

Analisis Perbandingan Model *Matrix Factorization* dan *K-Nearest Neighbor* dalam Mesin Rekomendasi *Collaborative* Berbasis Prediksi Rating

Janny Eka Prayogo¹, Aries Suharso², Adhi Rizal³

^{1,2,3}Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Singaperbangsa Karawang, Jl. HS. Ronggo Waluyo, Kec. Telukjambe Timur, Karawang, Indonesia, 41361
e-mail: ¹jannyeka.16125@student.unsika.ac.id, ²aries.suharso@unsika.ac.id,
³adhi.rizal@unsika.ac.id

Submitted Date: October 13th, 2020
Revised Date: January 02nd, 2021

Reviewed Date: December 30th, 2020
Accepted Date: January 04th, 2021

Abstract

Rating is a form of assessment of the likes or dislikes of a user or customer for an item. Where the higher the rating number given, the item is preferred by customers or users. In the recommendation engine, a set of ratings can be predicted and used as an object to generate a recommendation by the Collaborative Filtering method. In the Collaborative Filtering method, there is a rating prediction model, namely the Matrix Factorization and K-Nearest Neighbor models. This study analyzes the comparison of the two prediction models based on the value of Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and the prediction results generated using the movielens film rating dataset. From the analysis and testing results, it was found that MAE = 0.6371 and RMSE = 0.8305 for the Matrix Factorization model, while MAE = 0.6742 and RMSE = 0.8863 for the K-Nearest Neighbor model. The best model is Matrix Factorization because the MAE and RMSE values are lower than the K-Nearest Neighbor model and have the closest predicted rating results from the original rating value.

Keywords: Rating; Recommendation Engine; Collaborative Filtering; Matrix Factorization; K-Nearest Neighbor

Abstrak

Rating atau peringkat penilaian merupakan suatu bentuk penilaian suka atau tidak suka dari seorang pengguna atau pelanggan terhadap suatu item. Di mana semakin tinggi angka rating yang diberikan maka item tersebut lebih disukai oleh pelanggan atau pengguna. Di dalam mesin rekomendasi, kumpulan rating dapat diprediksi dan dijadikan sebuah objek untuk menghasilkan sebuah rekomendasi oleh metode *Collaborative Filtering*. Di dalam metode *Collaborative Filtering* terdapat model prediksi rating yaitu model *Matrix Factorization* dan *K-Nearest Neighbor*. Penelitian ini menganalisis perbandingan dari kedua model prediksi yang didasarkan pada nilai *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE) dan hasil prediksi yang dihasilkan dengan menggunakan dataset rating film *movielens*. Dari hasil analisis dan pengujian, didapat MAE = 0.6371 dan RMSE = 0.8305 untuk model *Matrix Factorization*, Sedangkan MAE = 0.6742 dan RMSE = 0,8863 untuk model *K-Nearest Neighbor*. Model terbaik adalah *Matrix Factorization* karena nilai MAE dan RMSE yg lebih rendah dari model *K-Nearest Neighbor* dan memiliki hasil prediksi rating yang paling mendekati dari nilai rating aslinya.

Kata Kunci: Rating; Mesin Rekomendasi; *Collaborative Filtering*; *Matrix Factorization*; *K-Nearest Neighbor*

1. Pendahuluan

Sistem Rekomendasi merupakan suatu sistem yang dapat memberikan sebuah rekomendasi sebuah item melalui penyaringan, pemilihan informasi dengan menggunakan

preferensi dari pengguna berupa profil, peringkat, perilaku untuk membantu suatu individu atau komunitas dalam menemukan konten atau item yang menarik yang kemungkinan besar akan dipilih, digunakan atau dibeli. sedangkan Mesin

Rekomendasi yaitu suatu algoritma terstruktur perhitungan matematika untuk mengidentifikasi dan mempelajari pola pengguna, pola item, dan kaitan antar keduanya untuk memprediksi item yang kemudian akan direkomendasikan (Sorde & Deshmukh, 2015).

Pada Mesin Rekomendasi memiliki dua metode yang sering digunakan, yaitu *content-based filtering* dan *collaborative filtering* (Adellya, Devi, & Tonara, 2015). Di dalam metode *content-based filtering* rekomendasi dihasilkan dari pemanfaatan sebuah hubungan atau relasi antar konten sebuah item dan profil pengguna, di mana konten dari sebuah item itu seperti *author*, *genre*, dll. Namun dalam metode *content-based filtering* memiliki sebuah kelemahan di mana apabila user memiliki lebih dari satu preferensi (suatu hal yang di sukai) maka metode ini tidak mampu melakukan sebuah rekomendasi. Kemudian metode *collaborative filtering* hadir untuk menutupi kelemahan dari *content-based filtering* tersebut. Di dalam metode *collaborative filtering*, rekomendasi dihasilkan berdasarkan prediksi sebuah rating (peringkat nilai) suatu item dari pengguna sebelumnya yang memiliki preferensi yang mirip.

Pada metode *collaborative filtering* terdapat dua model prediksi yang sering digunakan untuk memprediksi rating (peringkat nilai) yang kemudian akan menjadi sebuah rekomendasi. Model tersebut adalah model *matrix factorization* dan *k-nearest neighbor*. *Matrix factorization* atau faktorisasi matriks adalah salah satu teknik yang digunakan dalam sistem rekomendasi. *Matrix factorization* memodelkan interaksi antara pengguna dan item dibalik data historis rating (peringkat nilai) eksplisit dan implisit (Li et al., 2019). *Matrix factorization* mengkaracterkan antara pengguna dan item dengan sebuah vektor dari pola pemberian rating (peringkat nilai) pengguna. nilai keterkaitan yang besar antara pengguna dan item mengarah ke sebuah rekomendasi (Syahrani, Setiawan, & Suryani, 2015). Sedangkan *k-nearest neighbor* merupakan suatu bentuk model klasifikasi yang dapat mengklasifikasikan data berdasarkan jarak terdekat. Model KNN ini juga merupakan salah satu pendekatan yang lebih disukai untuk rekomendasi berbasis metode *collaborative filtering*, karena keuntungan dari *classifier* ini adalah secara konseptual sangat terkait dengan ide *collaborative filtering* yaitu: menemukan pengguna yang berpikiran sama (atau barang serupa), pada dasarnya konsep tersebut sama saja seperti mencari neighbor (tetangga) untuk pengguna atau barang

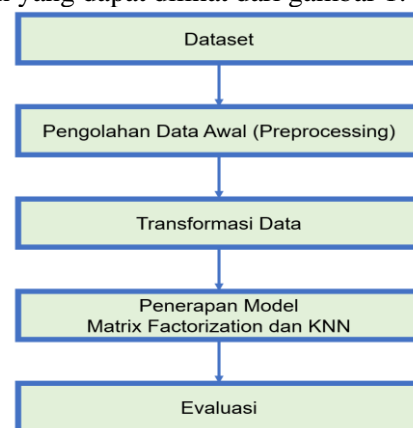
tertentu karena pendekatan ini akan menghitung “distance” antara rating (peringkat nilai) item target dengan item lainnya (Amatriain, Jaimes, Oliver, & Pujol, 2011).

Pada penelitian Sistem Rekomendasi E-Commerce Menggunakan *K-Nearest Neighbor* dalam menentukan top-n Rekomendasi produk untuk setiap pelanggan, menghasilkan Recall sebesar 0,404 pada K=100 menggunakan metode *Collaborative Filtering* (Prasetya, 2017). Sedangkan pada penelitian sistem rekomendasi artikel berita menggunakan metode *k-Nearest Neighbor* berbasis *website* menghasilkan prediksi rating (peringkat nilai) dengan nilai RMSE sebesar 14% atau nilai akurasi sebesar 86% (Abraham & Rahayu, 2017). Lalu pada penelitian rekomendasi buku menggunakan *Tags* dan *Latent* pada *Matrix Factorization* menghasilkan Nilai MAE 0,41 (Krishnamurty, Nurjanah, & Rismala, 2017). Sedangkan pada penelitian sistem rekomendasi E-Commerce menggunakan *Matrix Factorization*, menghasilkan Nilai MAE sebesar 0,0099 pada pembagian 95% data latih dan 5% data uji, kemudian menghasilkan nilai MAE sebesar 0,4743 pada 80% data latih dan 20% data uji (Syahrani et al., 2015).

Berdasarkan penelitian diatas model *Matrix Factorization* dan model *K-Nearest Neighbor* cukup baik untuk memprediksi sebuah peringkat yang kemudian dapat mengarah pada sebuah rekomendasi. Sehingga pada penelitian ini penulis akan berfokus pada analisis perbandingan antara model *matrix factorization* dan *k-nearest neighbor* berdasarkan nilai Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), dan hasil prediksi terbaik yang dihasilkan.

2. Metode Penelitian

Penelitian ini dilakukan dengan beberapa tahapan yang dapat dilihat dari gambar 1.



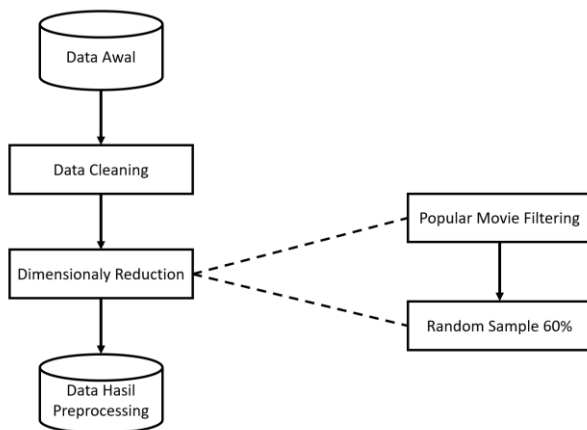
Gambar 1. Tahapan Proses Penelitian

2.1 Dataset

Data yang digunakan pada penelitian ini adalah data rating (peringkat nilai) pada dataset *movielens* yang didapat dari *Grouplens Researcher* <https://grouplens.org/datasets/movielens/>. Di mana terdapat data 100,000 rating oleh 610 *user* terhadap 9742 film.

2.2 Pengolahan Data Awal (Preprocessing)

Data rating (peringkat nilai) yang diperoleh belum bisa langsung diolah begitu saja karena data tersebut masih belum memiliki arti yang jelas (*Unimportant Attribute and Noisy*), sehingga harus dilakukan sebuah pengolahan data awal terlebih dahulu sebelum diproses pada tahap selanjutnya (Alasadi & Bhaya, 2017). Pada tahapan ini memiliki sub-tahapan yang dapat dilihat pada gambar 2.



Gambar 2. Tahapan *Preprocessing* Data

1. Data Cleaning

Pada tahapan ini akan dilakukan penghapusan beberapa atribut data yang tidak diperlukan dalam penelitian karena data tersebut dapat mengganggu hasil prediksi.

2. Dimensionality Reduction

Dimensionality Reduction bertujuan untuk menghindari pola yang *noisy* dan menghindari “*Memory Error*” atau kesalahan memori akibat proses komputasi terlalu banyak memakan memori karena kumpulan data yang sangat besar. Pada tahapan ini dilakukan penyaringan film populer berdasarkan rating (peringkat nilai) yang diperoleh sebanyak lebih dari sama dengan 50 rating (peringkat nilai).

```
movieId
1          Toy Story (1995)
2          Jumanji (1995)
3          Grumpier Old Men (1995)
6          Heat (1995)
7          Sabrina (1995)
...
109374     Grand Budapest Hotel, The (2014)
109487     Interstellar (2014)
112852     Guardians of the Galaxy (2014)
116797     The Imitation Game (2014)
122904     Deadpool (2016)
Name: title, Length: 450, dtype: object
```

Gambar 3. *Popular Movie Filtering*

Gambar 3 menunjukkan bahwa hasil penyaringan film populer memangkas total film yang sebelumnya berjumlah 942 film menjadi 450 film dan mengurangi dimensi data yang sebelumnya 100836 x 3 menjadi 41360 x 3. kemudian *dataset* akan direduksi kembali dengan cara diambil sebanyak 60% data sampel.

	userId	movieId	rating	
	81055	513	260	5.0
	55414	368	161	4.0
	71725	462	2959	3.5
	78008	484	1258	3.5
	11770	72	296	4.5

	18801	121	288	4.0
	9250	63	2683	3.0
	10215	66	3911	4.0
	54407	357	5989	4.5
	98737	608	364	3.5

24816 rows x 3 columns

Gambar 3. Hasil *Random Sample 60%*

Pada gambar 4 menjelaskan bahwa reduksi data dengan mengambil sampel data sebanyak 60% dapat mereduksi dimensi *dataset* yang sebelumnya 41360 x 3 menjadi 24816 x 3. Hal ini cukup untuk menghindari “*Memory Error*” pada saat proses prediksi.

2.3 Transformasi Data

Transformasi Data yaitu melakukan perubahan bentuk data menjadi *sparse matrix* dengan bentuk $m \times n$ array. Di mana m adalah id film dan n adalah id *users* dan *value* dari matriks adalah rating (peringkat nilai). Untuk mengubah bentuk dataframe *ratings*, akan digunakan sebuah fungsi *pivot* untuk mengubah bentuk data kedalam format yang lebar di mana film sebagai baris dan *users* sebagai kolom. Hal ini perlu dilakukan agar data dapat dikonsumsi oleh model untuk proses prediksi.

userId	1	2	3	4	5	6	7	8	9	10	...	601	602	603	604	605	606	607	608	609	610	
movieId																						
1	4.0	0.0	0.0	0.0	4.0	0.0	4.5	0.0	0.0	0.0	...	4.0	0.0	4.0	3.0	4.0	2.5	4.0	2.5	3.0	5.0	
2	0.0	0.0	0.0	0.0	0.0	4.0	0.0	4.0	0.0	0.0	...	0.0	4.0	0.0	5.0	3.5	0.0	0.0	2.0	0.0	0.0	
3	4.0	0.0	0.0	0.0	0.0	5.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	
6	4.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	...	0.0	3.0	4.0	3.0	0.0	0.0	0.0	0.0	0.0	5.0	
7	0.0	0.0	0.0	0.0	0.0	4.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	2.5	0.0	0.0	0.0	0.0	
...
109374	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5	
109487	0.0	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.5	...	5.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.5	
112852	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	4.5	
116797	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
122904	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	

450 rows × 606 columns

Gambar 4. Bentuk Matriks Rating (peringkat nilai)

2.4 Penerapan Model Matrix Factorization

Di dalam *Matrix Factorization* prediksi rating (peringkat nilai) menggunakan algoritma SVD (*Singular Value Decomposition*). Untuk menerapkan SVD dalam domain *Collaborative Filtering* memerlukan sebuah pemfaktoran matriks peringkat item-pengguna (Bokde, Girase, & Mukhopadhyay, 2015).

Untuk prediksi r_{ui} ditetapkan sebagai:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T p_u \quad (1)$$

Jika pengguna u tidak diketahui, maka bias b_u dan faktor p_u diasumsikan nol, hal yang sama berlaku untuk item i dengan b_i dan q_i .

Untuk estimasi semua yang tidak diketahui, sistem meminimalkan kesalahan kuadrat yang diatur oleh berikut ini:

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda(b_i^2 + b_u^2 + \|q_i\|^2 + \|p_u\|^2) \quad (2)$$

Dan untuk mempelajari vektor faktor (p_u dan q_i), sistem menimalisir kesalahan kuadrat yang diatur pada set rating (peringkat nilai) yang diketahui sebagai:

$$\min_{q^*, p^*} \sum_{(u,i) \in K} (r_{ui} - q_i^T p_u)^2 + \lambda(\|q_i\|^2 + \|p_u\|^2) \quad (3)$$

Disini, K adalah himpunan pasangan (u,i) , dan p diketahui sebagai himpunan *training set*. Sistem mempelajari model dengan menyesuaikan rating (peringkat nilai) yang diamati sebelumnya.

Tujuannya adalah untuk menggeneralisasi rating (peringkat nilai) sebelumnya dengan cara memprediksi rating (peringkat nilai) yang tidak diketahui untuk kedepannya. Konstanta λ mengontrol tingkat regulasi dan biasanya ditentukan oleh *cross-validation*.

2.5 Penerapan Model K-Nearest Neighbor

Dalam prediksi oleh model *K-Nearest Neighbor* prediksi rating (peringkat nilai) yang diberikan pengguna kepada item dirumuskan sebagai berikut:

$$\hat{r}_{ui} = \frac{\sum_{j \in N_u^k(i)} sim(i,j) \cdot r_{uj}}{\sum_{j \in N_u^k(i)} sim(i,j)} \quad (4)$$

Dengan \hat{r}_{ui} adalah estimasi *rating* (peringkat nilai) pengguna u terhadap item i , $\sum_{j \in N_u^k(i)}$ adalah k tertangga terdekat dari item i yang telah dirating oleh pengguna u , $sim(i,j)$ adalah nilai kemiripan antara item i dan item j dan r_{uj} adalah rating sebenarnya dari pengguna u terhadap item j (Cui, 2017).

2.6 Evaluasi

Pada tahap ini akan dilakukan komparasi antara model *matrix factorization* dan *k-nearest neighbor* berdasarkan nilai *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), dan hasil prediksi yang dihasilkan.

3. Hasil dan Pembahasan

Dataset yang telah ditransformasikan kemudian akan diolah dan diprediksi oleh kedua model, proses prediksi akan dilakukan dengan menentukan parameter pada tiap model lalu diuji dengan *5-Fold Cross Validation*, yang kemudian

akan dilakukan Tuning Parameter untuk menghasilkan parameter dengan hasil prediksi terbaik pada tiap model lalu membandingkannya.

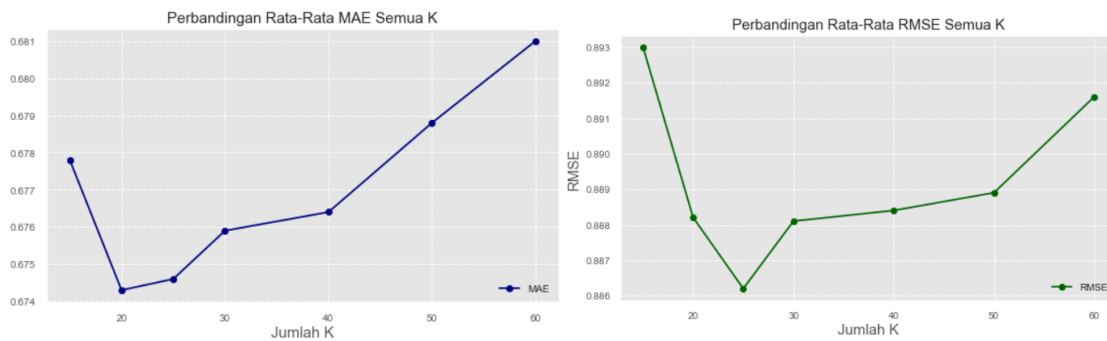
3.1 Pengujian Model K-Nearest Neighbor

Pengujian pada model *K-Nearest Neighbor* tahap pertama yaitu menentukan jumlah K tetangga, lalu tiap K akan diuji menggunakan *5-Fold Cross Validation*. Jumlah K yang akan diuji adalah K=15, K=20, K=25, K=30, K=40, K=50 dan K=60.

Tabel 1. Hasil MAE dan RMSE 5-Fold Cross Validation Model K-Nearest Neighbor

K	MAE	RMSE
15	0,6771	0,8916
20	0,6748	0,8888
25	0,6750	0,8876
30	0,6760	0,8888
40	0,6775	0,8891
50	0,6791	0,8904
60	0,6801	0,8905

Keterangan: *Cetak tebal menunjukkan angka MAE dan RMSE terendah (semakin rendah semakin baik)



Gambar 5. Visualisasi Hasil MAE dan RMSE model KNN

1. Tuning Parameter

Tuning parameter digunakan untuk mencari tahu kombinasi parameter mana yang menghasilkan akurasi terbaik.

```
knn_gs.best_params
{'rmse': {'k': 25, 'sim_options': {'name': 'cosine', 'user_based': False}},
 'mae': {'k': 25, 'sim_options': {'name': 'cosine', 'user_based': False}}}

knn_gs.best_score
{'rmse': 0.8863039603138676, 'mae': 0.6742019354689818}
```

Gambar 6. Hasil Tuning Parameter KNN

Berdasarkan gambar 6 diatas dapat dilihat bahwa hasil tuning parameter menunjukkan parameter terbaik untuk model *K-Nearest Neighbor* adalah dengan K=25, parameter tersebut menghasilkan nilai MAE sebesar 0,6742 dan RMSE sebesar 0,8863.

2. Hasil Prediksi

Berdasarkan Gambar 4 nilai rating yang kosong akan diisi dengan prediksi estimasi rating oleh model *KNN*. Dan untuk memeriksa ketepatan prediksi akan mencoba melihat hasil prediksi terhadap nilai rating yang sudah diberikan (tidak kosong). Tahap pertama dalam penelitian hasil prediksi ini akan melakukan percobaan prediksi

estimasi rating yang kosong pada 10 kolom matriks rating untuk *movieId=1* dan *movieId=16*.

Tabel 2. Hasil Prediksi Rating Kosong oleh KNN

movieId	userId	Prediksi Rating
1	2	3,9000
	3	0,7498
	4	3,6806
	6	4,1599
1	8	3,7607
	9	4,3125
	10	3,2797
16	1	4,5600
	2	3,9020
	3	0,7519
	4	3,2402
	5	3,6000
	7	3,9185
	8	3,8000
	9	4,3091

Keterangan: *Cetak tebal menunjukkan merupakan hasil prediksi terbesar

Berdasarkan hasil estimasi terbesar, *movieId=1* kemungkinan akan direkomendasikan ke *userId=9* dan *movieId=16* kemungkinan akan direkomendasikan ke *userId=1*.

Tabel 3. Hasil Prediksi Nilai Rating Asli oleh KNN

movieId	userId	Rating Asli	Rating Prediksi
1	1	4	4,6790
	5	4	3,7602
	7	4,5	3,8412
16	6	4	3,8405

Berdasarkan Tabel 3 hasil prediksi estimasi rating oleh model *KNN* sedikit agak mendekati dari nilai rating aslinya.

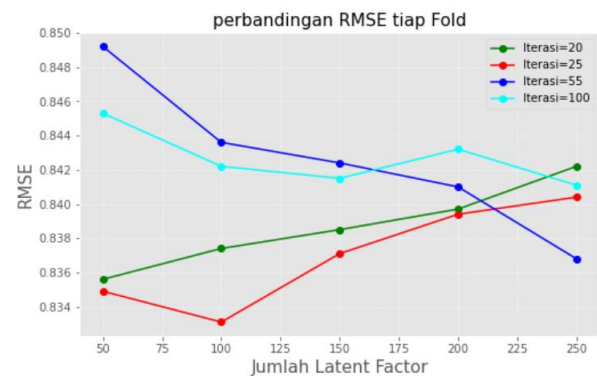
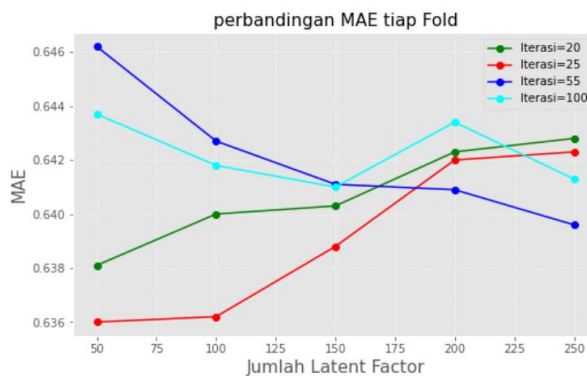
3.2 Pengujian Model Matrix Factorization

Pada pengujian *Matrix Factorization* ini akan dilakukan pengujian dengan jumlah *K latent factor* 50, 100, 150, 200 dan 250, yang akan di uji dengan iterasi 20, 25, 55, 100 dan menggunakan *learning rate* 0,005. Lalu akan dilakukan *5-Fold Cross Validation* untuk pengujianya.

Tabel 4. Hasil MAE dan RMSE *5-Fold Cross Validation Model Matrix Factorization*

Iterasi	Jumlah K Latent Factor	MAE	RMSE
20	50	0,6381	0,8356
	100	0,6400	0,8374
	150	0,6403	0,8385
	200	0,6423	0,8387
	250	0,6428	0,8422
25	50	0,6360	0,8349
	100	0,6362	0,8331
	150	0,6388	0,8371
	200	0,6420	0,8394
	250	0,6432	0,8404
55	50	0,6462	0,8492
	100	0,6418	0,8436
	150	0,6421	0,8424
	200	0,6434	0,8410
	250	0,6413	0,8368
100	50	0,6437	0,8453
	100	0,6418	0,8422
	150	0,6421	0,8415
	200	0,6434	0,8432
	250	0,6413	0,8411

Keterangan: *Cetak tebal menunjukkan angka MAE dan RMSE terendah (semakin rendah semakin baik)



Gambar 7. Visualisasi hasil MAE dan RMSE model MF

1. Tuning Parameter

```
mf_gs.best_params
{'rmse': {'n_factors': 250, 'n_epochs': 100, 'lr_all': 0.005},
 'mae': {'n_factors': 250, 'n_epochs': 100, 'lr_all': 0.005}}

mf_gs.best_score
{'rmse': 0.8305096417954854, 'mae': 0.6317640865453884}
```

Gambar 8. Contoh Keterangan Gambar

Berdasarkan gambar 8 diatas dapat dilihat bahwa hasil tuning parameter menunjukkan parameter terbaik untuk model *Matrix Factorization* adalah dengan jumlah latent factor =

250 dan iterasi = 100 yang menghasilkan nilai MAE sebesar 0,6317 dan RMSE sebesar 0,8305.

2. Hasil Prediksi

Tabel 5. Hasil Prediksi Rating Kosong oleh MF

movieId	userId	Prediksi Rating
1	2	3,5549
	3	2,1769
	4	3,7144
	6	4,2169
	8	4,4557
	9	4,2981
	10	3,4448
16	1	4,6274

movieId	userId	Prediksi Rating
	2	4,0018
	3	2,8481
	4	3,0870
	5	3,8254
	7	3,4486
	8	3,5941
	9	4,2667

Keterangan: *Cetak tebal merupakan hasil prediksi terbesar.

berdasarkan hasil estimasi prediksi rating terbesar, *movieId=1* kemungkinan akan direkomendasikan kepada *userId=8* dan *movieId=16* kemungkinan akan direkomendasikan kepada *userId=1*.

Tabel 6. Hasil Prediksi Nilai Rating Asli oleh MF

movieId	userId	Rating Asli	Rating Prediksi
1	1	4	4,1655
	5	4	3,9348
	7	4,5	4,3690
16	6	4	3,8009

Berdasarkan Tabel 6 hasil prediksi estimasi rating oleh model *Matrix Factorization* hampir mendekati dari nilai rating aslinya.

3.3 Analisis Nilai MAE dan RMSE Kedua Model

Dari beberapa pengujian yang telah dilakukan diatas, diambil hasil pengujian terbaik untuk setiap *tuning parameter*. Dari hasil tersebut kemudian akan dibandingkan nilai MAE dan RMSE yang diperoleh model *Matrix Factorization* dan model *K-Nearest Neighbor*. Berikut merupakan data pengujian rekomendasi dengan model *Matrix Factorization* dan *K-Nearest Neighbor* berdasarkan MAE dan RMSE.

Tabel 7. Perbandingan MAE dan RMSE Model KNN dan MF

Model	Parameter Terbaik	MAE	EMSE
K-Nearest Neighbor	K=25	0,6742	0,8863
Matrix Factorization	Latent Factor = 250 Iterasi = 100 Learning Rate = 0,005	0,6317	0,8305

Nilai MAE dan RMSE dari model *Matrix Factorization* lebih baik, hal ini disebabkan karena pada model *Matrix Factorization* terdapat

algoritma *Singular Value Decomposition* (SVD) di mana algoritma ini akan mereduksi matriks rating R' menjadi beberapa rating kecil dan kemudian dikaitkan hasil perkalian matriks U ($m \times \text{latent}$) dan V ($n \times \text{latent}$) untuk menghasilkan estimasi rating dan memiliki prosedur *Stochastic Gradient Descent* sebagai pengoptimalan *value* hasil perkalian dan melakukan iterasi pada setiap rating estimasi terhadap rating asli untuk mendapatkan nilai *error* minimum.

Dari hasil pengujian yang berada pada Tabel 7 dapat diambil kesimpulan sebagai berikut: Semakin banyak jumlah *latent factor* membuat nilai MAE dan RMSE meningkat, hal ini disebabkan semakin banyak jumlah *latent factor* semakin besar juga dimensi matriks U dan V di mana perkalian antara dua matriks tersebut akan menghasilkan nilai yang besar, sehingga nilainya jauh dari matriks R (matriks yang diharapkan) dan nilai rating prediksi akan semakin jauh dari rating yang diharapkan. Namun untuk menghindari *Overfitting* (prediksi model hanya bagus pada data tertentu, namun tidak bagus di data yang lain) fungsi *GridSearchCV()* memilih *latent factor* sebanyak 250.

Sedangkan pada model *K-Nearest Neighbor* matriks rating (R') akan dihitung dengan mencari nilai kemiripan antar film oleh algoritma *Cosine Similarity* dan mengelompokkan estimasi rating terdekat sesuai dengan K yang ditentukan dan mengembalikan nilai estimasi rating yang terbaik, nilai MAE dan RMSE bisa kemungkinan dipengaruhi oleh jumlah data *training*-nya, karena semakin besar jumlah data *training* maka nilai kemiripan antar item semakin besar, sehingga hasil prediksi akan mendekati nilai sebenarnya, dan membuat nilai MAE dan RMSE menjadi mengecil.

Jadi banyaknya prosedur yang dilakukan yang dimiliki pada setiap model dapat mempengaruhi hasil RMSE dan MAE. Dalam kasus ini untuk efektivitas waktu model *K-Nearest Neighbor* yang menjadi terbaik karena prosedur dalam memprediksi estimasi rating tidak sebanyak model *Matrix Factorization* namun untuk nilai MAE dan RMSE model *Matrix Factorization* menjadi yang terbaik.

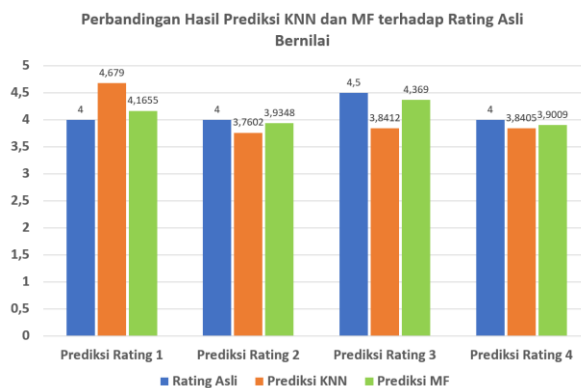
3.4 Analisis Hasil Prediksi Rating

Tabel 8. Perbandingan Prediksi Rating terhadap Rating Asli Kosong oleh Model KNN dan MF

movieId	userId	Rating Asli	Hasil Rating Prediksi	
			KNN	MF
1	2	0	3,9000	3,5549
	3	0	0,7598	2,1769
	4	0	3,6806	3,7144
	6	0	4,1599	4,2169
	8	0	3,7607	4,4557
	9	0	4,3125	4,2981
	10	0	3,2797	3,4448
16	1	0	4,5600	4,6274
	2	0	3,9020	4,0018
	3	0	0,7519	2,8481
	4	0	3,2402	3,0870
	5	0	3,6000	3,8254
	8	0	3,9185	3,4486
	9	0	3,8000	3,5941

Tabel 9. Perbandingan Prediksi Rating terhadap Rating Asli Bernilai oleh Model KNN dan MF

movieId	userId	Rating Asli	Hasil Rating Prediksi	
			KNN	MF
1	1	4	4,6790	4,1655
	5	4	3,7602	3,9348
	7	4,5	3,8412	4,3690
16	7	4	3,8405	3,9009



Gambar 9. Visualisasi Prediksi Model terhadap Rating Asli Bernilai

Kedua model berhasil memprediksi estimasi rating terhadap rating asli kosong (nol) dan rating asli yang bernilai. Pada prediksi rating asli kosong (nol), hasil terbesar digunakan untuk mengetahui kemungkinan rekomendasi yang dihasilkan pada setiap model, sedangkan pada prediksi rating asli yang bernilai, digunakan untuk mengetahui seberapa akurat hasil prediksi yang dihasilkan oleh kedua model.

Berdasarkan hasil prediksi terbesar oleh kedua model pada Tabel 8, dapat diketahui bahwa kedua model merekomendasi film dengan label *MovieId 1* kepada *user* yang berbeda dan merekomendasi film dengan label *MovieId 16* kepada *user* yang sama. Lalu berdasarkan tabel 9 dan Gambar 9, dapat diketahui bahwa prediksi rating oleh model *Matrix Factorization* lebih mendekati nilai rating aslinya daripada model *K-Nearest Neighbor*. Dalam hal ini hasil prediksi rating oleh model *Matrix Factorization* bisa dikatakan lebih akurat dibandingkan dengan model *K-Nearest Neighbor*. Hal ini juga dipengaruhi oleh nilai MAE dan RMSE pada model *Matrix Factorization* yang lebih rendah dibandingkan dengan model *K-Nearest Neighbor*.

4. Kesimpulan

Kesimpulan yang didapat setelah dilakukan pengujian dan analisis prediksi rating oleh kedua model didapat bahwa model dengan performa terbaik untuk mesin rekomendasi berbasis prediksi rating adalah *Matrix Factorization* dengan nilai MAE sebesar 0,6417 dan RMSE sebesar 0,8305 hasil tersebut lebih kecil dibandingkan dengan model *K-Nearest Neighbor* menghasilkan nilai MAE sebesar 0,6742 dan RMSE sebesar 0,8863, di mana prediksi dengan nilai MAE dan RMSE yang lebih kecil merupakan prediksi yang lebih akurat. Hal ini dibuktikan dengan hasil prediksi rating model *Matrix Factorization* yang mendekati nilai rating aslinya.

5. Saran

Disarankan dalam pengembangan penelitian selanjutnya yaitu dengan melakukan perbandingan model *Matrix Factorization* dan *K-Nearest Neighbor* dengan parameter uji yang berbeda, dibuatkan sebuah GUI agar bisa dilakukan penilaian hasil rekomendasi oleh *real user* dan menguji model dengan dataset yang berbeda.

Referensi

- Abraham, S., & Rahayu, Y. D. (2017). Sistem Rekomendasi Artikel Berita Menggunakan Metode K-Nearest Neighbor Berbasis Website. In *Prosiding SENSEI 2017* (pp. 179–187). Jember: Universitas Muhammadiyah Jember.
- Adellya, A., Devi, P., & Tonara, D. B. (2015). Rancang Bangun Recommender System dengan Menggunakan Metode Collaborative Filtering untuk Studi Kasus Tempat Kuliner di Surabaya. *JUISI Jurnal Informatika Dan Sistem Informasi*, 01(02), 102–112. <https://doi.org/10.1212/WNL.0b013e3181dbb664>

- Alasadi, S. A., & Bhaya, W. S. (2017). Review of Data Preprocessing Techniques. *Journal of Engineering and Applied Sciences*, 12(16), 4102–4107.
- Amatriain, X., Jaimes, A., Oliver, N., & Pujol, J. M. (2011). *Recommender Systems Handbook Chapter 2 Data Mining Methods for Recommender Systems*. Boston: Springer US. <https://doi.org/10.1007/978-0-387-85820-3>
- Bokde, D. K., Girase, S., & Mukhopadhyay, D. (2015). Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey ScienceDirect Matrix Factorization Model in Collaborative Filtering Algorithms: A Survey. *Procedia Computer Science*, 49, 136–146. <https://doi.org/10.1016/j.procs.2015.04.237>
- Cui, B. (2017). Design and Implementation of Movie Recommendation System Based on Knn Collaborative Filtering Algorithm. In *The 4th Annual International Conference on Information Technology and Applications (ITA 2017)* (Vol. 04008, pp. 8–12). Guangzhou: ITM Web Of Conferences. <https://doi.org/10.1051/itmconf/20171204008>
- Krishnamurty, S., Nurjanah, D., & Rismala, R. (2017). Sistem Rekomendasi Pada Buku Dengan Menggunakan Tags and Latent Factors. In *e-Proceeding of Engineering* (Vol. 4, pp. 4695–4701). Bandung: Universitas Telkom.
- Li, H., Liu, Y., Qian, Y., Mamoulis, N., Tu, W., & Cheung, D. W. (2019). HHMF: hidden hierarchical matrix factorization for recommender systems. *Data Mining and Knowledge Discovery*, 33(6), 1548–1582. <https://doi.org/10.1007/s10618-019-00632-4>
- Prasetya, C. S. D. (2017). Sistem Rekomendasi Pada E-Commerce Menggunakan K-Nearest Neighbor. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIIK)*, 4(3), 194–200.
- Sorde, R., & Deshmukh, S. (2015). Comparative Study on Approaches of Recommendation Systems. *International Journal of Computer Applications*, 118(2), 753–764. https://doi.org/10.1007/978-981-15-0947-6_72
- Syahrani, M. A., Setiawan, E. B., & Suryani, S. (2015). Analisis Perbandingan Sistem Rekomendasi dengan Faktorisasi Matriks dan. In *SNIKTI (Seminar Nasional Ilmu Komputasi dan Teknik Informatika)* (pp. 50–62). Bandung: Universitas Telkom.