

# Penerapan Metode *Content-Based Filtering* dalam Sistem Rekomendasi *Video Game*

Dewa Ayu Putri Diah Pramesti<sup>a1</sup> dan I Wayan Santiyasa<sup>a2</sup>

<sup>a</sup>Program Studi Informatika, Universitas Udayana  
Jimbaran, Badung, Bali, Indonesia

<sup>1</sup>dwayuputri35@gmail.com

<sup>2</sup>santiyasa@unud.ac.id (Corresponding author)

## Abstract

*Technological developments make people very fond of playing video games, especially teenagers. Due to the increasing number of fans, video games are being produced more and more. This makes the number of video games available very large. The diversity of video games makes users feel confused about which video game they should play next, especially if some users are new to video games. Therefore, a recommender system was built to recommend video games to users by using genre as a benchmark for providing recommendations. The system will provide recommendations according to the preferences desired by the user. The method used in this research is the content-based filtering method because it can produce a fairly high precision value. In this study, the results were obtained in the form of an average precision of 87.75%.*

**Keywords:** *recommender system, content-based filtering, video game*

## 1. Introduction

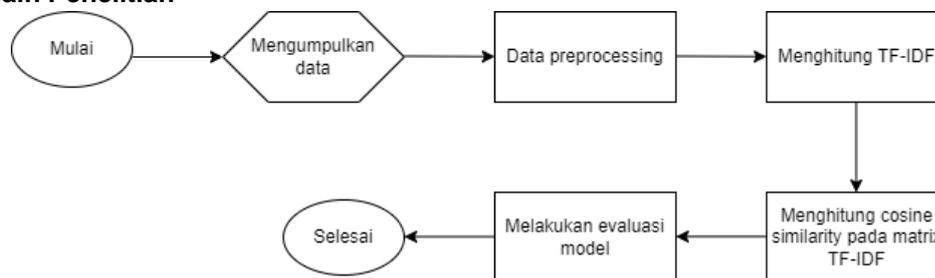
Saat ini, teknologi telah berkembang begitu pesat karena memberikan pengaruh yang sangat signifikan bagi keberlangsungan hidup masyarakat. Pengaruh tersebut salah satunya terjadi pada media massa. Berkembangnya teknologi membuat media massa juga ikut berkembang dari zaman ke zaman. Perkembangan tersebut salah satunya yaitu media massa kini menggunakan *audio visual* sebagai cara menyajikan pesan. Terdapat beberapa jenis *audio visual*, yaitu televisi, film, dan *video game* [1]. *Video game* merupakan salah satu jenis *audio visual* yang kini banyak digemari oleh masyarakat, khususnya oleh para remaja karena keseruannya dapat menghilangkan rasa bosan. Selain itu, *video game* juga menyediakan visualisasi yang menarik sehingga dapat membuat minat para pemainnya semakin meningkat. Oleh karena itu, jumlah produksi dari *video game* kian meningkat sehingga mengakibatkan jumlah *video game* yang beredar menjadi sangat banyak. Hal ini akan membuat masyarakat merasa kebingungan untuk memilih *video game* yang akan mereka mainkan selanjutnya, terlebih jika terdapat beberapa orang yang baru mengenal *video game*. Ada pun beberapa atribut yang dapat menjadi pertimbangan masyarakat dalam memilih *video game*, salah satunya adalah genre. Masyarakat akan memilih *video game* berdasarkan genre yang sesuai dengan preferensi mereka sehingga genre dapat dijadikan landasan untuk memberikan rekomendasi *video game* kepada masyarakat.

Sistem rekomendasi merupakan sebuah sistem yang dapat memudahkan para pengguna untuk mendapatkan rekomendasi satu atau lebih *item* berdasarkan data yang tersedia sehingga mereka dapat membuat keputusan sesuai dengan preferensi mereka [2]. Sistem rekomendasi *video game* dapat membantu para pengguna untuk memilih *video game* berdasarkan dengan preferensi genre mereka di antara banyaknya jumlah *video game* yang beredar. Sistem rekomendasi ini akan menggunakan genre sebagai tolok ukur sistem untuk memberikan rekomendasi kepada para pengguna, sehingga metode yang digunakan untuk memecahkan masalah yang telah dijabarkan sebelumnya adalah metode *content-based filtering*.

Dalam penelitian yang sebelumnya dilakukan oleh Zayyad pada tahun 2021, dibuat sistem rekomendasi buku menggunakan metode *content-based filtering*. Hasil yang didapatkan adalah presisi sebesar 85% menggunakan 20 sampel buku [3]. Hal tersebut membuktikan bahwa metode *content-based filtering* merupakan metode yang baik untuk melakukan rekomendasi.

## 2. Research Methods

### 2.1. Desain Penelitian



Gambar 1. Desain Penelitian

### 2.2. Pengumpulan Data

Penelitian ini menggunakan data sekunder, yaitu berupa data *video games* yang diambil dari situs kaggle.com sebanyak 16.026 records. Data ini memiliki tiga fitur, yaitu id dari *video game* dengan tipe data numerik, nama dari *video game* dengan tipe data *string*, dan genre dari *video game* dengan tipe data *string*.

### 2.3. Tahapan Prapemrosesan Data

Data *video games* yang telah dikumpulkan sebelumnya, akan melalui tahapan prapemrosesan data terlebih dahulu agar data tersebut menjadi lebih terstruktur sebelum digunakan dalam penelitian ini. Tahapan tersebut terdiri dari menghilangkan nilai *null* (baris yang tidak memiliki nilai atau kosong) dan menghapus nilai duplikat yang terdapat dalam fitur nama agar setiap *video games* memiliki nama yang unik.

### 2.4. TF-IDF

TF-IDF atau *Term Frequency-Inverse Document Frequency* merupakan sebuah algoritma yang biasanya digunakan untuk memberikan pembobotan pada setiap kata yang berada dalam satu atau beberapa dokumen [4]. Dalam metode *content-based filtering*, TF-IDF banyak digunakan untuk membentuk profil dari sebuah *item*. TF (*Term Frequency*) digunakan untuk menghitung berapa kali sebuah kata muncul dalam sebuah dokumen. Apabila jumlah kemunculan kata tersebut tinggi, dapat ditarik kesimpulan bahwa kata tersebut penting. TF dapat dihitung menggunakan persamaan (1).

$$tf_{i,j} = f_{i,j} \quad (1)$$

Keterangan:

$tf_{i,j}$  = *term frequency* dari kata  $i$  yang terdapat dalam dokumen  $j$

$f_{i,j}$  = jumlah kemunculan kata  $i$  dalam dokumen  $j$

Namun, jumlah kemunculan kata tersebut belum tentu menandakan bahwa kata yang berada pada sebuah dokumen penting atau tidak. Oleh karena itu, terdapat IDF (*Inverse Document Frequency*) yang dapat menghitung keseluruhan jumlah kemunculan kata dari seluruh dokumen yang ada. IDF dapat dihitung menggunakan persamaan (2).

$$idf_i = \log \left( \frac{N}{df_i} \right) \quad (2)$$

Keterangan:

$idf_i$  = inverse document frequency

$df_i$  = jumlah dokumen yang tersedia dengan kata  $i$  di dalamnya

$N$  = jumlah keseluruhan dokumen pada sistem

## 2.5. Content-Based Filtering

*Content-based filtering* adalah salah satu metode dari sistem rekomendasi yang didasarkan pada atribut, karakteristik, atau fitur yang dimiliki oleh sebuah *item* [5]. Atribut-atribut yang pernah mengalami interaksi dengan user akan dijadikan dasar untuk mencari item yang memiliki atribut yang serupa sehingga sistem dapat memberikan rekomendasi kepada pengguna. Hal tersebut dilakukan dengan melakukan penghitungan *similarity* atribut yang terdapat dalam *item*. Fungsi *similarity* yang paling banyak digunakan adalah *adjusted cosine similarity*, *cosine similarity*, dan *pearson coefficient*. Pada sistem rekomendasi *video game*, akan digunakan fungsi *cosine similarity* yang dapat mengukur kemiripan antara *video game* yang satu dengan *video game* yang lainnya. *Cosine similarity* akan menganggap *video game* sebagai vektor. *Cosine similarity* dapat dihitung dengan persamaan (3).

$$Sim(q, d_j) = \frac{q \times d_j}{|q| \times |d_j|} = \frac{\sum_{i=1}^n W_{i,q} \times W_{i,j}}{\sqrt{\sum_{i=1}^t (W_{i,q})^2} \sqrt{\sum_{i=1}^t (W_{i,j})^2}} \quad (3)$$

Keterangan:

$Sim(q, d_j)$  = *similarity* yang dimiliki oleh *query* dan dokumen

$|q|$  = jarak dari sebuah *query*

$|d_j|$  = jarak dari sebuah dokumen

$W_{i,j}$  = bobot yang dimiliki dokumen ke- $i$

$W_{i,q}$  = bobot yang dimiliki *query* dokumen ke- $i$

## 3. Result and Discussion

Pada tahapan menghitung TF-IDF, menghitung *cosine similarity*, dan melakukan evaluasi model, masing-masing didapatkan hasil yang akan dijabarkan sebagai berikut.

### 3.1. Menghitung TF-IDF

Hal pertama yang dilakukan dalam tahapan menghitung TF-IDF adalah menghapus regular expression yang terdapat dalam data. Regular expression yang dihapus adalah simbol -, \_, dan spasi. Selanjutnya, dilakukan perhitungan idf pada data genre. Akan didapatkan 19 genre berbeda yang telah dihilangkan regular expressionnya. Setelah itu, akan diambil 10 sampel *video game* untuk memeriksa bobot terhadap genre dari *video game* tersebut dan ditransformasikan ke bentuk matriks. Hasil dari penghitungan TF-IDF dapat dilihat pada gambar 2.

	card	platformer	casual	sports	massivelymultiplayer	arcade	action	boardgames	racing	adventure	educational	strategy	rpg	simulation	shooter	family	fighting	indie	puzzle
Motor Money Crash!	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	1.0	0.0	0.0	0.000000	0.0
Decker's Dive	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0
The Chickenator (itch)	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0
Mint (itch)	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0
Tyrant - A Twist of Fate	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0
Life of a Mobster	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.489852	0.0	0.0	0.597792	0.0	0.0	0.0	0.0	0.634579	0.0
Robot Tron	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0
placebo Action Adventure	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	1.0	0.0	0.0	0.000000	0.0
Bones (Chao)	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0
JETPAC (itch)	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.000000	0.0	0.0	0.000000	0.0	0.0	0.0	0.0	0.000000	0.0

Gambar 2. Hasil Penghitungan TF-IDF

Dapat dilihat bahwa sebuah *video game* bernama Motor Money Crash! memiliki bobot 1.0 pada genre *shooter*. Maka dari itu Motor Money Crash! merupakan *video game* yang bergenre *shooter*.

### 3.2. Menghitung Cosine Similarity

Setelah mendapatkan matriks TF-IDF, selanjutnya akan dilakukan penghitungan *cosine similarity*. Hasil penghitungan *cosine similarity* akan disimpan ke dalam variabel `cosine_sim`. Akan didapatkan array yang merupakan hasil dari penghitungan *cosine similarity* pada matriks TF-IDF seperti yang dapat dilihat pada gambar 3.

```
array([[1.          , 0.          , 0.25192014, ..., 0.          , 0.          ,
        0.75647584],
       [0.          , 1.          , 0.          , ..., 0.72778573, 0.          ,
        0.          ],
       [0.25192014, 0.          , 1.          , ..., 0.          , 0.          ,
        0.          ],
       ...,
       [0.          , 0.72778573, 0.          , ..., 1.          , 0.          ,
        0.          ],
       [0.          , 0.          , 0.          , ..., 0.          , 0.          ,
        1.          ],
       [0.75647584, 0.          , 0.          , ..., 0.          , 0.          ,
        1.          ]])
```

Gambar 3. Hasil Penghitungan Cosine Similarity pada Matriks TF-IDF

Selanjutnya, akan dibuat dataframe dari variabel `cosine_sim` yang baris dan kolomnya merupakan nama dari *video game*. Dataframe tersebut akan menunjukkan matriks *similarity* antara *video game* yang satu dengan *video game* lainnya. Diambil 5 sampel dari *video game* yang tersedia. Matriks *similarity* dapat dilihat pada gambar 4.

Shape: (16026, 16026)

	name	Doti-Doti no Gaea	Viridescent Dragon: Chapter 2	Breath of Life (Cameron98, rsellars)	Make up and drive	DragonCrash Demo
	Bracewell	0.0	0.000000	0.0	1.0	0.000000
	I Bearly Made It	0.0	0.000000	0.0	0.0	1.000000
	Empathy	0.0	0.540066	0.0	0.0	0.467814
	Citricidel	0.0	0.000000	0.0	0.0	0.614183
	2001 Emo Kids Need Weezer Ozma Saves the Day Tix Plz	0.0	0.000000	0.0	0.0	0.000000

Gambar 4. Matriks Similarity

Dapat dilihat bahwa *video game* bernama Bracewell dan Make Up and Drive memiliki nilai *similarity* 1.0. Maka dari itu, jika pengguna mencari *video game* bernama Bracewell, maka

sistem akan memberikan rekomendasi *video game* dengan genre *action*, dan rekomendasi tersebut akan dimunculkan juga ketika pengguna mencari *video game* Make Up and Drive.

### 3.3. Melakukan Evaluasi Model

Evaluasi model akan dilakukan menggunakan *precision*. *Precision* merupakan sebuah pengujian dasar yang biasanya digunakan untuk menguji sistem rekomendasi [6]. TP atau *true positive* berarti sistem yang menghasilkan informasi relevan, sedangkan FP atau *false positive* berarti sistem yang menghasilkan seluruh *item* yang tersedia. Dengan kata lain, *precision* merupakan pengujian yang membandingkan hasil dari sistem dengan informasi yang diminta oleh pengguna. *Precision* dapat dihitung menggunakan persamaan (4).

$$Precision = \frac{TP}{TP+FP} \times 100\% \quad (4)$$

Keterangan:

TP = informasi relevan yang dihasilkan sistem

FP = keseluruhan *item* yang tersedia

Untuk menghitung *precision* pada sistem rekomendasi *video game*, akan dilakukan pengambilan 10 sampel *video game* yang nilai *precision*-nya akan dijumlahkan kemudian dihitung rata-ratanya. Jika sistem berhasil merekomendasikan banyak *video game* yang memiliki kemiripan dengan *video game* yang dicari oleh pengguna, itu berarti nilai *precision* semakin tinggi. Hasil pengujian dari sistem rekomendasi *video game* dapat dilihat pada tabel 1.

Tabel 1. Hasil Pengujian Sistem

No	Nama <i>Video Game</i>	<i>Precision</i>
1.	Planet R-12	97,5%
2.	Space Engineers	92,5%
3.	Drums Hero	79%
4.	Eating Hearts	100%
5.	Infestation: The New Z	92,5%
6.	City Climber	78%
7.	The Second Shadow (Free Demo)	100%
8.	VR Darts	77%
9.	Toricky	84%
10.	Brainbox	77%
Rata-Rata		87,75%

Pengujian dari sistem rekomendasi *video game* menggunakan metode *content-based filtering* menghasilkan rata-rata *precision* sebesar 87,75%. Nilai rata-rata tersebut menunjukkan bahwa sistem rekomendasi *video game* menggunakan metode *content-based filtering* dapat merekomendasikan *item* sesuai dengan preferensi pengguna.

## 4. Conclusion

Berdasarkan penelitian yang telah dilakukan, sistem rekomendasi *video game* yang telah dibangun memiliki rata-rata *precision* sebesar 87,75% dan sudah berjalan dengan baik. Hal ini membuktikan bahwa sistem rekomendasi menggunakan metode *content-based filtering* dapat memberikan jawaban yang tepat berdasarkan informasi yang pengguna minta sehingga sistem rekomendasi ini dapat membantu para pengguna untuk mencari *video game* dengan genre yang sesuai dengan preferensi mereka.

## References

- [1] N. I. Siregar and P. Muljono, "Pengaruh Perilaku Bermain Video Game Berunsur Kekerasan Terhadap Perilaku Agresi Remaja," *Jurnal Sains Komunikasi dan Pengembangan Masyarakat [JSKPM]*, vol. 1, no. 3, pp. 261-276, 2017.
- [2] R. H. Mondy, A. Wijayanto and Winarno, "RECOMMENDATION SYSTEM WITH CINTENT-BASED FILTERING METHOD FOR CULINARY TOURISM IN MANGAN APPLICATION," *ITSMART: Jurnal Ilmiah Teknologi dan Informasi*, vol. 8, no. 2, pp. 65-72, 2019.
- [3] M. R. A. Zayyad, *Sistem Rekomendasi Buku Menggunakan Metode Content Based Filtering*, Yogyakarta, 2021.
- [4] M. W. Putri, A. Muchayan and M. Kamisutara, "Sistem Rekomendasi Produk Pena Eksklusif Menggunakan Metode Content-Based Filtering dan TF-IDF," *JOINTECS (Journal of Information Technology and Computer Science)*, vol. 3, no. 1, pp. 229-236, 2020.
- [5] P. Nastiti, "Penerapan Metode Content-Based Filtering Dalam Implementasi Sistem Rekomendasi Tanaman Pangan," *TEKNIKA*, vol. 8, no. 1, pp. 1-10, 2019.
- [6] W. G. S. Parwita, "Pengujian Akurasi Sistem Rekomendasi Berbasis Content-Based Filtering," *Jurnal Ilmiah Ilmu Komputer*, vol. 14, no. 1, pp. 27-32, 2019.