

PROYEK AKHIR
MATA KULIAH KECERDASAN BUATAN
SEMESTER GANJIL 2023-2024

Sistem Rekomendasi Film Berbasis Konten dengan
Pendekatan *Rule-Based Reasoning* Menggunakan *Cosine Similarity*



Disusun oleh:

Kelompok 7 Kelas E

- | | |
|-------------------------------|-----------------|
| 1. I Wayan Ivan Zenatmaja | 225150207111020 |
| 2. Muhammad Hasan Fadhlillah | 225150207111026 |
| 3. Muhammad Husain Fadhlillah | 225150207111027 |
| 4. Hilda Tri Fatikasari Hilal | 225150207111023 |
| 5. M. Arsyah Zain Yashifa | 225150200111008 |

Dosen Pengampu : Dr. Ir. Drs. Achmad Ridok, M.Kom

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER UNIVERSITAS BRAWIJAYA
MALANG
2023

DAFTAR ISI

DAFTAR GAMBAR.....	3
DAFTAR TABEL.....	4
BAB I PENDAHULUAN.....	5
1.1. Latar Belakang.....	5
1.2. Rumusan Masalah.....	6
1.3. Batasan Masalah.....	6
1.4. Tujuan.....	7
1.5. Manfaat.....	8
1.6. Sistematika Penyusunan Laporan.....	8
BAB II TINJAUAN PUSTAKA.....	9
2.1. <i>Cosine Similarity</i>	9
2.2. <i>Association Rule</i>	11
BAB III METODOLOGI DAN PERANCANGAN.....	14
3.1. Identifikasi Masalah.....	14
3.2. Studi Literatur.....	14
3.3. Proses Pengambilan Data.....	14
3.4. Pengolahan Data dan Analisis Data.....	14
3.5. Rancangan antar muka.....	15
3.6. Algoritma metode yang diusulkan.....	15
3.7. Proses Manualisasi.....	16
3.8. Rancangan program.....	16
3.9. Perancangan Skenario Uji Coba.....	17
BAB IV IMPLEMENTASI DAN PEMBAHASAN.....	18
4.1. Implementasi Tampilan Program.....	18
4.2. Implementasi Metode.....	18
4.3. Proses Pengujian.....	22
4.4. Hasil Pengujian dan Analisis.....	23
BAB V KESIMPULAN DAN SARAN.....	24
5.1. Kesimpulan.....	24
5.2. Saran.....	24
DAFTAR PUSTAKA.....	26

DAFTAR GAMBAR

Gambar 2.1 Konsep Cara Kerja <i>Content-based Filtering</i>	9
Gambar 3.1 <i>Cosine Similarity</i>	16
Gambar 4.1 Implementasi Program.....	18

DAFTAR TABEL

Tabel 3.1 Feature Selection.....	15
Tabel 3.2 Data Cleaning.....	15
Tabel 3.3 TF-IDF.....	15
Tabel 4.1 <i>Source Code</i>	19
Tabel 4.2 Hasil Pengujian.....	24

BAB I

PENDAHULUAN

1.1. Latar Belakang

Society 5.0, era digital yang ditandai dengan perkembangan pesat teknologi informasi, telah membawa perubahan signifikan dalam berbagai aspek kehidupan, termasuk cara netizen memaknai dan mengilhami Sistem Rekomendasi. Sistem ini, meskipun tampak sederhana, telah menjadi bagian penting dari kehidupan sehari-hari, terutama dalam ranah hiburan seperti rekomendasi film.

Alasan mengapa fenomena ini diambil sebagai bahan proyek kami adalah karena pentingnya memahami bagaimana teknologi dan algoritma dapat membantu mempersonalisasi pengalaman pengguna, khususnya dalam konteks rekomendasi film. Dengan memahami bagaimana sistem ini bekerja, kita dapat meningkatkan efisiensi dan efektivitas rekomendasi, sehingga pengguna mendapatkan rekomendasi yang lebih sesuai dengan preferensi mereka.

Sistem rekomendasi memiliki komponen yang di antaranya *Candidate Generations Systems*, metode di mana dihasilkan sebuah subset yang lebih kecil dan direkomendasikan kepada pengguna dari ribuan *item* yang tersedia. Tipe dari *Candidate Generations System* yang digunakan adalah *Content-based filtering System*. Dari kalimatnya, berbasiskan dengan konten, menebak perilaku pengguna berdasarkan fitur *item* yang diharapkan bereaksi secara positif (Tidak memerlukan data pengguna lain untuk rekomendasi satu pengguna). *Filtering System* menggunakan sumber daya *implicit feedback*, pengguna membuka sebuah film yang didasarkan atas sebuah genre yang ada, genre ini nantinya akan menentukan rekomendasi film selanjutnya yang diharapkan sesuai dengan keinginan pengguna.

Dalam penelitian ini, kami mengusulkan konsep baru dalam sistem rekomendasi, yaitu penggunaan metode Cosine Similarity dan Rule-Based Reasoning. Metode Cosine Similarity efektif untuk mengukur kesamaan atribut-atribut dari suatu film seperti genre, aktor, sutradara, dan atribut lainnya berdasarkan vektor representasi dari atribut-atribut tersebut. Dalam konteks perhitungan kesamaan, Cosine Similarity mengukur sudut antara dua vektor. Ketika dua vektor mendekati sudut nol (cosine similarity mendekati 1), itu menunjukkan bahwa kedua vektor tersebut sangat mirip. Sementara itu, Rule-Based Reasoning menerapkan aturan secara langsung pada serangkaian fakta yang didasarkan atas kebiasaan perilaku pengguna. Teknik Rule-Based Reasoning yang digunakan adalah *association rule*.

Oleh karena itu, kami berharap dapat mengembangkan sistem rekomendasi film yang lebih baik dan lebih personal. Sistem ini akan memberikan pengalaman yang lebih memuaskan kepada pengguna dengan menggabungkan analisis konten film, *cosine similarity*, dan *rule-based reasoning*. Dengan demikian, penting untuk terus mengukur kinerja sistem dari perspektif pengguna dan memastikan bahwa sistem memenuhi kebutuhan dan ekspektasi mereka, sehingga dapat memberikan pengalaman yang lebih baik bagi pengguna. Selain itu, penelitian ini juga diharapkan dapat memberikan kontribusi dalam pengembangan teknologi informasi di era Society 5.0.

1.2. Rumusan Masalah

Berikut adalah rumusan masalah yang akan dibahas dalam proyek ini:

1. Bagaimana meningkatkan efisiensi dan efektivitas sistem rekomendasi film, sehingga pengguna mendapatkan rekomendasi yang lebih sesuai dengan preferensi mereka. Hal ini melibatkan pemahaman tentang bagaimana Candidate Generations Systems bekerja, khususnya Content-based filtering System, dan bagaimana sistem ini menggunakan sumber daya implicit feedback untuk menghasilkan rekomendasi yang sesuai dengan keinginan pengguna.
2. Bagaimana pengembangan konsep baru dalam sistem rekomendasi, yaitu penggunaan metode Cosine Similarity dan Rule-Based Reasoning, dapat diintegrasikan dalam sistem rekomendasi untuk menghasilkan rekomendasi yang lebih baik dan lebih personal.
3. Bagaimana cara terbaik untuk mengukur kinerja sistem dari perspektif pengguna dan memastikan bahwa sistem memenuhi kebutuhan dan ekspektasi mereka, serta bagaimana hasilnya dapat digunakan untuk meningkatkan sistem rekomendasi.

1.3. Batasan Masalah

Dalam perancangan sistem rekomendasi film, terdapat beberapa constraint (batasan) yang perlu diperhatikan untuk memastikan kinerja dan efektivitas sistem. Berikut adalah beberapa constraint yang umumnya dihadapi dalam pengembangan sistem rekomendasi film:

1. Keterbatasan Data

- Ketidaklengkapan Data

Jika dataset yang digunakan tidak lengkap, sistem mungkin kesulitan memberikan rekomendasi yang akurat.

- Ketidakkonsistenan Data

Inkonsistensi atau ketidakpastian dalam data bisa mempengaruhi kualitas rekomendasi.

2. Keterbatasan Kategori Film

- Terbatasnya Jenis Film

Sistem rekomendasi mungkin tidak memberikan rekomendasi yang baik jika terdapat keterbatasan dalam jenis film yang ada dalam database.

3. Permasalahan Cold Start

- Cold Start User

Sistem mungkin kesulitan memberikan rekomendasi kepada pengguna baru yang belum memiliki riwayat penilaian atau preferensi film.

4. Keterbatasan Algoritma

- **Ketidakmampuan Menangani Big Data**

Jika sistem tidak dirancang untuk menangani jumlah pengguna atau film yang sangat besar, kinerjanya dapat terpengaruh.

- **Ketidakmampuan Menangani Perubahan Dinamis:**

Sistem harus mampu menanggapi perubahan preferensi pengguna seiring waktu.

5. Keterbatasan Evaluasi

- **Ketidakmampuan Mengukur Kualitas Rekomendasi**

Menilai kualitas rekomendasi dengan tepat adalah tantangan, dan matrik evaluasi yang buruk dapat memberikan umpan balik yang tidak akurat.

6. Masalah Kebijakan dan Etika

- **Privasi Pengguna**

Perlindungan privasi pengguna adalah hal penting, dan sistem perlu memastikan bahwa data pribadi pengguna tidak disalahgunakan.

- **Bias dan Fairness**

Sistem dapat memiliki bias yang tidak disengaja dalam memberikan rekomendasi, dan perlu diupayakan untuk menghindari ketidakadilan atau diskriminasi.

7. Kompleksitas Implementasi

- **Keterbatasan Teknologi**

Keterbatasan dalam teknologi atau sumber daya komputasi dapat membatasi kompleksitas algoritma yang dapat diimplementasikan.

Dalam pengembangan sistem rekomendasi film, penting untuk mempertimbangkan dan mengatasi constraint ini agar sistem dapat memberikan rekomendasi yang relevan, akurat, dan sesuai dengan kebutuhan pengguna.

1.4. Tujuan

Proyek ini bertujuan untuk meningkatkan efisiensi dan akurasi sistem rekomendasi film berbasis konten dengan memanfaatkan metode Cosine Similarity dan Rule-Based Reasoning dalam era Society 5.0.

Dengan menggabungkan analisis konten film, cosine similarity, dan reasoning berbasis aturan, diharapkan proyek ini dapat memberikan solusi yang lebih personal dan memuaskan bagi pengguna.

1.5. Manfaat

Pengerjaan proyek akhir ini akan memberikan sejumlah manfaat yang signifikan. Pertama, pengembangan sistem rekomendasi film berbasis konten dengan memanfaatkan metode Cosine Similarity dan Rule-Based Reasoning diharapkan dapat meningkatkan kualitas rekomendasi film, menciptakan pengalaman menonton yang lebih personal dan memuaskan bagi pengguna. Manfaat ini mencakup peningkatan kepuasan pengguna, potensi peningkatan retensi pengguna, dan mendorong interaksi lebih lanjut dengan platform hiburan digital.

Kedua, proyek ini dapat memberikan kontribusi pada bidang teknologi informasi dengan mengintegrasikan pendekatan baru dalam sistem rekomendasi. Dengan demikian, manfaatnya mencakup kemajuan dalam pemahaman dan penerapan teknologi informasi, serta memberikan wawasan baru terhadap perkembangan sistem rekomendasi di era Society 5.0.

Selain itu, proyek ini dapat menjadi landasan untuk penelitian lanjutan dalam pengembangan sistem rekomendasi yang lebih canggih dan adaptif. Melalui penerapan metode-metode baru, proyek ini memiliki potensi untuk merangsang inovasi di bidang rekomendasi konten digital dan memberikan sumbangan berharga pada perkembangan teknologi informasi secara keseluruhan.

1.6. Sistematika Penyusunan Laporan

Sistematika penyusunan laporan ini secara garis besar meliputi beberapa bab, yaitu sebagai berikut:

BAB I : Pendahuluan

Menguraikan mengenai latar belakang, rumusan masalah, batasan masalah tujuan dan manfaat dari pengerjaan proyek akhir ini.

BAB II : Tinjauan Pustaka

Menguraikan tentang dasar teori dan referensi yang mendasari yang digunakan untuk menyelesaikan permasalahan pada proyek akhir..

BAB III : Metodologi dan Perancangan

Menguraikan tentang metode dan langkah kerja yang dilakukan dalam proses perancangan dan implementasi Metode yang diusulkan penyelesaian proyek akhir. Pada bagian ini juga disertakan contoh kasus sederhana penyelesaian menggunakan metode yang diusulkan.

BAB IV : Implementasi dan Pembahasan

Membahas tentang hasil skenario dan analisis setiap skenario pengujian yang ada.

BAB V : Kesimpulan dan Saran

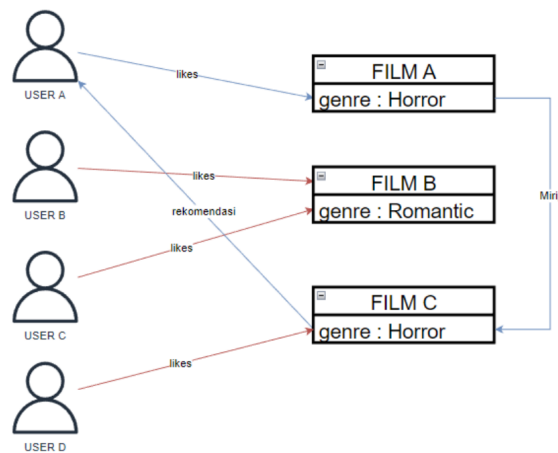
Memuat kesimpulan yang diperoleh dari penelitian implementasi metode yang yang diusulkan untuk menyelesaikan masalah yang diajukan.

BAB II

TINJAUAN PUSTAKA

2.1. Cosine Similarity

1. **“Perancangan Sistem Rekomendasi Film menggunakan metode Content-based Filtering”** karya Evan Salim, Jeanny Pragantha, Manatap Dolok Lauro yang telah diterbitkan oleh Universitas Tarumanegara. Karena berbasis konten, metode yang digunakan ialah, Cosine Similarity dan terselip pula TF-IDF. Mereka saling melengkapi dan tidak bisa dipisahkan. Memberikan penjelasan mengenai metode berbasis konten yang berguna bagi sumber kode sistem rekomendasi secara utuh.



Gambar 2.1 Konsep Cara Kerja *Content-based Filtering*

Rumus perhitungan TF-IDF:

$$W(t, d) = tf(t, d) * idf(t) \quad (1)$$

Keterangan :

d = data ke-d

t = teks ke-t dari kata kunci

W = bobot dokumen ke-d terhadap kata ke-t

tf = term frekuensi

idf= inverse document frequency

Rumus perhitungan *Cosine-similarity*:

$$\text{similarity}(x, y) = \frac{\sum x_i \cdot y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}} \quad (2)$$

Keterangan :

$\sum x_i \cdot y_i$ = vektor (dot) produk dari vektor x dan y

$\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}$ = panjang produk antara vektor x dan y

2. **"Building a movie content based recommender using tf-idf"** karya Alexandre Escolà Nixon yang telah dipublikasikan di platform Medium Towards Data Science Inc. Memberikan penjelasan mengenai metode cosine similarity dan TF-IDF yang berguna bagi sumber kode sistem rekomendasi secara utuh.

$$w_{i,j} = tf_{i,j} \cdot \log \left(\frac{N}{df_i} \right)$$

$tf_{i,j}$ = number of occurrences of i in j

$df_{i,j}$ = number of documents

N = total number of documents

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

$$\mathbf{A} \cdot \mathbf{B} = \|\mathbf{A}\| \|\mathbf{B}\| \cos \theta$$

3. **"Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms"** karya Mohamed Chiny, Marouane Chihab, Omar Bencharef, dan Younes Chihab yang telah dipublikasikan di platform ResearchGate. Memberikan penjelasan mengenai metode cosine similarity dan TF-IDF yang berguna bagi kemaslahatan sumber kode sistem rekomendasi secara utuh.

$$\cos \theta = \frac{A \cdot B}{\|A\| \|B\|}$$

$$tf_i = \frac{n_i}{\sum_k n_k} \quad (1) \quad idf_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|} \quad (2)$$

2.2. Association Rules

1. “Implementasi Algoritma Apriori untuk Analisa Pemilihan Tipe Genre Film Anime (Studi Kasus : myanimelist.net)” artikel jurnal karya Mochammad Abdul Azis, Nur Hadiano, Jaja Miharja, dan Saifulloh Rifai yang telah diterbitkan oleh STMIK Nusa Mandiri Jakarta.

Terinspirasi dari jurnal STMIK Nusa Mandiri Jakarta, terlihat bahwasanya di sini Tipe Genre Film Anime. Sedangkan, untuk sumber kode yang digunakan adalah untuk tipe genre film acak.

Untuk informasi-informasi yang digunakan terkait metode *association rule*,

$$Support(A)$$

$$= \frac{JumlahTransaksiMengandungA}{Total Transaksi} * 100\%$$

$$Confidence(A \rightarrow B)$$

$$= \frac{JumlahTransaksiMengandungA \& B}{Jumlah TransaksiMengandung A} * 100\%$$

Nilai support dari masing masing tipe genre

$$= \frac{\sum Pemilihan Mengandung A}{\sum Pemilihan} * 100\%$$

Kombinasi 2 Item Set

$$= \frac{\sum Pemilihan Mengandung A \cap B}{\sum Pemilihan} * 100\%$$

2. “A way to compare measures in association rule mining” tesis karya Peter Fjällström yang telah diterbitkan oleh Umeå universitet. Memberikan penjelasan secara rinci dan tuntas mengenai *association rule* secara keseluruhan.

$$scount(x) = |\{t | x \subseteq t, t \in T\}|,$$

$$Support, s(a) = \frac{scount(a)}{N}$$

$$Support, s(a \rightarrow b) = \frac{scount(a \cup b)}{N}$$

$$Confidence, c(a \rightarrow b) = \frac{scount(a \cup b)}{scount(a)}$$

$$Lift, l(a \rightarrow b) = \frac{c(a \rightarrow b)}{s(b)}$$

$$\text{Conviction, } conv(a \rightarrow b) = \frac{1 - s(b)}{1 - c(a \rightarrow b)}$$

$$\phi(a \rightarrow b) = \frac{p(a, b) - p(a)p(b)}{\sqrt{p(a)(1 - p(a))p(b)(1 - p(b))}}$$

- when $c(b \rightarrow a) \geq s(a)$ (positive case) is:

$$\text{Certainty factor, } CF(b \rightarrow a) = \frac{c(b \rightarrow a) - s(a)}{1 - s(a)}$$

- when $c(b \rightarrow a) < s(a)$ (negative case), it is:

$$\text{Certainty factor, } CF(b \rightarrow a) = \frac{c(b \rightarrow a) - s(a)}{s(a)}$$

$$\begin{aligned} \chi^2 &= \sum_{i,j} n \frac{(p(ij) - p(i.)p(.j))^2}{p(i.)p(.j)} = n \left\{ \sum_{i,j} \frac{(p(ij))^2}{p(i.)p(.j)} - 1 \right\} \\ &= nE_{AB}\{X\} \end{aligned}$$

$$\phi = \frac{p(11) - p(1.)p(.1)}{\sqrt{p(1.)(1 - p(1.))p(.1)(1 - p(.1))}}$$

$$I_1 = I(p(a|b) \geq p(a))$$

$$I_2 = I(p(a|b) < p(a))$$

$$M_1 = \frac{|p(a|b) - p(a)|}{(1 - p(a))^{I_1}(p(a))^{I_2}}$$

$$M_2 = \frac{|p(a|b) - 0.5|}{0.5}$$

$$M = \sqrt{M_1 M_2}$$

3. **“Confidence in an association rule”** dokumentasi yang telah diterbitkan oleh IBM.

Berisi penjelasan mengenai rumus confidence yang berguna di dalam referensi kode sumber sistem rekomendasi yang telah melibatkan metode Association Rules.

- *Confidence of an association rule* adalah nilai persentase yang menunjukkan betapa seringnya frekuensi *rule head* muncul daripada keseluruhan grup yang mengandung *rule body*. Semakin tinggi nilainya, semakin besar *head items* muncul dalam suatu grup jika diketahui semua *body items* terdapat dalam kelompok tersebut.
- *Confidence* = m/n , yang nilainya adalah sebagai berikut :
- m = The number of groups containing the joined rule head and rule body
- n = The number of groups containing the rule body

4. **“Lift in an association rule”** dokumentasi yang telah diterbitkan oleh IBM.

Berisi penjelasan mengenai rumus Lift yang berguna di dalam referensi kode

sumber sistem

rekomendasi yang telah melibatkan metode Association Rules.

- $Lift = confidence / expected_confidence = confidence / (s(body) * s(head) / s(body)) = confidence / s(head)$
- $s(body) = The\ support\ of\ the\ rule\ body$
- $s(head) = The\ support\ of\ the\ rule\ head$
- Nilai *Lift* yang lebih besar dari 1 menunjukkan bahwa *rule body* dan *rule head* sering muncul bersamaan dari yang diharapkan, hal ini berarti kemunculan *rule body* berpengaruh positif terhadap kemunculan *rule head*.
- Nilai *Lift* yang lebih kecil dari 1 menunjukkan bahwa *rule body* dan *rule head* jarang muncul bersamaan dari yang diharapkan, hal ini berarti kemunculan *rule body* berdampak negatif pada kemunculan *rule head*.
- Nilai *Lift* yang mendekati 1 menunjukkan bahwa *rule body* dan *rule head* muncul bersamaan hampir sesering yang diharapkan, ini berarti kemunculan *rule body* hampir tidak berpengaruh pada kemunculan *rule head*.

BAB III

METODOLOGI DAN PERANCANGAN

3.1. Identifikasi Masalah

Pilihan film yang terlalu banyak terkadang membuat pengguna bingung terkait pemilihan judul film yang bagus, tepat, dan menyenangkan setelah pengguna menonton film sebelumnya. Proyek ini akan fokus pada seorang pengguna bernama "Hasan". Hasan telah menonton film Bacheha-Ye aseman, Ben-Hur, dan Office Space, yang mana beberapa film tersebut merupakan film favoritnya dan ingin mendapatkan rekomendasi film lain yang mirip dengan preferensinya.

Kami akan mengembangkan sistem rekomendasi film berbasis konten yang akan menganalisis fitur-fitur konten film, seperti genre, sutradara, pemain utama, dan elemen-elemen lainnya, menggunakan cosine similarity. Pendekatan "Rule-based Reasoning" akan digunakan untuk mengaplikasikan aturan-aturan yang disusun berdasarkan preferensi Hasan, misalnya, "Jika Hasan menyukai film A dan film A memiliki genre yang sama dengan film B, maka rekomendasikan film B kepada Hasan."

3.2. Studi Literatur

Dibahas lebih lanjut melalui BAB II Tinjauan Pustaka, untuk metode Cosine Similarity mereferensikan terhadap tiga karya ilmiah yang berasal dari Kanada, Indonesia, dan Maroko. Sedangkan, untuk metode Association Rule mereferensikan terhadap empat karya ilmiah yang berasal dari Indonesia, Swedia, dan Amerika Serikat.

3.3. Proses Pengambilan Data

Data yang digunakan adalah data yang berasal dari <https://www.kaggle.com/datasets/harshitshankhdhar/imdb-dataset-of-top-1000-movies-and-tv-shows> berupa tabel "imdb_top_1000" dengan format csv. Data dibaca melalui library Pandas dengan bahasa pemrograman Python.

3.4. Pengolahan Data dan Analisis Data

Data yang digunakan memiliki total 13 features, diantaranya yaitu *Poster_Link*, *Series_Title*, *Released_Year*, *Certificate*, *Runtime*, *Genre*, *IMDB_Rating*, *Overview*, *Meta_score*, *Director*, *Star*, *No_of_votes*, dan *Gross*. Namun features yang digunakan hanya feature *Series_Title* yaitu judul dari film yang ada di dataset dan *Genre* yaitu genre dari film yang ada di dataset. Feature *Genre* digunakan sebagai rujukan untuk membandingkan antara satu film dengan film lainnya sehingga kita dapat mengetahui tingkat kesamaan genre dari kedua film tersebut menggunakan algoritma *cosine similarity*.

Tabel 3.1

```
data = data[['Series_Title', 'Genre']]
```

Langkah berikutnya yang dilakukan setelah menyeleksi *feature* yang akan digunakan adalah memastikan tidak ada data yang kosong dan juga tidak ada yang duplikat. Hal ini dilakukan karena data yang tidak bersih dapat memberikan hasil analisis yang bias atau tidak akurat sehingga dapat mempengaruhi perhitungan statistik. Membersihkan data dapat meningkatkan performa model rekomendasi. Data yang bersih dan bebas dari gangguan dapat membantu model mengidentifikasi pola yang lebih baik dan memberikan rekomendasi yang lebih akurat.

Tabel 3.2

```
data.isnull().sum()
data.duplicated().sum()
```

Selanjutnya dilakukan analisis data genre menggunakan TF-IDF untuk mengonversi daftar genre menjadi matriks TF-IDF serta menghapus kata-kata umum dalam bahasa Inggris yang umumnya tidak memberikan nilai informasi yang tinggi. Matriks TF-IDF memberikan representasi numerik dari setiap film berdasarkan kemunculan kata-kata (genre) dalam dataset. Nilai-nilai dalam matriks tersebut memberikan bobot untuk setiap genre dalam setiap film, dengan memperhitungkan seberapa sering genre tersebut muncul dalam film tertentu (TF) dan seberapa umum genre tersebut secara keseluruhan dalam dataset (IDF) sehingga kita dapat mengidentifikasi seberapa unik suatu genre dalam suatu film dan memahami kemiripan atau perbedaan antara film berdasarkan genre mereka.

Tabel 3.3

```
genre_list = data["Genre"].tolist()
title = data["Series_Title"].tolist()
tfidf = text.TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(genre_list)
```

3.5. Rancangan antar muka

Antar muka yang digunakan berfokus pada CLI (Command Line Interface). Pengguna dapat melakukan *hard-coded* agar output yang diterima sesuai dengan permintaan pengguna (Tidak bisa menerima input secara langsung).

3.6. Algoritma metode yang diusulkan

Dalam pengembangan sistem rekomendasi film berbasis konten dengan pendekatan rule-based reasoning, digunakan dua algoritma utama: *cosine similarity*

untuk pengukuran kesamaan genre antar film dan *rule-based* berupa *association rule* menggunakan algoritma Apriori untuk menentukan aturan asosiasi atau menemukan hubungan tiap-tiap genre.

Cosine similarity digunakan untuk mengukur sejauh mana dua vektor (representasi film) memiliki arah yang sama. Formula cosine similarity antara dua vektor ditunjukkan pada gambar di bawah.

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Gambar 3.1 *Cosine Similarity*

dengan $\mathbf{A} \cdot \mathbf{B}$ adalah hasil perkalian dot antara vektor A dan B, dan $\|\mathbf{A}\|$ serta $\|\mathbf{B}\|$ masing-masing adalah norma Euclidean dari A dan B.

Association rule dengan algoritma Apriori adalah metode analisis data yang mencari hubungan asosiasi antara variabel dalam dataset. Prosesnya melibatkan identifikasi itemset, perhitungan dukungan (support) untuk setiap itemset, pembuatan aturan asosiasi berdasarkan tingkat kepercayaan tertentu, dan evaluasi kinerja aturan. Contoh aturan asosiasi dalam konteks rekomendasi film dapat mencakup pola seperti "Jika pengguna menyukai film A, maka mereka kemungkinan besar juga akan menyukai film B." Apriori memungkinkan identifikasi pola-pola ini dengan menggabungkan itemset yang memiliki dukungan atau support di atas ambang batas tertentu.

3.7. Proses Manualisasi

Proses yang melibatkan manusia di dalamnya. Hal ini berafiliasi dengan aktivitas pengguna terkait input pengguna, feedback pengguna, evaluasi, penyesuaian rule, uji coba, data film yang rinci, dan monitoring secara rutin.

3.8. Rancangan program

Rancangan program untuk sistem rekomendasi film berbasis konten dengan pendekatan rule-based reasoning menggunakan cosine similarity dapat dibagi menjadi beberapa tahap

1. Pengumpulan dan Pemrosesan Data:

Mengumpulkan data film dan melakukan pemrosesan data untuk mengonversi informasi film berupa genre ke dalam bentuk vektor atau matriks yang dapat digunakan untuk perhitungan cosine similarity.

2. Cosine Similarity:
Menggunakan TfidfVectorizer untuk mengukur cosine similarity antara vektor representasi film. Penerapan cosine similarity untuk menghasilkan matriks kesamaan antar film.
3. Rule-Based Reasoning:
Menerapkan aturan-aturan berbasis logika atau *association rule* untuk memperbaiki atau memodifikasi rekomendasi berdasarkan perilaku pengguna. Menyesuaikan bobot atau skor berdasarkan preferensi dan aturan yang diterapkan.
4. Implementasi Antarmuka Pengguna:
Membuat antarmuka pengguna yang memungkinkan pengguna memberikan masukan
5. Evaluasi Kinerja:
Melakukan uji coba untuk mengevaluasi sejauh mana sistem memberikan rekomendasi yang sesuai dengan preferensi pengguna.
6. Pemeliharaan Sistem:
Menyusun prosedur pemeliharaan yang memastikan konsistensi dan peningkatan sistem seiring waktu.

3.9. Perancangan Skenario Uji Coba

Skenario uji coba dirancang untuk menguji fungsionalitas dan kinerja sistem rekomendasi. Beberapa skenario yang mungkin dilibatkan di antara lain adalah

1. Uji Coba Kesamaan:
Memeriksa sejauh mana sistem dapat menghasilkan rekomendasi film yang mirip dengan film yang sudah ditonton oleh pengguna.
2. Uji Coba Antarmuka Pengguna:
Mengevaluasi keterlibatan pengguna dengan antarmuka termasuk kemudahan penggunaan.
3. Uji Coba Umpan Balik Pengguna:
Mengumpulkan umpan balik pengguna terhadap rekomendasi yang diberikan dan melihat sejauh mana sistem dapat memperbaiki rekomendasi berdasarkan umpan balik tersebut.
4. Uji Coba Pemeliharaan:
Melakukan uji coba untuk memastikan bahwa sistem tetap efektif dan konsisten setelah pemeliharaan rutin, seperti pembaruan data atau penyesuaian aturan.

BAB IV

IMPLEMENTASI DAN PEMBAHASAN

4.1. Implementasi Tampilan Program

Tampilan ini masih berupa *CLI (Command Line Interface)* yang masih belum dapat berinteraksi dengan pengguna secara sempurna.

```
[16] Mengisi inputan user_title sebagai film-film yang telah ditonton oleh pengguna agar bisa merekomendasikan film yang sesuai ataupun mendekati keinginan pengguna

[ ] user_title = [title[101], title[300], title[800]] # Harus berupa array
print_recommendations(user_title, title, similarity, indices, data, association_rules_data)

Nilai cosine similarity pada indeks film ke- : [(179, 1.1827760858526037), (281, 1.1827760858526037), (966, 1.1827760858526037), (137, 1.1594301760722343), (193, 1.1594301760722343)]
Recommendations for 'Toy Story', 'Ben-Hur', 'Office Space':
1. The Great Escape
2. Barry Lyndon
3. Apollo 13
4. Queen
5. The Gold Rush
6. Hunt for the Wilderpeople
7. Almost Famous
8. Nebraska
9. Interstate 60: Episodes of the Road
10. The Peanut Butter Falcon
```

Gambar 4.1 Implementasi Program

4.2. Implementasi Metode

Berikut ini adalah penjelasan kode program yang telah kami buat. Program dapat diakses secara *online* melalui tautan di bawah ini.

Link program: [AI2023_E7_Sistem Rekomendasi Film.ipynb](#)

Tabel 4.1 *Source Code*

```
import pandas as pd
import numpy as np
from sklearn.metrics.pairwise import cosine_similarity
from sklearn.feature_extraction.text import TfidfVectorizer
from mlxtend.frequent_patterns import apriori, association_rules
from mlxtend.preprocessing import TransactionEncoder
from google.colab import files
import warnings
warnings.filterwarnings('ignore')
```

Kode di atas digunakan untuk memanggil library yang tersedia di python yang nantinya akan digunakan untuk melakukan pemrosesan dan pembuatan sistem rekomendasi film.

```
uploaded = files.upload()
data = pd.read_csv('imdb_top_1000.csv')
data = data.iloc[:1000]
```

Kode di atas digunakan untuk membaca atau import file yang akan menjadi dataset di

project ini

```
data = data[['Series_Title', 'Genre']]
print(data)
```

Kode di atas akan memperbarui dataset 'data' sehingga hanya memiliki kolom 'Series_Title' dan 'Genre'

```
data.isnull().sum()
data.duplicated().sum()
```

Kode di atas digunakan untuk mengecek apakah di dalam dataset tersebut ada data yang kosong atau duplikat sehingga apabila memang ada, diperlukan penanganan lebih untuk memperbaiki kualitas data baik seperti menghapus data duplikat, menghapus data kosong, imputasi, dan lain-lain.

```
# Menggunakan TF-IDF untuk representasi teks genre
genre_list = data["Genre"].tolist()
title = data["Series_Title"].tolist()
tfidf = text.TfidfVectorizer(stop_words='english')
tfidf_matrix = tfidf.fit_transform(genre_list)
print("Perhitungan TF-IDF")
print(tfidf_matrix)
```

Kode di atas digunakan untuk merepresentasikan teks genre dari suatu data menggunakan TF-IDF. Dengan ini, kita dapat melihat representasi TF-IDF dari teks genre dalam bentuk matriks. Setiap baris matriks mewakili satu elemen dari genre_list, dan setiap kolom mewakili kata atau frasa tertentu yang muncul dalam seluruh teks genre. Nilai dalam matriks mencerminkan bobot kata atau frasa tersebut dalam setiap elemen genre.

```
# Menghitung Cosine Similarity
similarity = cosine_similarity(tfidf_matrix)
print("Perhitungan Cosine Similarity")
print(similarity)
```

Kode di atas menggunakan cosine similarity untuk mengukur sejauh mana setiap pasang elemen dalam genre_list (menggunakan representasi TF-IDF) mirip satu sama lain.

```
indices = pd.Series(data.index,
index=data["Series_Title"]).drop_duplicates()
# Association Rules
def get_association_rules(data, min_support=0.1,
min_confidence=0.5):
# Menggunakan TransactionEncoder untuk mengkonversi data menjadi
```

```

format biner
te = TransactionEncoder()
te_ary = te.fit(data).transform(data)
df = pd.DataFrame(te_ary, columns=te.columns_)

# Menggunakan Apriori untuk menemukan itemset yang sering muncul
frequent_itemsets = apriori(df, min_support=min_support,
use_colnames=True)

# Menggunakan Association Rules untuk mendapatkan aturan asosiasi
rules = association_rules(frequent_itemsets, metric="confidence",
min_threshold=min_confidence)

return rules
# Menjalankan Association Rules pada data genre
association_rules_data =
get_association_rules(data["Genre"].str.split(',').apply(lambda x:
[i.strip() for i in x]), min_support=0.05, min_confidence=0.2)
print(association_rules_data)

```

Kode di atas menggunakan algoritma asosiasi Apriori untuk menemukan aturan asosiasi dalam dataset genre film. Aturan tersebut mencakup informasi tentang sejauh mana genre-genre tertentu sering muncul bersama dan sejauh mana kepercayaan antara genre-genre tersebut.

```

# Method recommendation
def recommendation(user_title, title, similarity, indices, data,
association_rules_data):
n = len(user_title) # Jumlah film yang sudah ditonton oleh pengguna
user_index = [] # Index film yang sudah ditonton
for i in range(n):
user_index.append(indices[user_title[i]])
index = indices[title] # Index film yang akan direkomendasikan
score = list(enumerate(similarity[index]))

# Menghitung user score
user_score = []
for i in range(n):
user_score.append(score[user_index[i]])
# Menghindari merekomendasikan film yang sudah ditonton
user_score[i][1][user_index[:]] = 0.0

```

```

# Menggunakan aturan asosiasi untuk memodifikasi skor
for i in range(n):
    user_score[i][1][user_index[:]] = user_score[i][1][user_index[:]] *
    get_association_score(user_title[i], association_rules_data)

# Dilakukan enumerasi
us_list = list(enumerate(user_score))

# Menghitung total kesamaan kosinus untuk setiap film
us_sum = [0 for _ in range(len(title))]
for i in range(n):
    us_sum += us_list[i][1][1]
us_sum_list = list(enumerate(us_sum))

# Mengurutkan daftar film yang direkomendasikan
us_sorted = sorted(us_sum_list, key=lambda x: x[1], reverse=True)
print("Nilai cosine similarity pada indeks film ke- :", us_sorted)
us_sorted = us_sorted[:10] # Mengambil 10 film teratas
mov_rec = [i[0] for i in us_sorted]

return [data['Series_Title'][i] for i in mov_rec]

```

Fungsi 'recommendation' di atas memadukan informasi dari cosine similarity dan association rule untuk memberikan rekomendasi film kepada pengguna.

```

# Fungsi untuk mendapatkan skor asosiasi berdasarkan aturan asosiasi
def get_association_score(genre, association_rules_data):
    associated_genres =
    association_rules_data[association_rules_data['antecedents'].apply(
        lambda x: genre in x)]

    if not associated_genres.empty:
        return associated_genres['confidence'].mean()
    else:
        return 1.0 # Jika tidak ada aturan asosiasi, kembalikan skor 1.0

```

Fungsi `get_association_score` digunakan untuk mendapatkan skor asosiasi berdasarkan aturan asosiasi yang telah ditemukan. Fungsi ini membantu mengintegrasikan informasi dari aturan asosiasi ke dalam skor yang dihitung pada fungsi `recommendation`. Skor ini kemudian digunakan untuk memodifikasi skor kemiripan kosinus untuk setiap film yang

telah ditonton oleh pengguna.

```
def print_recommendations(user_title, title, similarity, indices,
data, association_rules_data):
    recommendations = recommendation(user_title, title, similarity,
indices, data, association_rules_data)

    print("Recommendations for '{}':".format(user_title))

    for idx, movie_title in enumerate(recommendations, start=1):
        print("{} . {}".format(idx, movie_title))
```

Fungsi `print_recommendations` ini digunakan untuk mencetak rekomendasi film ke layar berdasarkan film-film yang telah ditonton oleh pengguna, cosine similarity antar film, dan association rules yang telah ditemukan.

```
user_title = [title[101], title[300], title[800]] # Harus berupa
array
print_recommendations(user_title, title, similarity, indices, data,
association_rules_data)
```

Cuplikan kode di atas membuat `user_title` yaitu sebuah daftar judul film yang diasumsikan sebagai film-film yang sudah ditonton oleh pengguna. Kemudian memanggil fungsi `print_recommendations` untuk mendapatkan dan mencetak rekomendasi film.

4.3. Proses Pengujian

Pengujian sistem dilakukan dengan 2 (dua) cara, yaitu dengan alpha testing dan beta testing. Uji alpha testing dilakukan dengan metode black box testing, yaitu dengan cara menguji fungsionalitas aplikasi apakah sudah berjalan sesuai yang dirancang atau belum. Beberapa hasil pengujian dengan black box testing dapat dilihat pada Tabel 4.2 yang berada di 4.4 Hasil Pengujian dan Analisis.

4.4. Hasil Pengujian dan Analisis

Poin Pengujian	Hasil yang Diharapkan	Hasil	Status
Rekomendasi Film	Menampilkan film-film hasil rekomendasi	Berhasil menampilkan rekomendasi film	Valid

Tabel 4.2 Hasil Pengujian

- Poin Pengujian: “Rekomendasi Film”
Hal ini menunjukkan tujuan dari pengujian, yaitu untuk memastikan bahwa sistem atau aplikasi dapat memberikan rekomendasi film kepada pengguna. Sistem ini menggunakan algoritma *cosine similarity* dan *association rule (rule-based reasoning)* untuk menghasilkan rekomendasi yang paling relevan dengan preferensi pengguna.
- Hasil yang Diharapkan: “Menampilkan film-film hasil rekomendasi”
Hal ini berarti bahwa setelah pengguna meminta rekomendasi, sistem diharapkan dapat menampilkan daftar film yang direkomendasikan.
- Hasil: “Berhasil menampilkan rekomendasi film”
Hal ini menunjukkan hasil aktual dari pengujian, yaitu sistem berhasil melakukan apa yang diharapkan. Dalam hal ini, hasilnya positif, yang berarti bahwa sistem berhasil menampilkan daftar film yang direkomendasikan.
- Status: “Valid”
Status ini menunjukkan bahwa sistem rekomendasi film telah berhasil dan hasilnya dapat diandalkan. Hal ini berarti bahwa fitur ini berfungsi dengan baik dan sesuai dengan yang diharapkan serta menegaskan bahwa sistem atau aplikasi telah lulus pengujian untuk poin yang diuji, yaitu kemampuannya dalam menampilkan rekomendasi film.

Secara keseluruhan, hasil pengujian dan analisis ini menunjukkan bahwa sistem rekomendasi film ini telah dirancang dan diimplementasikan dengan baik. Hal ini menunjukkan bahwa algoritma *cosine similarity* dengan pendekatan *rule-based reasoning* dapat memberikan rekomendasi film berbasis konten yang relevan dan memenuhi harapan yang diinginkan.

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil rekomendasi yang dijalankan, dapat diambil kesimpulan sebagai berikut:

1. Penggunaan metode *cosine similarity* dan *association rule* mampu menghasilkan sistem rekomendasi film yang tepat guna setelah diuji oleh metode black box testing.

Metode *cosine similarity* dan *association rule* telah terbukti efektif dalam menghasilkan sistem rekomendasi film yang tepat guna. Metode tersebut mampu menghasilkan rekomendasi film yang sesuai, seperti yang ditunjukkan oleh hasil pengujian menggunakan metode black box testing. Keberhasilan ini mengindikasikan kemampuan sistem untuk secara efektif memanfaatkan teknik-teknik tersebut dalam merancang rekomendasi film yang relevan.

2. Berbasis konten menggunakan data film secara keseluruhan yang dapat terbaca secara maksimal.

Keunggulan sistem berbasis konten terlihat dalam penggunaan data film secara keseluruhan. Hal ini memastikan bahwa sistem dapat memahami dan menganalisis karakteristik film secara menyeluruh sehingga dapat meningkatkan kemampuan sistem untuk memberikan rekomendasi yang lebih akurat. Keberhasilan ini menegaskan pentingnya pilihan data yang menyeluruh dalam meningkatkan kualitas rekomendasi.

3. Sistem rekomendasi akan menghasilkan 10 film pilihan teratas yang disarankan bagi pengguna.

Sistem ini dirancang untuk menghasilkan 10 film pilihan teratas sebagai rekomendasi bagi pengguna. Hal ini memungkinkan pengguna memiliki berbagai pilihan dan memungkinkan mereka untuk memilih film yang paling sesuai dengan preferensi mereka.

5.2. Saran

Saran dalam perancangan sistem rekomendasi ini berdasarkan hasil pengujian secara manual adalah sebagai berikut:

1. Realisasi Menjadi Sistem Aplikasi

Sistem rekomendasi ini memiliki potensi untuk direalisasikan menjadi sistem aplikasi. Ini berarti bahwa sistem ini dapat diintegrasikan ke dalam aplikasi yang ada atau menjadi dasar untuk aplikasi baru. Dengan demikian, pengguna dapat dengan mudah mengakses rekomendasi film kapan saja dan di mana saja, yang akan meningkatkan pengalaman menonton film mereka.

2. Pengujian dan Validasi Lebih Lanjut

Melakukan pengujian dan validasi lebih lanjut terhadap sistem, termasuk uji coba dengan dataset yang lebih besar dan variasi untuk memastikan keandalan dan keberlanjutan kinerja sistem.

3. Optimalkan Algoritma Rekomendasi

Terus mengembangkan algoritma rekomendasi untuk meningkatkan ketepatan dan keberagaman rekomendasi. Pembaruan ini harus disertai dengan evaluasi terhadap dampaknya pada kualitas rekomendasi.

4. Perbarui Data Film Secara Berkala

Menjaga keakuratan rekomendasi dengan memperbarui secara berkala data film yang digunakan oleh sistem.

Dengan mengimplementasikan saran-saran ini, diharapkan sistem rekomendasi film berbasis konten ini dapat terus berkembang dan memberikan pengalaman yang lebih baik bagi pengguna secara keseluruhan.

DAFTAR PUSTAKA

- Fjällström, P. (2016). A way to compare measures in association rule mining. Retrieved from <https://www.diva-portal.org/smash/get/diva2:956424/FULLTEXT01.pdf>
- IBM. (2021a). Confidence in an association rule. Retrieved from https://www.ibm.com/support/knowledgecenter/en/SSEPGG_9.7.0/com.ibm.im.model.doc/c_confidence_in_an_association_rule.html
- IBM. (2021b). Lift in an association rule. Retrieved from https://www.ibm.com/support/knowledgecenter/en/SSEPGG_10.1.0/com.ibm.im.model.doc/c_lift_in_an_association_rule.html
- Chiny, M., Chihab, M., Bencharef, O., & Chihab, Y. 2022. "Netflix Recommendation System based on TF-IDF and Cosine Similarity Algorithms." INTERNATIONAL CONFERENCE ON BIG DATA, MODELLING AND MACHINE LEARNING (BML'21). Retrieved from https://www.researchgate.net/publication/360856267_Netflix_Recommendation_System_based_on_TF-IDF_and_Cosine_Similarity_Algorithms. DOI: 10.5220/0010727500003101.
- Azis, M. A., Hadiano, N., Miharja, J., & Rifai, S. 2018. Implementasi Algoritma Apriori untuk Analisa Pemilihan Tipe Genre Film Anime (Studi Kasus: myanimelist.net). Jurnal PILAR Nusa Mandiri, Volume 15 , Hal 209 - 216. STMIK Nusa Mandiri Jakarta. Retrieved from <https://ejournal.nusamandiri.ac.id/index.php/pilar/article/download/41/36/>.
- Alexandre Escolà Nixon. Building a movie content based recommender using tf-idf. 2020. Medium. Towards Data Science Inc. Retrieved from <https://towardsdatascience.com/content-based-recommender-systems-28a1dbd858f5>.
- Salim, E., Pragantha, J., & Lauro, M.D. Perancangan Sistem Rekomendasi Film menggunakan metode Content-based Filtering. Universitas Tarumanegara. Retrieved from https://linter.untar.ac.id/repository/penelitian/buktipenelitian_10390001_7A281222103549.pdf.