine similarity matrix* yang telah dibuat. Setelah dievaluasi menggunakan model KNN, penggunaan metode ini cukup baik dengan jumlah akurasi 0.9325087251077807.

3.2.2 Saran

Pendekatan *content based filtering* hanya mengandalkan riwayat *item* yang pernah di rating. Keterbatasan metode ini yaitu ketika *user* meminta rekomendasi suatu item yang memiliki jenis konten yang berbeda dengan item yang sudah pernah dipilih maka rekomendasi yang diharapkan tidak muncul. Dibutuhkan suatu pengembangan sistem yang mampu memanfaatkan rating *user* lain untuk memprediksi item yang mungkin akan disukai oleh seorang *user*.

DAFTAR PUSTAKA

- [1] B. Subaeki dan N. K. Wangsanegara, "IMPLEMENTASI NATURAL LANGUAGE PROCESSING DALAM PENGUKURAN KETEPATAN EJAAN YANG DISEMPURNAKAN (EYD) PADA ABSTRAK SKRIPSI MENGGUNAKAN ALGORITMA FUZZY LOGIC," *JURNALTEKNIKINFORMATIKA*, vol. 8, no. 2, p. 2, 2015.
- [2] I. Dwicahya, "Perbandingan sistem rekomendasi film metode user-based dan item based collaborative filtering," Sanata Dharma University, Yogyakarta, 2018.
- [3] R. Oktorina, W. Maharani dan Y. Firdaus, "CONTENT BASED RECOMMENDER SYSTEM MENGGUNAKAN ALGORITMA APRIORI," *Konferensi Nasional Sistem dan Informatika*, p. 125, 2010.
- [4] B. Long, J. Bian, A. Dong dan Y. Cang, "Enhancing Product Search by Best-Selling Prediction in E-commerce," *international conference on Information and knowledge management*, 2015.