

# Sistem Rekomendasi Film Berbasis Website dengan Metode Prototype Menggunakan Algoritma K-Nearest Neighbors (KNN)

Alifia Rahma Fitrianti<sup>1</sup>, Asih Rohmani<sup>2</sup>, Widjanarto<sup>3</sup>

Universitas Dian Nuswantoro, Jalan Imam Bonjol No. 207 Semarang, 50131

e-mail: <sup>1</sup>alifia49@gmail.com, <sup>2</sup>aseharsoyo@dsn.dinus.ac.id

Diterima: 23 Oktober 2020; Direvisi: 18 Nopember 2020; Disetujui: 23 Nopember 2020

## Abstrak

Industry perfilman dunia merupakan salah satu industry yang tidak terpengaruh dengan maraknya hiburan digital seperti munculnya media social, program televisi yang beragam dan game. Industry film yang terus melakukan produksi ini semakin menambah informasi film yang melimpah di internet. Kondisi ini justru membuat para penikmat film menjadi kebingungan ketika harus memilih film kesukaannya. Sistem rekomendasi menyediakan informasi berdasarkan interaksi pengguna dan item yang telah terekam sebelumnya. Pendekatan content-based filtering merupakan salah satu pendekatan pada sistem rekomendasi yang mampu memberikan rekomendasi item berdasarkan fitur atau atribut yang memiliki kemiripan dengan item yang telah berinteraksi dengan pengguna. Penelitian ini akan membahas pembangunan sistem rekomendasi dengan metode prototype menggunakan algoritma K-Nearest Neighbors (KNN) serta pendekatan content-based filtering. K-Nearest Neighbors digunakan untuk mengklasifikasikan objek baru berdasarkan atribut dan sampel dari data training. Metode pembangunan sistem prototype memiliki beberapa tahapan yaitu communication, quick plan and modelling design, construction of prototype, serta deployment delivery and feedback. Berdasarkan hasil pengujian, sistem yang dibangun dapat memberikan rekomendasi film dengan nilai presisi 45% dan recall 60%.

**Kata kunci:** Sistem rekomendasi, content-based filtering, k-nearest neighbors, movie

## Abstract

The world film industry is one industry that is not affected by the rise of digital entertainment, such as the emergence of social media, various television programs and games. The film industry, which continues to produce, adds to the abundant film information on the internet. This condition actually makes film lovers confused when they have to choose their favorite film. The recommendation system provides information based on user interactions and previously recorded items. The content-based filtering approach is an approach to a recommendation system that is able to provide item recommendations based on features or attributes that have similarities with items that users have interacted with. This study will discuss the development of a recommendation system with a prototype method using the K-Nearest Neighbors (KNN) algorithm and a content-based filtering approach. K-Nearest Neighbors is used to classify new objects based on attributes and samples from training data. The prototype system development method has several stages, namely communication, quick plan and modeling design, construction of the prototype, and deployment delivery and feedback. Based on the test results, the system built can provide film recommendations with precision value 45% and recall 60%.

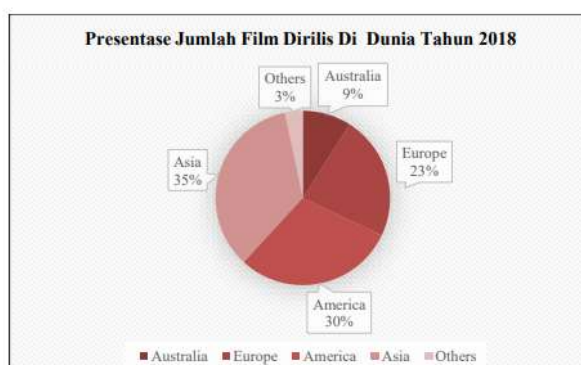
**Keywords:** Recommendation systems, content-based filtering, k-nearest neighbors, movie

---

## 1. PENDAHULUAN

Salah satu hiburan yang masih banyak disukai oleh masyarakat adalah film layar lebar yang biasa diputar di bioskop. Industri perfilman layar lebar termasuk industri yang masih kuat bertahan ditengah maraknya kemunculan media-media hiburan lain seperti media sosial, keberagaman program televisi dan game. Hal ini mungkin karena film merupakan sebuah media yang dapat menuangkan realitas kehidupan ke dalam sebuah layar lebar [1].

Film merupakan sebuah karya gambar bergerak yang dapat dilihat serta didengar, dimana dalam pembuatannya didasarkan pada asas sinematografi yang direkam oleh teknologi perekam seperti piringan video dan pita video. Berbagai film diproduksi dengan menggunakan variasi teknologi dan teknik sinematik sesuai genre. Proses produksi film berlangsung di seluruh dunia dan kian berkembang setiap tahunnya [2]. Film juga menjadi cerminan budaya bangsa dan media representasi kondisi tatanan sosial suatu bangsa, sehingga mengakibatkan munculnya berbagai macam industri film berdasarkan wilayah produksi. Industri film berkembang hingga seluruh dunia, seperti di Amerika Serikat, Eropa, Asia, China, Australia, dan global Selatan [3].



Gambar 1. Persentase jumlah film dirilis di berbagai wilayah di dunia tahun 2018

Gambar 1 menunjukkan jumlah film yang dirilis di berbagai wilayah di dunia pada tahun 2018. Pada tahun tersebut, industri film negara Amerika Serikat atau yang biasa dikenal sebagai sinema Hollywood merilis film sebanyak 225 judul film, sinema Asia merilis film sebanyak 263 judul film, sinema Eropa merilis 174 judul film, sinema Australia merilis 62 judul film dan sisanya adalah sinema di negara-negara lain di dunia [3].

Banyaknya produksi film dari berbagai wilayah negara ini tentunya semakin menambah informasi sinema yang beredar di internet. Hal ini kadang menimbulkan kebingungan bagi seseorang untuk memilih film yang disukainya. Ada beberapa motivasi seseorang datang ke bioskop dan melihat sebuah film. Sebagian orang datang ke bioskop dan memilih untuk menonton sebuah judul film tertentu karena dia memang menyukai film tersebut, akan tetapi sebagian lagi hanya karena membutuhkan hiburan keluar rumah, sehingga begitu sampai di bioskop dia memilih film sembarang untuk ditonton. Menurut survey, sebagian besar penikmat film layar lebar menyukai film yang diadaptasi dari sebuah karya sastra seperti novel [2].

Banyaknya informasi film yang tersedia di internet seharusnya bermanfaat dan memberikan kemudahan bagi masyarakat, akan tetapi pada kenyataannya, banyaknya pilihan film yang ada membuat seseorang bingung memilih mau menonton yang mana, sehingga kadang mereka terpaksa memilih dengan acak. Begitu pula bagi perusahaan film maupun perusahaan penyedia streaming film online (*video-on-demand streaming video*), sulit bagi perusahaan untuk membuat konten yang dimiliki, "*stand out*" dari banyaknya film yang ada. Tidak mudah bagi perusahaan untuk mengetahui keinginan, selera, dan kesukaan calon penonton terhadap suatu film yang ingin dilihat. Perusahaan memerlukan suatu terobosan untuk dapat memberikan akses konten film yang bagus dan cocok bagi penonton. Penonton seringkali tidak memilih satupun dari

pilihan konten yang tersedia dan membatalkan keinginannya untuk menonton sebuah film [3]. *Consumer research* Netflix, perusahaan Amerika di bidang *entertainment* dalam *video on-demand streaming video*, menyebutkan bahwa pengguna Netflix kehilangan ketertarikan setelah kurang-lebih 60 hingga 90 detik setelah memulai memilih, setelah melihat sekitar 10 hingga 20 judul film dimana tiga dilihat dengan detailnya [3]. Jika pengguna batal memilih suatu film untuk ditonton, hal tersebut dapat mengakibatkan kerugian bagi perusahaan industri film maupun penyedia layanan *streaming* film online. Namun sebaliknya, bila perusahaan mampu memberikan dan menampilkan pilihan film sesuai selera pengguna, maka akan meningkatkan keuntungan, karena pengguna akan tetap melanjutkan menonton film dan terus berada dalam “lingkungannya”. Oleh karena itu diperlukan suatu sistem yang dapat memberikan saran mengenai film yang sesuai dengan keinginan atau selera pengguna. Sistem rekomendasi merupakan suatu alat dan teknik yang menyediakan saran-saran yang berguna bagi pengguna. Dalam beberapa tahun terakhir, ketertarikan terhadap sistem rekomendasi telah meningkat cukup pesat [2]. Beberapa perusahaan besar seperti Amazon, AirBnB, Spotify, Google, dan Netflix, mengembangkan mesin rekomendasi, mereka berfokus pada data secara intensif sehingga dapat memanfaatkannya pada *data science*. Perusahaan tersebut menggunakan sistem rekomendasi untuk menyaring jutaan konten guna menemukan saran yang tepat untuk pengguna. *Content-based filtering* merupakan pendekatan sistem rekomendasi yang memberikan rekomendasi item yang mirip dengan yang disukai pengguna di waktu yang lalu, dimana kemiripan suatu item dihitung berdasarkan fitur atau atribut yang berkaitan dengan item yang dibandingkan [2]. Contohnya seperti bila seorang pengguna memberikan *rating* terhadap film yang memiliki *genre* horor, maka sistem akan memberikan rekomendasi film lain yang memiliki *genre* yang sama. Sistem rekomendasi dapat menggunakan teknik data mining dalam implementasinya, salah satunya adalah klasifikasi dengan algoritma *K-Nearest Neighbors* (KNN). *K-Nearest Neighbors* digunakan dengan mencari kelompok *k* objek dalam data training paling mirip dengan objek suatu data baru (data testing), dimana hasil dari *query instance* yang baru algoritma ini diklasifikasikan berdasarkan mayoritas dari kategori pada KNN [4]. Fokus pada penelitian ini adalah membangun sistem rekomendasi film berbasis web dengan metode prototype menggunakan algoritma *K-Nearest Neighbors*.

## 2. METODE PENELITIAN

Penelitian ini akan dilakukan dengan melaksanakan beberapa tahap-tahap penelitian yaitu pengumpulan data, pengolahan data, dan evaluasi. Pengumpulan data dilakukan untuk mendapatkan data-data yang diperlukan dalam penelitian, terdapat tiga cara yang dilakukan dalam pengumpulan data yaitu wawancara, observasi dan studi literatur. Setelah itu dilakukan pengumpulan dataset untuk selanjutnya dapat diolah. Pertama dilakukan *data preprocessing* yang meliputi *data cleaning* dan *data selection*, kemudian dilakukan pemisahan data menjadi *data training* dan *data testing*. Proses klasifikasi dilakukan menggunakan algoritma *K-Nearest Neighbors*, dan selanjutnya evaluasi kinerja algoritma diukur dengan menggunakan *confusion matrix* untuk mendapatkan nilai *precision* dan *recall*. Pengembangan sistem menggunakan metode prototype atau biasa dikenal dengan *prototyping*. Metode prototype merupakan salah satu metode dalam pengembangan software berupa model fisik kerja sistem yang berfungsi sebagai versi awal dari suatu sistem.

Dataset yang digunakan dalam penelitian ini diunduh dari situs penyedia data *Kaggle*. Dataset yang didapatkan berformat *.csv* (*comma separated values*). File film tersebut berisi data-data film yang memiliki atribut: *movieid*, *title*, *genres*, *imdbid*, *plot*, *popularity*, *poster*, *length*, dan *rate*.

Pendekatan *content-based filtering* dirancang untuk memanfaatkan deskripsi rangkaian atribut yang dimiliki oleh item. Pendekatan ini mencoba mencocokkan pengguna terhadap item yang mirip dengan item yang disukai di masa lampau [5]. Kemiripan tidak harus didasarkan pada korelasi peringkat pengguna lain, namun berdasarkan atribut dari item yang disukai pengguna.

*Content-based* sebagian besar berfokus pada peringkat pengguna sendiri dan atribut dari item yang disukai. Metode *content-based* memungkinkan untuk memberikan rekomendasi meskipun untuk item baru karena metode ini dapat memanfaatkan atribut dari item baru dan digunakan guna membuat prediksi. Pada level paling dasar, terdapat dua sumber data yang digunakan dalam *content based* yaitu deskripsi dari beragam item dan profil pengguna. Profil pengguna berkaitan dengan atribut dari berbagai item terhadap rating pengguna [5]. Selain itu juga terdapat tiga komponen utama dalam *content-based*, yaitu : 1) *Preprocessing* dan ekstraksi fitur, contohnya seperti penjual buku memiliki deskripsi buku dan kata kunci yang mendeskripsikan isi konten, judul, author, penerbit, dan tahun terbit, deskripsi tersebut dapat dikonversi menjadi *set* kata kunci; 2) Pembelajaran profil pengguna, yaitu membangun profil pengguna dengan memanfaatkan data pengguna. Profil pengguna berisi kumpulan informasi selera pengguna maupun kebutuhannya; 3) *Filtering* dan rekomendasi, yaitu memanfaatkan profil pengguna untuk menghasilkan rekomendasi yang sesuai dengan profil pengguna.

*Data preprocessing* merupakan tahapan untuk mengolah data mentah menjadi data yang lebih berkualitas. Dalam tahap ini akan dilakukan beberapa tahapan seperti *data cleaning* dan *data selection*. *Data cleaning* merupakan proses pembersihan terhadap data yang *error* maupun tidak akurat, sedangkan *data selection* adalah proses yang dilakukan dengan mempertahankan data yang dibutuhkan serta membuang atribut yang tidak diperlukan dalam pengolahan data. Selanjutnya data dibagi menjadi *data training* dan *data testing* untuk dapat digunakan pada proses klasifikasi. Pemisahan data dilakukan dengan rasio 80:20, *data training* 80% dan *data testing* 20%. Pada penelitian ini digunakan algoritma *K-Nearest Neighbors* (KNN) untuk proses klasifikasi. *K-Nearest Neighbors* melakukan klasifikasi dengan menggunakan *data training* dan *testing* yang didapatkan pada proses sebelumnya [6]. Langkah-langkah KNN adalah menentukan parameter *k*, menghitung *euclidean distance* tiap objek terhadap *data training*, mengurutkan hasil perhitungan jarak, dan mengumpulkan objek sesuai parameter *k* dan mengelompokkan terhadap yang paling mayoritas. Berikut ini rumus *euclidean distance* :

$$D_{Euclidean}(x,y) = \sqrt{\sum_{k=1}^n (x_k - y_k)^2} \quad (1)$$

Pada persamaan (1), *k* merupakan variabel data, *n* adalah banyaknya parameter, *x* adalah *data testing* sedangkan *y* adalah *data training*.

*Confusion matrix* merupakan salah satu cara untuk mengevaluasi dan meninjau keakuratan serta menentukan *error* pada perhitungan yang telah dilakukan. Penggunaan metode *confusion matrix* digambarkan pada tabel berikut ini :

Tabel 1. *Confusion matrix*

		Classified	
		Positive	Negative
Actual	Positive	True Positive (TP)	False Negative (FN)
	Negative	False Positive (FP)	True Negative (TN)

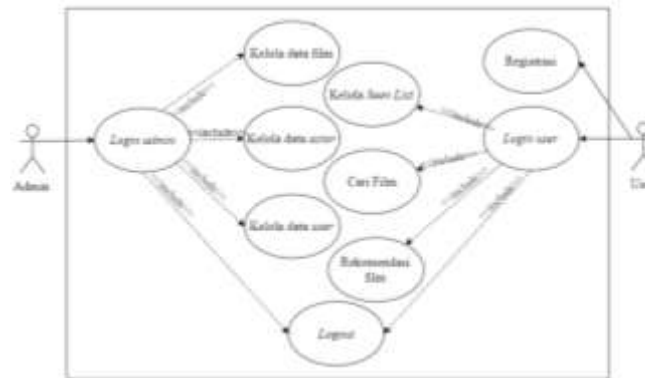
Berdasarkan Tabel 1, *True Positive* (TP) adalah hasil prediksi sistem yang positif serta sesuai dengan target yang positif, *True Negative* (TN) merupakan hasil prediksi sistem yang negatif serta sesuai dengan target yang negatif, sedangkan *False Positive* (FP) adalah hasil prediksi sistem yang positif namun hasil pada target negatif, dan *False Negative* (FN) adalah hasil prediksi sistem yang negatif namun hasil pada target positif. Perhitungan presisi diambil dari  $(TP) / (TP + FP)$ , sedangkan untuk menghitung *recall* menggunakan rumus  $(TP) / (TP + FN)$ .

### 3. HASIL DAN PEMBAHASAN

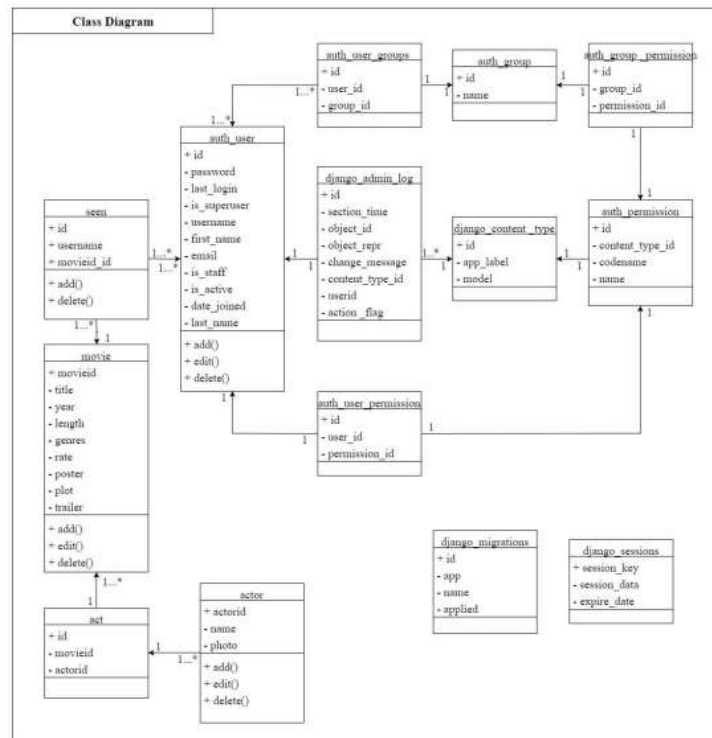
#### 3.1 Communication

Tahap komunikasi melibatkan *developer system* dan pengguna untuk mengidentifikasi kebutuhan yang berupa garis besar kebutuhan dasar dari sistem yang akan dirancang. Pada tahap ini dilakukan pengumpulan data mengenai masalah yang terjadi dengan wawancara, observasi, dan studi literatur, serta identifikasi masalah dan sistem yang diusulkan. Pada dasarnya masyarakat membutuhkan referensi yang bisa menjawab film apa saja yang sesuai dengan kesukaan mereka. Banyaknya informasi yang beredar di internet justru membingungkan mereka untuk memilih. Sistem yang diusulkan untuk dibangun merupakan sistem rekomendasi film yang dapat membantu pengguna untuk mendapatkan saran film sesuai selera dan tipe film yang telah ditonton sebelumnya yang dilihat dari *genre* serta *rating*. Sistem ini dapat digunakan penonton film untuk mendapatkan saran film serta untuk perusahaan sebagai sarana menyalurkan film yang dimiliki dan ditawarkan kepada pengguna sesuai selera pengguna.

#### 3.2 Quick Plan and Modelling Quick Design



Gambar 2. Use Case Diagram Sistem Rekomendasi Film



Gambar 3. Class Diagram Sistem Rekomendasi Film

*Use Case Diagram* menggambarkan apa saja yang terjadi dalam suatu sistem (kegiatan) serta siapa yang melakukan kegiatan tersebut (aktor). Aktor yang terlibat dalam penggunaan sistem rekomendasi film ini adalah admin dan user yang tampak pada gambar 2.

*Class Diagram* merupakan gambaran alur database yang ada dalam suatu program sistem. Diagram yang juga dikenal sebagai kelas diagram ini membantu dalam menggambarkan atau visualisasi struktur sistem yang dilihat dari pendefinisian seluruh kelas yang digunakan untuk membangun sistem. Class diagram untuk pembangunan sistem rekomendasi film digambarkan pada gambar 3.

### 3.3 Construction of Prototype

Pada gambar 4 merupakan proses *data cleaning* yang dimulai dengan menghapus duplikasi data namun tetap mempertahankan data yang pertama muncul. Kemudian mengecek apakah terdapat *missing values* yaitu data *null* atau kosong. Langkah selanjutnya mengecek tipe data apakah telah sesuai dengan yang dibutuhkan atau belum. Perhitungan tidak dapat berjalan apabila terdapat data *string*, maka dilakukan konversi data *string* menjadi *integer*.

```
# Removing duplicates
movies_df.drop_duplicates(keep="first", inplace=True)

movies_df.columns
Index(['movieid', 'title', 'genres', 'imdbid', 'plot', 'popularity', 'poster',
       'length', 'rate'],
      dtype='object')

#Missing Values
#menulit cek data null
movies_df.isnull().sum()

movieid    0
title      0
genres     0
imdbid     0
plot       0
popularity 0
poster     0
length     0
rate       0
dtype: int64

movies_df.shape
(1554, 8)

#From String to Numeric
#blank helynya gausah
numberValue = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,
movies_df['genres'].unique()
movies_df.replace(to_replace=['Action', 'Action|Adventure'],
)

movies_df.head()

   movieid  rate  genres
0         1   7.7     23
1         2   6.9     15
2         3   6.5     38
3         4   6.1     32
4         5   5.7     30
```

Gambar 4. Proses *Data Cleaning*

*Data selection* pada gambar 5 dilakukan dengan mempertahankan data yang dibutuhkan serta membuang atribut yang tidak diperlukan dalam pengolahan data. Data yang diperlukan dalam analisis ini adalah *genres* dan *rate*, maka data lainnya yang perlu dihapus adalah *title*, *imdbid*, *plot*, *popularity*, *poster*, dan *length*.

```
#Dropping Data
movies_df.drop(['title', 'imdbid', 'plot', 'popularity', 'poster', 'length'],
              axis=1, inplace=True)
movies_df.head()

   movieid  genres  rate
0         1  Animation  7.7
1         2  Adventure  6.9
2         3   Comedy  6.5
3         4   Comedy  6.1
4         5   Comedy  5.7
```

Gambar 5. Proses *Data Selection*

Setelah melakukan *data cleaning* dan *selection*, selanjutnya data dibagi menjadi *data training* dan *data testing* untuk dapat digunakan pada proses klasifikasi tampak pada gambar 6.

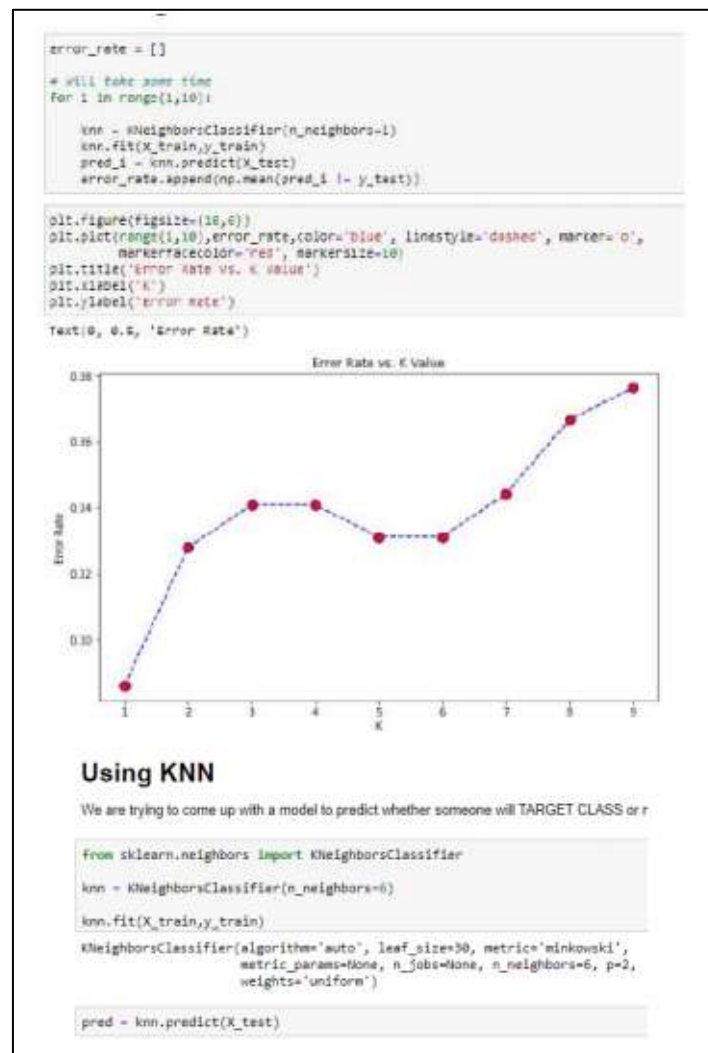
Data awal yang ada adalah sebanyak 1.554 *record*, kemudian akan dibagi menjadi *data training* sebanyak 80% dan *data testing* 20%.

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(scaled_features, movies_df['genres'],
                                                  test_size=0.2, random_state=0)
```

Gambar 6. Proses Split *Data Training* dan *Data Testing*

### 3.4 Klasifikasi Algoritma *K-Nearest Neighbors*

Perhitungan klasifikasi Algoritma *K-Nearest Neighbors* diawali dengan menentukan nilai  $k$ . Nilai  $k$  dapat mempengaruhi tingkat akurasi perhitungan, maka diperlukan perhitungan untuk mencari nilai  $k$  yang optimal. Implementasi *K-Nearest Neighbors* ditunjukkan pada gambar 7 berikut :



Gambar 7. Klasifikasi dengan KNN

Perhitungan hasil kinerja klasifikasi dilakukan dengan menggunakan *confusion matrix* kemudian akan didapatkan tingkat presisi dan *recall* dari perhitungan yang telah dilakukan. Pada gambar 8 berikut ini akan menunjukkan hasil pengukuran *confusion matrix*.



```
# NOW WITH K=5
knn = KNeighborsClassifier(n_neighbors=5)

knn.fit(X_train,y_train)
pred = knn.predict(X_test)

print('WITH K=5')
print('\n')
print(confusion_matrix(y_test,pred))

WITH K=5

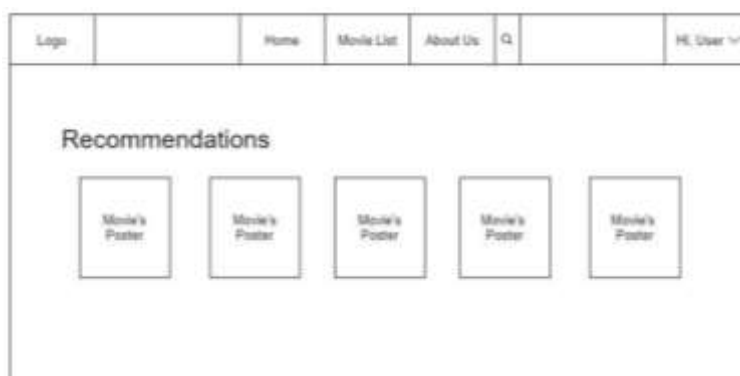
[[ 6  2  0 ...  0  0  0]
 [ 1 10  0 ...  0  0  0]
 [ 0  0  1 ...  0  0  0]
 ...
 [ 0  0  0 ...  6  1  0]
 [ 0  0  0 ...  0  0  0]
 [ 0  0  0 ...  2  0  0]]
```

Gambar 8. Hasil *Confusion Matrix*

Berdasarkan perhitungan yang telah dilakukan dengan *data testing* sebanyak 20% dan *data training* 80% dari 1.554 *record* data, serta *k* sebanyak 5, didapatkan presisi 45% dan *recall* 60%.

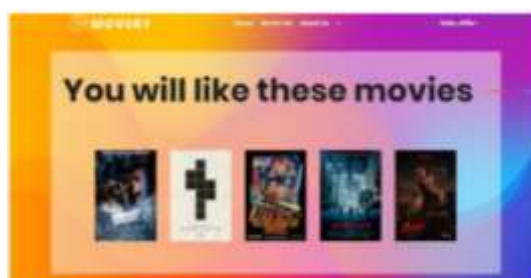
### 3.5 Desain Input dan Output

Desain Input dan Output merupakan tahapan untuk membuat konsep desain sebelum sistem dirancang. Desain tersebut dalam bentuk kerangka *interface* serta menampilkan visualisasi tampilan sistem. Gambar 9 merupakan contoh dari desain output.



Gambar 9. Desain Output Rekomendasi Film

### 3.6 Implementasi *User Interface*

Gambar 10. Implementasi *User Interface* Sistem Rekomendasi Film

Dalam gambar 10 Implementasi *user interface* didasarkan pada desain input/output yang telah dilakukan sebelumnya. Implementasi *user interface* memiliki tujuan untuk mendesain



rancangan antarmuka suatu sistem agar pengguna dapat berinteraksi dengan sistem, sehingga pengguna dapat mengakses informasi yang ada pada sistem.

### 3.7 Deployment Delivery and Feedback

Metode uji coba sistem yang digunakan adalah metode *blackbox testing*. *Blackbox testing* merupakan sebuah metode pengujian perangkat lunak tanpa perlu mengetahui struktur internal kode program namun hanya mengamati hasil eksekusi melalui uji fungsional sistem seperti pada tabel 2 dibawah ini.

Tabel 2. Contoh Pengujian *Blackbox* oleh User

No	User Interface	Input	Output	Hasil Uji
1	Registrasi dengan tidak mengisi <i>username</i> , <i>password</i> dan <i>password confirmation</i> pada masing-masing field.	<i>Username</i> : () <i>Password</i> : () <i>Password</i> <i>Confirmation</i> : ()	Hasil <i>field username</i> , <i>password</i> dan <i>password confirmation</i> : “Please fill out this field”	Komponen berjalan dengan baik
2	Registrasi dengan mengisi <i>username</i> dan <i>password</i> serta mengosongkan <i>password confirmation</i> .	<i>Username</i> : (benar) <i>Password</i> : (benar) <i>Password</i> <i>Confirmation</i> : ()	Hasil <i>field password confirmation</i> : “Please fill out this field”	Komponen berjalan dengan baik
3	Registrasi dengan mengisi <i>username</i> lalu <i>password</i> dan <i>password confirmation</i> yang sama.	<i>Username</i> : (benar) <i>Password</i> : (benar) <i>Password</i> <i>Confirmation</i> : (benar)	Hasil muncul notifikasi “Your account has been successfully registered. Please login”	Komponen berjalan dengan baik

## 4. KESIMPULAN

Berdasarkan pembahasan yang sudah diuraikan, maka dapat ditarik kesimpulan dalam penelitian ini bahwa metode *content based filtering* berhasil diterapkan dalam sistem dimana sistem memberikan output berupa rekomendasi judul film berdasarkan kemiripan genre dan rating. Akan tetapi karena hanya menggunakan 2 variabel saja, maka sistem rekomendasi ini menghasilkan nilai presisi 45% dan nilai recall 60%. Sistem rekomendasi film ini dapat digunakan oleh perusahaan dalam industri film untuk memberikan referensi atau rekomendasi film kepada penonton berdasarkan preferensi film yang telah ditonton sebelumnya, kemudian rekomendasi yang dihasilkan dapat meningkatkan *knowledge* penonton mengenai berbagai film baru dan menonton film tersebut sehingga pendapatan perusahaan film dapat terus meningkat

## 5. SARAN

Untuk penelitian selanjutnya akan lebih baik menggunakan atribut film lainnya seperti pemain, negara asal, maupun tahun rilis untuk meningkatkan keakuratan hasil rekomendasi film. Kemudian bisa dicoba pendekatan yang lain seperti *Collaborative Filtering* atau *Hybrid Filtering* untuk mendapatkan hasil rekomendasi yang lebih baik.

## DAFTAR PUSTAKA

- [1] Pheni Cahya Kartika, *Rasionalisasi Perspektif Film Layar Lebar Beradaptasi Karya Sastra*, Jurnal Pena Indonesia, Volume 2, Nomor 2, Oktober 2016, ISSN: 22477-5150
- [2] Ricci, F., Rokach, L., Shapira, B., 2011, *Recommender Systems Handbook*, Springer.

- 
- [3] Hunt, N., dan Gomez-Urbe, C A., 2015, “*The Netflix Recommender System: Algorithms, Business Value, and Innovation*,” ACM Transactions on Management Information Systems, vol. 6, no. 4.
  - [4] Rachman, Y. F., Saptono, R., dan Winarno, 2018, “*Comparison of C4.5 Algorithm and K-Nearest Neighbors on the Classification of Multiple Intelligence Test Results for Recommended Student Lecture*,” ITSMART: Jurnal Ilmiah Teknologi dan Informasi, vol. 7, no. 2, 108-114.
  - [5] MI Fathurrahman, D Nurjanah, R Rismala, *Sistem Rekomendasi pada Buku dengan Menggunakan Metode Trust-Aware Recommendation* , e-Proceeding of Engineering : Vol.4, No.3 Desember 2017 , Page 4966, ISSN : 2355-9365
  - [6] R. R. Sani, J. Zeniarja, and A. Luthfiarta, “Penerapan Algoritma K-Nearest Neighbor pada Information Retrieval dalam Penentuan Topik Referensi Tugas Akhir,” *J. Appl. Intell. Syst.*, vol. 1, no. 2, pp. 123–133, 2016
-