

Perbandingan *User-Based* dan *Item-Based* pada Sistem Rekomendasi Film Kombinasi Teknik Reduksi Dimensi dan Clustering

RIZKI ASHURI PRATAMA, YUNUS SAFI'I, MAULIDHAN ADY NUGRAHA, ANIS SATUS SOBIHAH, NOOR IFADA

Program Studi Teknik Informatika, Universitas Trunojoyo,, Madura, Indonesia
Email: noor.ifada@trunojoyo.ac.id

ABSTRAK

Sistem rekomendasi mampu menghasilkan daftar film hasil personalisasi yang mungkin menarik bagi *user* dengan mempelajari kegiatan *user* dalam memberikan *rating*. Sistem rekomendasi diklasifikasikan dalam tiga pendekatan: *Content-Based Filtering*, *Collaborative Filtering (CF)*, dan *Hybrid Filtering*. Pendekatan CF lebih *popular* dibandingkan dua pendekatan lainnya. CF memiliki dua model, yakni CF *user-based* (UB) dan CF *item-based* (IB). Namun, pada CF terdapat permasalahan yaitu waktu komputasi yang lama karena dimensi data yang besar, kelangkaan data dan akurasinya. Untuk mengatasinya terdapat dua tahap yang dapat dikombinasikan pada CF, yaitu reduksi dimensi menggunakan algoritma *Singular Value Decomposition* (SVD) dan *clustering* menggunakan algoritma K-Means (KM). Tujuan dari penelitian ini adalah melakukan perbandingan hasil akurasi antara sistem rekomendasi film yang menggunakan metode SVD-KM-UB dan SVD-KM-IB pada *dataset* MovieLens. Hasil yang didapatkan pada *dataset* MovieLens, metode SVD-KM-UB lebih unggul daripada metode SVD-KM-IB. Metode SVD-KM-UB mengalami persentase kenaikan pada seluruh variasi *Top-N* dengan peningkatan terbesar pada *Top-20*, yaitu sebesar 5836,4%.

Kata kunci: sistem rekomendasi, collaborative filtering, k-means, SVD

ABSTRACT

Recommendation systems can generate a list of personalized movies that might interest users by studying their rating activities. Recommendation systems are classified into three approaches: Content-Based Filtering, Collaborative Filtering (CF), and Hybrid Filtering. CF is more popular than the other two approaches. CF has two models, namely CF user-based (UB) and CF item-based (IB). However, problems encountered in CF, i.e., long computation time due to large data dimensions, data scarcity and accuracy. To overcome this, there are two stages that can be combined in CF, namely dimension reduction using the Singular Value Decomposition (SVD) algorithm and clustering using the K-Means (KM) algorithm. The purpose of this research is to compare the performances between SVD-KM-UB and SVD-KM-IB methods on movie recommendation systems. Experiment results show that SVD-KM-UB is superior compared to SVD-KM-IB. SVD-KM-UB achieves significant percentage increase at any Top-N where the biggest increase is 5836.4% at Top-20.

Keywords: recommendation system, collaborative filtering, k-means, SVD

1. PENDAHULUAN

Di era yang sudah kaya akan informasi ini, masyarakat tidak lagi kesulitan dalam mencari berbagai informasi. Namun, banyaknya informasi di internet menjadikan sebuah tantangan bagi manusia untuk menemukan informasi yang sesuai dengan kebutuhan secara cepat (**Dwicahya et al., 2019**). Termasuk juga pada informasi mengenai film. Film menjadi salah satu hiburan bagi masyarakat di saat waktu senggang. Film memiliki jumlah penambahan yang drastis setiap tahunnya. Banyaknya film terkadang membuat masyarakat bingung saat akan memilih. Namun, ada beberapa masyarakat yang sudah memiliki film favorit. Tidak juga pada masyarakat biasa yang hanya menonton film tanpa memiliki film favorit. Sistem rekomendasi adalah salah satu solusi untuk menyelesaikan permasalahan ini (**Ifada, Fitriantama, et al., 2020**).

Sistem rekomendasi membantu *user* dalam memilih informasi yang dibutuhkan dari banyaknya informasi yang tersedia. Sistem rekomendasi mampu menghasilkan satu set daftar film hasil personalisasi yang mungkin menarik bagi *user* dengan mempelajari kegiatan *user* dalam memberikan *rating* (**Chen et al., 2018**). Sistem ini dapat memprediksi *rating* user terhadap film yang kemudian digunakan untuk menghasilkan rekomendasi film (**Ifada, Rahman, et al., 2020**). Film tentunya sangat bervariasi mulai dari *genre* yang tiap *user* pasti memiliki perbedaan. Sistem rekomendasi film bertujuan untuk memberikan rekomendasi film yang diminati untuk *user*.

Sistem rekomendasi diklasifikasikan dalam tiga pendekatan yakni *Content-Based Filtering*, *Collaborative Filtering* (CF), dan *Hybrid Filtering* (combination). *Content-Based Filtering* merekomendasikan film berdasarkan kesamaan konten film, namun pendekatan ini secara alami hanya relevan ketika adanya informasi film (**Salloum & Rajamanthri, 2021; Thakker et al., 2021**). CF merekomendasikan film dengan menggunakan prinsip *user* yang memiliki persamaan menyukai terhadap film yang serupa dan film yang serupa disukai oleh *user* yang serupa (**Chen et al., 2018; Lops et al., 2011**). *Hybrid Filtering* adalah pendekatan yang menghasilkan rekomendasi dengan mengkombinasikan *Content-Based Filtering* dan CF, tujuannya adalah untuk meningkatkan akurasi dan efisiensi proses rekomendasi (**Salloum & Rajamanthri, 2021; Thorat et al., 2015**).

Pendekatan CF merupakan pendekatan yang lebih popular dibandingkan pendekatan *Content-Based Filtering* dan *Hybrid Filtering* (**Aggarwal, 2016; Konstan & Riedl, 2012; Thakker et al., 2021**). CF memiliki dua model yakni, *user-based* (UB) dan *item-based* (IB) (**Ifada et al., 2018**). UB memprediksi *rating* yang diberikan oleh *user* ke film berdasarkan *rating* yang diberikan oleh *user* serupa lainnya untuk film yang sama. Sedangkan IB memprediksi *rating* yang diberikan *user* terhadap suatu film berdasarkan *rating* yang diberikan oleh *user* lain terhadap film yang serupa. *User* atau film serupa lainnya disebut tetangga. UB dan IB mengandalkan tetangga terdekat untuk memprediksi *rating*. Perbedaan antara UB dan IB adalah jenis tetangganya.

Permasalahan yang umum terjadi pada CF diantaranya adalah kompleksitas atau waktu komputasi yang lama karena dimensi data yang besar, kelangkaan data dan akurasinya (**Ifada, Rahman, et al., 2020**). Dalam mengatasi permasalahan tersebut terdapat penelitian yang menggunakan teknik penambangan data sebagai kombinasi dari pendekatan CF yang terdiri dari dua tahap yaitu reduksi dimensi dan *clustering* (**Chen et al., 2018**).

Untuk tahap yang pertama yaitu reduksi dimensi menggunakan algoritma *Singular Value Decomposition* (SVD) (**Son et al., 2017**). SVD dapat membantu pada komputasi waktu yang lama karena kelebihannya dalam mengefisiensi waktu proses untuk data dengan skala besar

(Zarzour et al., 2018). Sedangkan pada tahap *clustering* akan menggunakan algoritma K-Means. Algoritma K-Means merupakan algoritma *clustering* yang umum digunakan karena mudah diimplementasikan dengan melakukan pengelompokan data berdasarkan titik pusat *cluster* yang paling dekat dengan data. K-Means dapat membantu mengelompokkan data dengan meminimalkan kesamaan data antar *cluster* dan memaksimalkan kesamaan data dalam satu *cluster* (Kuswandi et al., 2018).

Tujuan dari penelitian ini adalah melakukan perbandingan hasil akurasi antara sistem rekomendasi film yang menggunakan pendekatan UB dan IB dengan kombinasi penerapan teknik reduksi dimensi dan teknik *clustering*.

2. METODE

Berikut merupakan metode yang akan dilakukan pada penelitian ini:

2.1. Pengumpulan Data

Data yang akan digunakan pada penelitian ini merupakan data *rating* film yang didapatkan dari MovieLens yang bisa didapatkan dari website GroupLens (<https://files.grouplens.org/datasets/movielens/ml-100k.zip>). Jumlah data yang didapatkan adalah sebanyak 100.000 *rating* dengan 1.682 film yang diberikan *rating* oleh 943 *user* yang digunakan untuk membuat matriks *rating* $R \in \mathbb{R}^{943 \times 1682}$ (Harper & Konstan, 2015).

2.2. Rancangan Sistem

Pada penelitian ini akan membandingkan dua metode. Metode pertama merupakan kombinasi SVD (*user*), K-Means (*user*) dan UB yang disebut dengan SVD-KM-UB. Metode kedua merupakan kombinasi SVD (*item*), K-Means (*item*) dan IB yang disebut dengan SVD-KM-IB. Pada metode SVD-KM-UB algoritma K-Means (*user*) akan menggunakan matriks U yang dihasilkan algoritma SVD (*user*) untuk melakukan *clustering user*, sedangkan pada metode SVD-KM-IB algoritma K-Means (*item*) akan menggunakan matriks V^T yang dihasilkan algoritma K-Means (*item*) untuk melakukan *clustering item*.

Gambar 1 merupakan kerangka kerja metode yang digunakan pada penelitian ini, yaitu metode SVD-KM-UB dan SVD-KM-IB dengan K terbaik untuk K-Means dan F (tetangga terdekat) terbaik untuk pendekatan CF. Pada penelitian ini menggunakan *input* berupa data matriks *rating* dari MovieLens, selanjutnya akan dilakukan *process* pada masing-masing metode dan menghasilkan *output* prediksi *ranking* rekomendasi *Top-N* film untuk masing-masing metode.

2.2.1. Reduksi dimensi

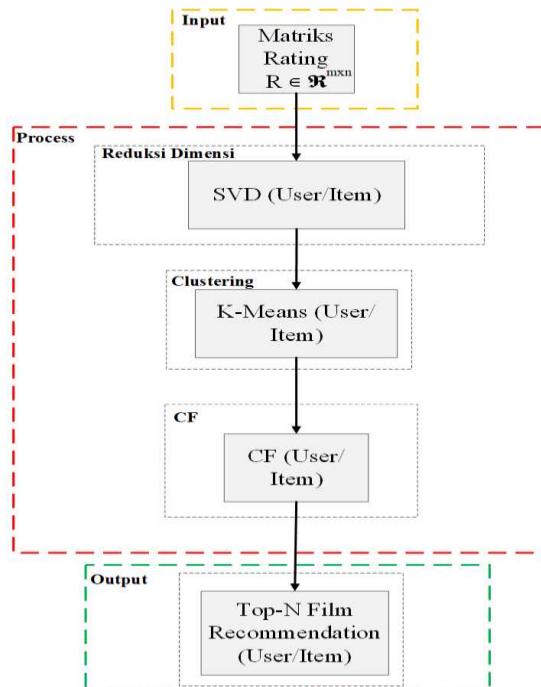
Reduksi dimensi merupakan sebuah tahapan untuk menghasilkan variabel acak yang telah dipertimbangkan. Algoritma SVD merupakan salah satu algoritma yang ada pada reduksi dimensi. SVD adalah algoritma aljabar *linier* yang digunakan untuk menguraikan suatu matriks menjadi tiga matriks baru, yaitu matriks *orthogonal* U , matriks *diagonal* S , dan matriks *transpose orthogonal* V seperti pada Persamaan (1) (Qasem and Mohammadzadeh 2021). Algoritma SVD dapat dilihat pada Gambar 2.

$$A_{m \times n} = U_{m \times m} \times S_{m \times n} \times V^T_{n \times n} \quad (1)$$

Dimana:

- $A_{m \times n}$: Matriks A dengan nilai $m > n$
 $U_{m \times m}$: Matriks U dengan ukuran $m \times m$

- $S_{m \times n}$: Matiks *diagonal* dengan ukuran $n \times n$ dengan elemen matriks positif atau nol
 $V_{n \times n}^T$: Matriks *orthogonal* dengan ukuran $n \times n$ dimana hasil dari *transpose* matriks V



Gambar 1. Kerangka kerja metode

Algoritma SVD

Input: Matriks rating R

Tahap:

1. Melakukan *transpose* pada matriks A^T
 2. Membentuk matriks $A^T \times A$
 3. Menghitung nilai *eigen* dari $A^T \times A$
 4. Menghitung nilai singular value: $S_n = \sqrt{C_n}$
 5. Membuat matriks *diagonal* dari nilai (S) untuk membentuk matriks S
- $$S = \begin{bmatrix} S_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & S_n \end{bmatrix}$$
6. Menghitung *vector eigen* menggunakan nilai *eigen* yang sudah didapat untuk membentuk matriks
 7. Melakukan *transpose* pada matriks V untuk mendapatkan nilai *right eigen vector* V^T

Output: Matriks U, Matriks S dan Matriks V^T

Gambar 2. Algoritma SVD

2.2.2. Clustering

Clustering merupakan proses pengelompokan *user/item* ke dalam *cluster* berdasarkan suatu kemiripan yang dimiliki (**Ifada et al., 2019**). Algoritma K-Means merupakan salah satu algoritma *clustering* untuk mengelompokkan sejumlah N data menjadi K kelompok dengan

meminimalkan jumlah jarak kuadrat antara setiap titik dengan pusat kelompok terdekatnya (*centroid*) (**Fränti & Sieranoja, 2018**). Angka K didefinisikan secara apriori atau berdasarkan dugaan awal secara *random* (apriori). Setiap kelompok digambarkan dengan pusat kelompok pada ruang fitur (**Károly et al., 2018**).

Perhitungan jarak antara objek dengan *centroid* dapat dilakukan dengan rumus *Euclidian Distance* menggunakan Persamaan (2).

$$D_{(i,j)} = \sqrt{(x_{1i} - x_{1j})^2 + (x_{2i} - x_{2j})^2 + \dots + (x_{ki} - x_{kj})^2} \quad (2)$$

Dimana:

- (i,j) : Jarak data i ke pusat *cluster* j
- x_{ki} : Data ke- i pada atribut k
- x_{kj} : Pusat centroid j pada atribut k

Dalam memperbarui setiap *centroid* dilakukan dengan menggunakan Persamaan (3).

$$\bar{V}_{lj} = \frac{1}{N_i} \sum_{k=0}^n X_{kj} \quad (3)$$

Dimana:

- \bar{V}_{lj} : Centroid dari rata-rata cluster i untuk variabel j
- N_i : Jumlah data yang menjadi anggota *cluster*
- X_{kj} : Nilai k yang ada di *cluster* untuk variabel j
- j : Indeks variabel

Algoritma untuk K-Means dapat dilihat pada Gambar 3.

Algoritma K-Means

Input: Jumlah *cluster* dan matriks U untuk SVD-KM-UB dan matriks V^T untuk SVD-KM-IB

Tahap:

1. Menginisialisasi titik *centroid* secara acak
2. Menghitung jarak setiap film ke setiap titik *centroid* dengan menggunakan Persamaan (2), menetapkan setiap film ke *centroid* terdekatnya
3. Memperbarui setiap *centroid* dengan mengambil rata-rata poin film ditugaskan ke *cluster* terbaik menggunakan Persamaan (3)
4. Mengulangi langkah 2 dan 3 sampai konvergen

Output: K cluster

Gambar 3. Algoritma K-Means

2.2.3. Collaborative Filtering (CF) dengan *User-Based* (UB)

Collaborative dengan UB menghitung prediksi *rating* berdasarkan kemiripan *user*. Kemiripan antar *user* dapat dihitung menggunakan fungsi *Pearson Similarity* karena memiliki hasil yang terbaik untuk menghitung kemiripan *rating user* (**Ifada et al., 2019**). Persamaan (4) merupakan fungsi *Pearson Similarity* untuk menghitung kemiripan antar *user u* dan *v*:

$$\text{Sim } U(u, v) = \frac{\sum_{k \in I_u \cap I_v} (r_{uk} - \hat{\mu}_u) \cdot (r_{vk} - \hat{\mu}_v)}{\sqrt{\sum_{k \in I_u \cap I_v} (r_{uk} - \hat{\mu}_u)^2} \cdot \sqrt{\sum_{k \in I_u \cap I_v} (r_{vk} - \hat{\mu}_v)^2}} \quad (4)$$

Dimana:

- r_{uk} : Rating dari user u terhadap film k
- r_{vk} : Rating dari user v terhadap film k
- $\hat{\mu}_u$: Rata-rata rating user u
- $\hat{\mu}_v$: Rata-rata rating user v

Untuk menghitung prediksi rating user u terhadap film i berdasarkan *user-based* dapat menggunakan Persamaan (5):

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in S_u(i)} r_{vi} \cdot \text{Sim } U(u, v)}{\sum_{v \in S_u(i)} |\text{Sim } U(u, v)|} \quad (5)$$

Dimana:

- $S_u(i)$: Top- F tetangga terdekat user u mengenai item i

Daftar rekomendasi *Top-N* untuk target user u , $\text{Top}_u(N)$ dibuat berdasarkan pengurutan besar ke kecil dari daftar prediksi rating (Ifada et al., 2018). Berikut merupakan Persamaan (6) untuk penentuan *Top-N*:

$$\text{Top}_u(N) := \underset{i \in \hat{I}_u}{\underset{\hat{r}_{uj}}{\operatorname{argmax}}}^k \quad (6)$$

Dimana:

- \hat{I}_u : Daftar film yang belum diberikan rating oleh target user u
- \hat{r}_{uj} : Prediksi rating user u memberikan rating ke film j

2.2.3. Collaborative Filtering (CF) dengan Item-Based

CF merekomendasikan item dengan menggunakan prinsip user yang memiliki persamaan menyukai terhadap item yang serupa dan item yang serupa disukai oleh user yang serupa (**Chen et al., 2018**). Pada pendekatan CF memori dibagi menjadi 2 model yakni, UB dan IB. UB akan menghitung kemiripan user dengan minat user target, kemudian merekomendasikan film favorit user pada user lain yang memiliki minat serupa. Sedangkan IB akan merekomendasikan user ke film yang mirip dengan yang disukai sebelumnya (**Ifada et al., 2019**).

IB memprediksi rating yang diberikan user terhadap suatu film berdasarkan rating yang diberikan oleh user lain terhadap film yang serupa. Dalam penelitian ini kemiripan antar film dihitung menggunakan fungsi *Adjusted Cosine Similarity*, karena terbaik dalam menghitung kemiripan film (**Ifada et al., 2019**). Persamaan (7) untuk menghitung similaritas dengan *Adjusted Cosine Similarity*:

$$\text{Sim } I(i,j) = \frac{\sum_{u \in U_i \cap U_j} (r_{ui} - \hat{\mu}_u) \cdot (r_{uj} - \hat{\mu}_u)}{\sqrt{\sum_{u \in U_i \cap U_j} (r_{ui} - \hat{\mu}_u)^2} \sqrt{\sum_{u \in U_i \cap U_j} (r_{uj} - \hat{\mu}_u)^2}} \quad (7)$$

Dimana:

r_{uj} : Rating dari user u terhadap film j

r_{ui} : Rating dari user u terhadap film i

$\hat{\mu}_u$: Rata-rata rating dari user u

Menghitung prediksi *rating* dari *user* untuk produk dengan Persamaan (8).

$$\hat{r}_{uj} = \frac{\sum_{i \in \text{RatedItems}(u)} \text{sim}(i,j) * r_{ui}}{\sum_{i \in \text{Ratealitems}(j)} |\text{sim}(i,j)|} \quad (8)$$

Dimana:

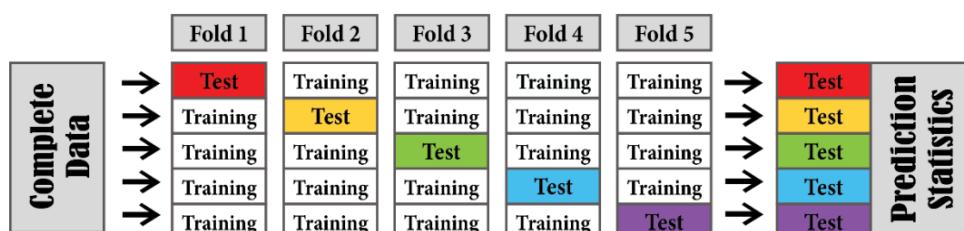
\hat{r}_{uj} : Prediksi *rating* user u memberikan *rating* ke film j

$\text{sim}(i,j)$: Kemiripan user i dengan user j

r_{ui} : Rating user u terhadap film i

2.3. Evaluasi Prediksi *Ranking* Rekomendasi *Top-N*

Metode evaluasi digunakan untuk menguji hasil dari penelitian. Pada penelitian ini metode evaluasi yang akan digunakan adalah *k-Fold Cross Validation* dengan nilai $k = 5$ yang membagi data menjadi data *training* sebesar 80% dan data *test* sebesar 20%. Gambar 4 menunjukkan skema atau gambaran kerja *5-Fold Cross Validation* dalam melakukan pembagian data menjadi data *training* sebesar 80% dan data *test* sebesar 20%.



Gambar 4. Skema 5-Fold Cross Validation

Performa dari sistem rekomendasi dalam menentukan *Top-N* atau pemberian ranking dapat dievaluasi menggunakan Precision, Recall, F1-Score, Average Precision (AP), Mean Average Precision (MAP), Discounted Cumulative Gain (DCG), dan Normalized Discounted Cumulative Gain (NDCG) (**Sivanaiah et al., 2020**).

Pada sistem temu kembali informasi, *Discounted Cumulative Gain* (DCG) merupakan teknik lain yang digunakan untuk mengukur kualitas dari *ranking* rekomendasi *Top-N*. DCG merupakan jumlah bobot dari tingkat relevansi film yang diberi *ranking*. *Normalized Discounted Cumulative Gain* (NDCG) adalah normalisasi dari DCG. NDCG digunakan sebagai ukuran relevansi hasil yang dihasilkan oleh model yang dibangun (**Sivanaiah et al., 2020**). Persamaan (9) untuk menentukan nilai DCG, Persamaan (10) untuk menentukan nilai IDCG, sedangkan Persamaan (11) untuk menentukan nilai NDCG:

$$DCG(\text{Test}_u, I_u, N) := \sum_{n=1}^N \frac{1}{\log_2(1+n)} \times \mathbb{I}(I_u(n) \in \text{Test}_u) \quad (9)$$

$$IDCG(N) := \sum_{n=1}^N \frac{1}{\log_2(1+n)} \quad (10)$$

$$NDCG(D_{\text{test}}, N) := \frac{DCG(\text{Test}_u, I_u, N)}{IDCG(N)} \quad (11)$$

Dimana:

$DCG(\text{Test}_u, I_u, N)$: Nilai DCG setiap target user u
N	: Posisi N teratas
Test_u	: film yang ada di data $test$ untuk target $user u$
\hat{I}_u	: Kumpulan film yang diprediksi untuk target $user u$
D_{test}	: Data $user$ yang terdapat pada data $test$

3. HASIL DAN PEMBAHASAN

Penelitian ini membuat perbandingan akurasi antara SVD-KM-UB dan SVD-KM-IB pada data MovieLens.

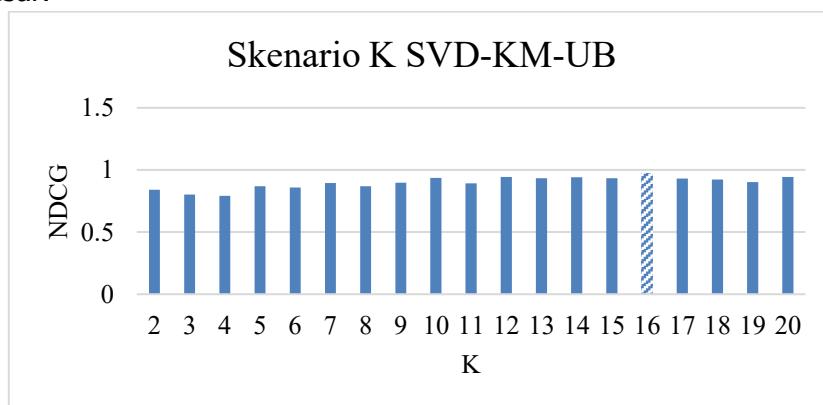
3.1. Hasil Eksperimen

Pada penelitian ini melakukan uji skenario untuk mendapatkan K terbaik dan F terbaik pada masing-masing metode SVD-KM-UB dan SVD-KM-IB.

3.1.1. Uji Skenario pada SVD-KM-UB

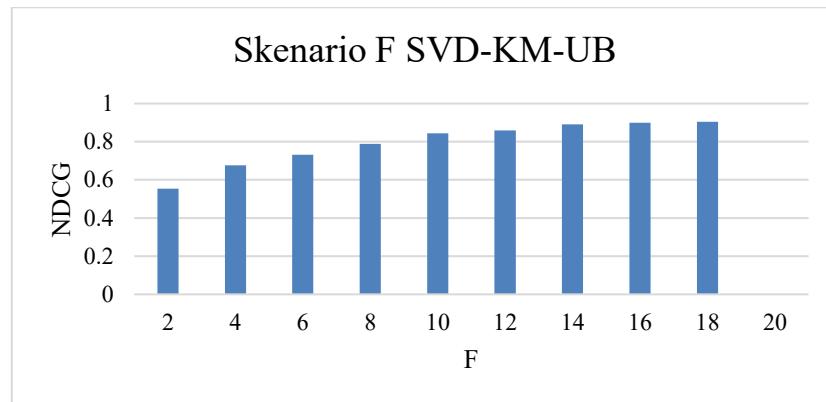
Pada SVD-KM-UB terdapat tahapan mencari K terbaik untuk K-Means dan F terbaik untuk pendekatan CF berdasarkan *Top-20*:

- Skenario K . Skenario K digunakan untuk menemukan nilai K terbaik untuk SVD-KM-UB. Nilai K merupakan banyaknya cluster yang akan dibentuk pada teknik KM berdasarkan Persamaan (3) dan Persamaan (2). Pada penelitian ini menggunakan skenario $K = \{2, 3, 4, \dots, 18, 19, 20\}$ dan K terbaik didapat dengan membandingkan rata-rata setiap nilai NDCG pada 5-fold. Berdasarkan Gambar 5, Nilai K terbaik untuk SVD-KM-UB adalah 16, yaitu sebesar 0,972. Semakin besar nilai K belum tentu menghasilkan nilai NDCG yang semakin besar.



Gambar 5. Skenario K SVD-KM-UB

- Skenario F . Skenario F digunakan untuk menemukan jumlah *neighbour* terbaik untuk SVD-KM-UB. Nilai F merupakan banyaknya *Top-F* tetangga terdekat yang digunakan untuk menghitung prediksi pada Persamaan (5). Penelitian ini menggunakan skenario $F = \{2,4,6,\dots,16,18,20\}$ dan F terbaik didapat dengan membandingkan rata-rata setiap NDCG pada 5-fold. Berdasarkan Gambar 6, nilai F terbaik untuk metode SVD-KM-UB adalah 20, yaitu sebesar 0,925. Semakin besar nilai F maka semakin besar pula nilai NDCG yang dihasilkan.

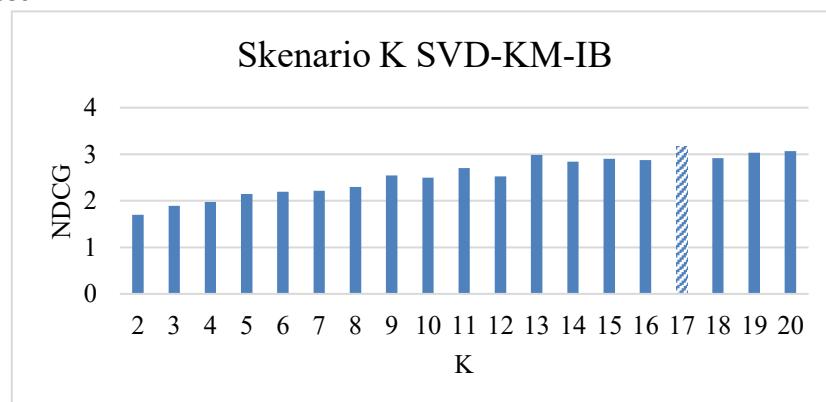


Gambar 6. Skenario F SVD-KM-UB

3.1.2. Uji Skenario pada SVD-KM-IB

Pada CF IB terdapat tahapan mencari K terbaik F terbaik untuk pendekatan CF berdasarkan *Top-20*:

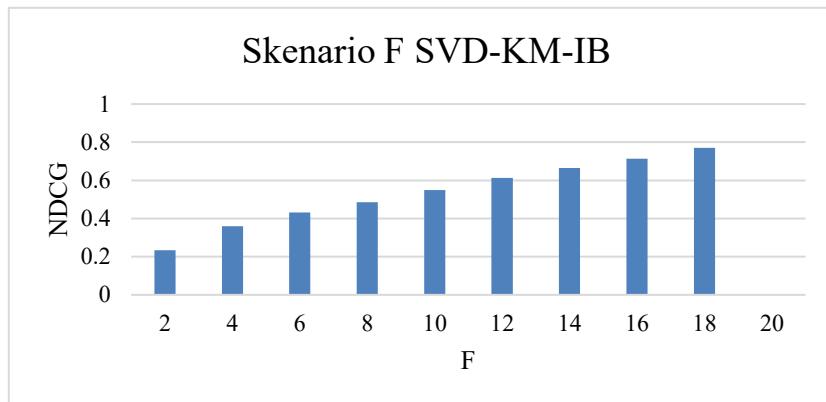
- Skenario K . Skenario K digunakan untuk menemukan nilai K terbaik untuk SVD-KM-IB. Nilai K merupakan banyaknya *cluster* yang akan dibentuk pada teknik KM berdasarkan Persamaan (3) dan Persamaan (2). Pada penelitian ini menggunakan skenario $K = \{2,3,4,\dots,18,19,20\}$ dan K terbaik didapat dengan membandingkan rata-rata setiap nilai NDCG pada 5-fold. Berdasarkan Gambar 7, Nilai K terbaik untuk SVD-KM-IB adalah 17, yaitu sebesar 3,168. Semakin besar nilai K belum tentu menghasilkan nilai NDCG yang semakin besar.



Gambar 7. Skenario K SVD-KM-IB

- Skenario F . Skenario F digunakan untuk menemukan jumlah *neighbour* terbaik untuk SVD-KM-IB. Nilai F merupakan banyaknya *Top-F* tetangga terdekat yang digunakan untuk

menghitung prediksi pada Persamaan (5). Penelitian ini menggunakan skenario $F = \{2,4,6,\dots,16,18,20\}$ dan F terbaik didapat dengan membandingkan rata-rata setiap NDCG pada 5-fold. Berdasarkan Gambar 8, nilai F terbaik untuk metode SVD-KM-IB adalah 20, yaitu sebesar 0,797. Semakin besar nilai F maka semakin besar pula nilai NDCG yang dihasilkan.

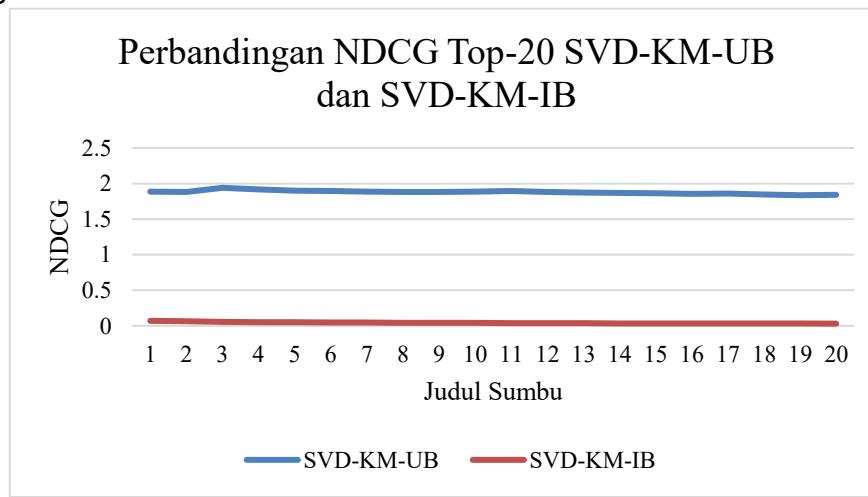


Gambar 8. Skenario F untuk SVD-KM-IB

3.2. Perbandingan Metode SVD-KM-UB dan SVD-KM-IB

Berdasarkan hasil eksperimen pada SVD-KM-UB dan SVD-KM-IB maka didapatkan model untuk metode SVD-KM-UB dan metode SVD-KM-IB. Model SVD-KM-UB menggunakan nilai $K = 16$ dan $F = 20$. Model SVD-KM-IB menggunakan nilai $K = 17$ dan $F = 20$. Selanjutnya dilakukan perbandingan antara kedua metode menggunakan modelnya masing-masing.

- Perbandingan nilai NDCG variasi $Top-N$ SVD-KM-UB dan SVD-KM-IB. Model yang dihasilkan pada masing-masing metode diuji dengan matriks NDCG berdasarkan $Top-1$ hingga $Top-20$. Gambar 9 menunjukkan bahwa metode SVD-KM-UB secara keseluruhan $Top-N$ selalu unggul dibandingkan metode SVD-KM-IB.



Gambar 9. Perbandingan NDCG Top-N SVD-KM-UB dan SVD-KM-IB

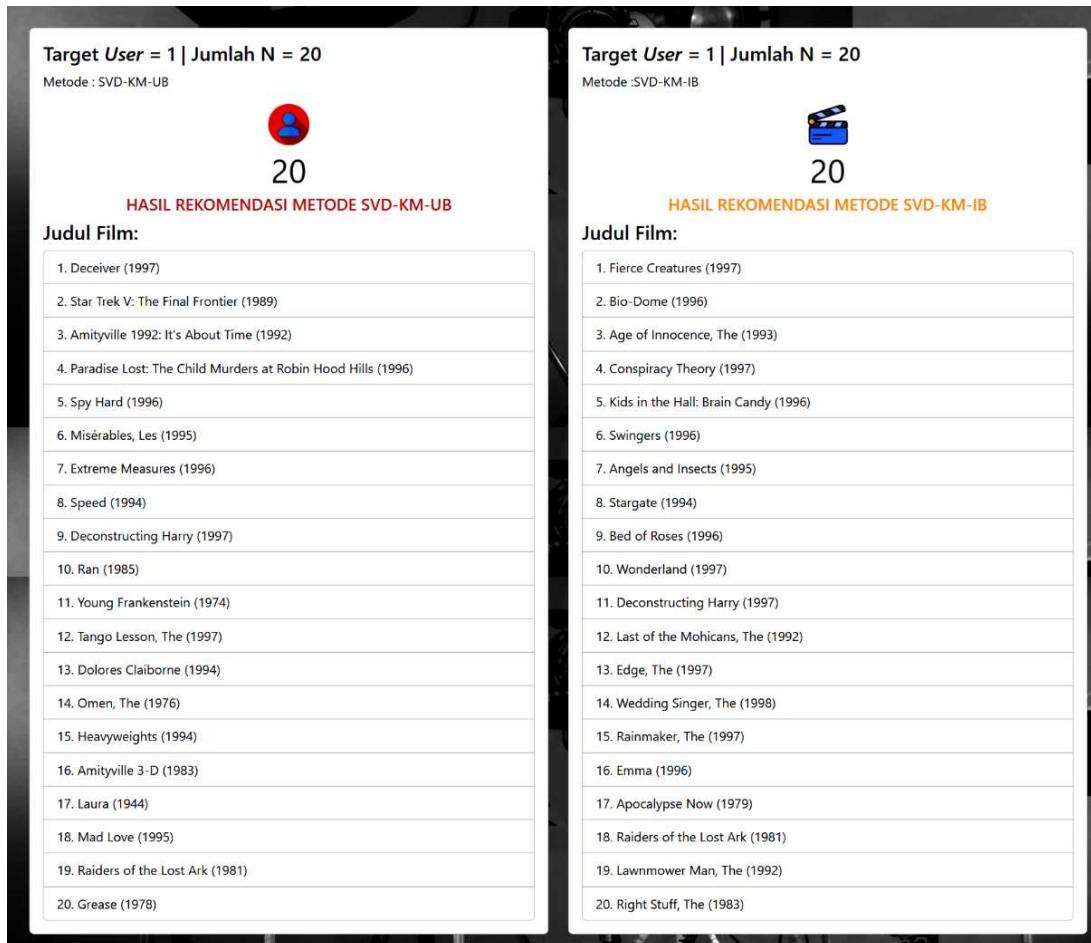
- Persentase kenaikan nilai NDCG variasi $Top-N$ SVD-KM-UB terhadap SVD-KM-IB. Model yang dihasilkan diuji menggunakan metric NDCG berdasarkan $Top-1$ hingga $Top-20$. Tabel

1 menunjukkan persentase kenaikan nilai NDCG *Top-1* hingga *Top-20* SVD-KM-UB terhadap SVD-KM-IB.

Tabel 1. Persentase kenaikan NDCG *Top-N* SVD-KM-UB dan SVD-KM-IB

Top-N	NDCG		Persentase Kenaikan
	SVD-KM-UB	SVD-KM-IB	
@1	1,888	0,071	2577,223%
@2	1,883	0,063	2879,096%
@3	1,940	0,057	3279,883%
@4	1,916	0,053	3488,147%
@5	1,899	0,050	3704,880%
@6	1,894	0,048	3885,220%
@7	1,888	0,046	4019,528%
@8	1,881	0,044	4185,278%
@9	1,882	0,042	4340,845%
@10	1,886	0,041	4532,697%
@11	1,896	0,039	4717,799%
@12	1,879	0,038	4836,304%
@13	1,874	0,037	4989,938%
@14	1,867	0,036	5125,933%
@15	1,864	0,035	5261,228%
@16	1,854	0,034	5376,455%
@16	1,857	0,033	5512,683%
@17	1,847	0,032	5606,104%
@18	1,836	0,032	5693,915%
@19	1,839	0,031	5836,415%
@20	1,888	0,071	2577,223%

- Perbandingan hasil rekomendasi film metode SVD-KM-UB dan SVD-KM-IB. Sistem rekomendasi film yang dibangun dengan metode SVD-KM-UB dan SVD-KM-IB akan menampilkan hasil rekomendasi film berdasarkan target *user* dan *Top-N* film. Gambar 10 merupakan contoh hasil rekomendasi film untuk target *user* 1 dengan *Top-20* dari masing-masing metode dan menunjukkan hasil yang berbeda. Dapat dilihat film dengan judul “Deconstructing Harry (1997)” dalam SVD-KM-UB berada pada peringkat 9, sedangkan pada SVD-KM-IB berada pada peringkat 11. Hal yang sama terjadi pada film berjudul “Raiders of the Lost Ark” pada SVD-KM-UB berada pada peringkat 19, sedangkan pada SVD-KM-IB berada pada peringkat 18.



Gambar 10. Graphic User Interface (GUI) Perbandingan Hasil Rekomendasi Film Metode SVD-KM-UB dan SVD-KM-IB

4. KESIMPULAN

Setelah dilakukan skenario eksperimen dan penelitian maka dapat disimpulkan bahwa model pada metode SVD-KM-UB menggunakan nilai $K = 16$ untuk K-Means dan $F = 20$ untuk pendekatan CF. Sedangkan model untuk metode SVD-KM-IB adalah $K = 17$ dan $F = 20$. Pada perbandingan nilai NDCG variasi $Top-N$ untuk model metode SVD-KM-UB dan SVD-KM-IB dengan menggunakan *dataset* MovieLens, diketahui bahwa metode SVD-KM-UB secara keseluruhan $Top-N$ selalu unggul dibandingkan metode SVD-KM-IB dengan persentase kenaikan terbesar terdapat pada $Top-20$, yaitu sebesar 5836,4%. Kesimpulannya, sistem rekomendasi film yang menggunakan metode SVD-KM-UB lebih unggul dibandingkan metode SVD-KM-IB pada *dataset* MovieLens.

UCAPAN TERIMA KASIH

Penelitian ini adalah hasil kegiatan Program Kreativitas Mahasiswa skema Artikel Ilmiah (PKM-AI) 2023 yang diselenggarakan oleh Direktorat Jenderal Pendidikan Tinggi, Riset, dan Teknologi (DIKTI) melalui Direktorat Pembelajaran dan Kemahasiswaan (Belmawa).

DAFTAR RUJUKAN

- Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Switzerland: Springer International Publishing.
- Chen, R., Hua, Q., Chang, Y.-S., Wang, B., Zhang, L., & Kong, X. (2018). A survey of collaborative filtering-based recommender systems: From traditional methods to hybrid methods based on social networks. *IEEE Access*, 6, 64301-64320.
- Dwicahya, I., Rosa, P. H., & Nugroho, R. (2019). *Movie recommender system comparison of user-based and item-based collaborative filtering systems*. Paper presented at the *1st International Conference on Science and Technology for an Internet of Things*, Yogyakarta, Indonesia, (pp. 1-6).
- Fränti, P., & Sieranoja, S. (2018). K-means properties on six clustering benchmark datasets. *Applied Intelligence*, 48, 4743-4759.
- Harper, F., & Konstan, J. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, 5(4), 1-19.
- Ifada, N., Fitriantama, M. N., Sophan, M. K., & Wahyuni, S. (2020). *Collaborative Filtering Item Recommendation Methods based on Matrix Factorization and Clustering Approaches*. Paper presented at the *10th Electrical Power, Electronics, Communications, Controls and Informatics Seminar (EECCIS)*, Malang, Indonesia, (pp. 226-230).
- Ifada, N., Prasetyo, E. H., & Mula'ab. (2018). *Employing sparsity removal approach and Fuzzy C-Means clustering technique on a movie recommendation system*. Paper presented at the *The 2018 International Conference on Computer Engineering, Network and Intelligent Multimedia (CENIM)*, Surabaya, Indonesia, (pp. 157-162).
- Ifada, N., Rahman, T. F., & Sophan, M. K. (2020). *Comparing Collaborative Filtering and Hybrid based Approaches for Movie Recommendation*. Paper presented at the *The 6th Information Technology International Seminar (ITIS)*, Surabaya, Indonesia, 219-223.
- Ifada, N., Susanti, S., & Mula'ab. (2019). Impact of Imputation on Cluster-based Collaborative Filtering Approach for Recommendation System. *Kursor*, 10(1), 13-20.
- Károly, A. I., Fullér, R., & Galambos, P. (2018). Unsupervised clustering for deep learning: A tutorial survey. *Acta Polytechnica Hungarica*, 15(8), 29-53.
- Konstan, J. A., & Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User Modeling and User-Adapted Interaction*, 22(1-2), 101-123.
- Kuswandi, D., Surahman, E., Thaariq, Z. Z. A., & Muthmainnah, M. (2018). *K-Means clustering of student perceptions on project-based learning model application*. Paper presented

at the 2018 4th International Conference on Education and Technology (ICET), Malang, Indonesia, (pp. 9-12).

Lops, P., Gemmis, M., & Semeraro, G. (2011). Content-based Recommender Systems: State of the Art and Trends. In *Recommender Systems Handbook*, (pp. 73-105): Springer US.

Salloum, S., & Rajamanthri, D. (2021). Implementation and evaluation of movie recommender systems using collaborative filtering. *Journal of Advances in Information Technology*, 12(3).

Sivanaiah, R., Sakaya Milton, R., & Mirnalinee, T. T. (2020). Content boosted hybrid filtering for solving pessimistic user problem in recommendation systems. *Intelligent Data Analysis*, 24(6), 1477-1496.

Son, N. T., Dat, D. H., Trung, N. Q., & Anh, B. N. (2017). *Combination of dimensionality reduction and user clustering for collaborative-filtering*. Paper presented at the The 2017 International Conference on Computer Science and Artificial Intelligence (CSAI /17), Jakarta, Indonesia, (pp. 125-130).

Thakker, U., Patel, R., & Shah, M. (2021). A comprehensive analysis on movie recommendation system employing collaborative filtering. *Multimedia Tools and Applications*, 80, 28647-28672.

Thorat, P. B., Goudar, R. M., & Barve, S. (2015). Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications*, 110(4), 31-36.

Zarzour, H., Al-Sharif, Z., Al-Ayyoub, M., & Jararweh, Y. (2018). *A new collaborative filtering recommendation algorithm based on dimensionality reduction and clustering techniques*. Paper presented at the 9th international conference on information and communication systems (ICICS), Irbid, Jordan, (pp. 102-106).