

# Sistem Rekomendasi Produk Elektronik Berbasis Collaborative Filtering Menggunakan Matrix Factorization

Ustami Rajib

Fakultas Informatika, Universitas Telkom, Bandung  
[ustamirajib@students.telkomuniversity.ac.id](mailto:ustamirajib@students.telkomuniversity.ac.id)

Agung Toto Wibowo

Fakultas Informatika, Universitas Telkom, Bandung  
[agungtoto@telkomuniversity.ac.id](mailto:agungtoto@telkomuniversity.ac.id)

**Abstrak** — Sistem rekomendasi adalah suatu program yang melakukan prediksi suatu item, dalam pembuatan sistem rekomendasi terdapat Beberapa metode yang dapat digunakan diantaranya Collaborative Filtering karena dianggap mampu memberikan saran item yang lebih akurat. pendekatan Collaborative Filtering karena dianggap mampu memberikan saran item yang lebih akurat. Pada penelitian ini akan dibuat sistem rekomendasi menggunakan 3 Algoritma Turunan MF yaitu Singular Value Decomposition (SVD), SVD++, Non-Negative Matrix Factorization NMF terhadap dataset Amazon Review dengan Studi Kasus Elektronik, ini perlu diaplikasikan dalam penelitian sistem rekomendasi, karena data Elektronik ini mempunyai jumlah data yang sangat besar. Dalam penelitian ini akan dilakukan uji coba terhadap beberapa parameter yang meliputi n-epochs, n-factor dalam mekanisme 5-fold cross-validation. Untuk menangani data yang terlalu besar, penulis melakukan random sampling sebesar 25% dari total dataset untuk mengurangi beban komputasi. Dari hasil uji coba didapatkan performansi rata-rata terbaik MAE = 1.0384 dan RMSE = 1.3139 yaitu pada Algoritma SVD.

**Kata kunci**— Produk Elektronik, Sistem Rekomendasi, Collaborative Filtering, Matrix Factorization, Cross Validation

## I. PENDAHULUAN

Sistem rekomendasi adalah suatu program yang melakukan prediksi suatu item, seperti rekomendasi film, musik, buku, berita dan lain sebagainya yang membuat pengguna tertarik[1]. Collaborative filtering merupakan metode dalam memberikan rekomendasi yang didasarkan pada kesukaan banyak user terhadap suatu produk untuk membangun sistem rekomendasi dengan konsep di mana opini dari pengguna lain Digunakan untuk memprediksi item yang diminati oleh pengguna[2]. Collaborative filtering (CF) dibagi menjadi dua kelas yaitu memory-based dan model-based, dalam metode CF berbasis model terdapat berbagai algoritma, Seperti Matrix factorization (MF) [3]. Penelitian ini akan membuat sistem rekomendasi menggunakan tiga Algoritma Turunan MF yaitu Singular Value Decomposition (SVD), SVD++ dan Non-Negative Matrix Factorization (NMF) terhadap dataset Amazon Review dengan Studi Kasus Elektronik, ini perlu diaplikasikan dalam sistem rekomendasi, karena data Elektronik ini mempunyai jumlah data Produk Elektronik yang sangat besar, dan dalam dataset ini pun terdapat fitur-fitur yang dibutuhkan dalam membuat sistem rekomendasi ini. Penulis mengusulkan tiga algoritma turunan MF untuk membuat model sistem rekomendasi produk elektronik berdasarkan pendekatan CF. Untuk membangun model dengan menggunakan dataset Amazon Review dengan studi kasus Electronics, Penulis memilih dataset ini karena fitur-fitur yang terdapat pada data sesuai dengan model yang akan dibangun. Selain itu, algoritma turunan MF yang penulis gunakan adalah Singular

Value Decomposition (SVD), SVD++ dan NMF. Dan membandingkan hasil kinerjanya dengan algoritma BaseLineOnly Berdasarkan nilai yang diperoleh dari hasil evaluasi error. Performa metode CF kemudian dievaluasi menggunakan Mean Absolute Error (MAE) dan Root Mean Squared Error (RMSE).

## II. KAJIAN TEORI

### A. Sistem Rekomendasi

Sistem rekomendasi merupakan sistem yang memberikan saran untuk item atau produk yang mungkin menarik bagi pengguna tertentu. Tujuan utama sistem rekomendasi dimaksudkan untuk individu yang belum memiliki pengalaman sebelumnya atau kemampuan untuk memilih item, barang[3].

### B. Collaborative Filtering

Collaborative Filtering merupakan cara untuk membuat rekomendasi dengan melihat nilai pengguna lain untuk item tertentu [4]. Pendekatan Collaborative filtering dikategorikan ke dalam dua kelas yaitu Memory-based dan Model-based [5]. Model-based yaitu metode yang dapat meminimalkan masalah cold-start, ketersebaran data dan skalabilitas [6]. Memory-based juga metode yang berfokus pada relasi antara pengguna user-based atau item-based dengan cara memanfaatkan interaksi yang sudah pernah dilakukan antara pengguna dan item [7].

### C. Matrix Factorization

Matrix Factorization merupakan jenis pendekatan dimana matriks yang besar dipecah menjadi matriks yang lebih kecil. Disebutkan juga bahwa metode ini meningkatkan kinerja dari sistem rekomendasi dan kelebihan dari metode ini dapat menemukan faktor laten lalu mengurangi dimensi[8].

### D. Singular Value Decomposition (SVD)

SVD adalah sebuah teknik faktorisasi SVD bekerja dengan menerima input berupa sebuah matriks (dapat berupa implicit feedback) [3]. Secara umum rumus SVD dituliskan sebagai berikut :

$$X = P \cdot \Sigma \cdot Q^T \quad (1)$$

### E. SVD++

(SVD++) merupakan pengembangan dari SVD dengan mempertimbangkan rating implisit [9]. Rating implisit ini dapat berupa Riwayat penelusuran atau item yang dinilai oleh pengguna [10]. Persamaan SVD++ dapat dituliskan pada persamaan sebagai berikut :

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T \left( p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right) \quad (2)$$

### F. Non-Negative Matrix Factorization

Non-Negative Matrix Factorization (NMF) adalah metode memfaktorisasi matriks dengan dengan cara dekomposisi sebuah matriks menjadi 2 buah matriks yang lebih kecil metode ini memfaktorisasi matriks yang mempertahankan asumsi bahwa semua nilai dalam matriks faktor adalah non-negatif. Pendekatan ini berguna terutama ketika data bersifat non-negatif, seperti dalam kasus peringkat pengguna terhadap *item* [11] Rumus NMF dapat ditulis sebagai berikut :

$$\hat{r}_{ui} = q_i^T p_u \tag{3}$$

G. Benchmark Method

Benchmark Method adalah sebuah metode perbandingan digunakan dalam penelitian ini adalah *BaselineOnly* dalam *BaselineOnly*[10]. Untuk menghitung prediksi rating dapat ditulis pada persamaan berikut :

$$\hat{r}_{ui} = b_{ui} = \mu + b_u + b_i \tag{4}$$

H. Mean Absolute Error

Mean Absolute Error adalah metode dalam proses evaluasi akurasi suatu sistem dengan Teknik membandingkan nilai hasil prediksi dengan nilai aslinya [12]. Semakin besar nilai MAE maka semakin tidak akurat prediksi rating dari sistem rekomendasi, sebaliknya semakin rendah nilai MAE maka semakin menjukan keakuratannya dalam prediksi rating dari sistem rekomendasi [3]. Rumus MAE dapat ditulis dengan menggunakan persamaan sebagai berikut :

$$MAE = \frac{\sum_{i=1}^n |P_i - q_i|}{N} \tag{5}$$

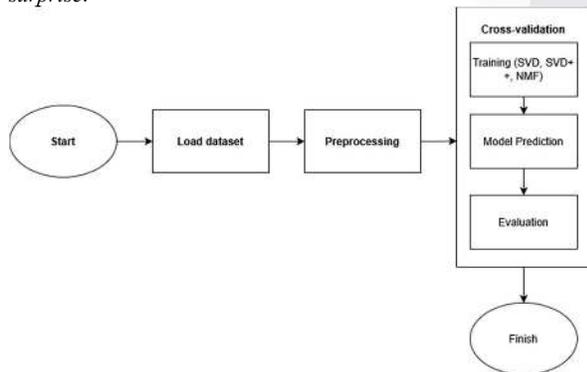
I. Root Mean Squared Error

Root Mean Squared Error menghitung seberapa besar rata-rata kuadrat dari perbedaan nilai prediksi dibandingkan dengan nilai yang aktual [3], rumus RMSE dapat ditulis pada persamaan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (p_{ui} - r_{ui})^2} \tag{6}$$

III. METODE

Sistem rekomendasi Produk Elektronik yang dibuat, memerlukan beberapa tahapan yang harus dilakukan dengan baik. Dapat dilihat Pada gambar 3.1, dibawah yang merupakan tahapan sistem rekomendasi Produk Elektronik dengan metode CF yang dimulai dengan memuat dataset yang telah diunduh dari *website Amazon Reviews* (<https://amazon-reviews-2023.github.io>) dengan studi kasus *Electronics*, dalam penelitian ini penulis melakukan beberapa prosedur yang terdiri dari beberapa tahap yaitu memulai dari *load dataset*, *preprocessing data* dan *cross-validation*. Ketiga algoritma yang diusulkan diimplementasikan menggunakan *library surprise*.



GAMBAR 3.1  
Flowchart Penelitian

A. Dataset

Dataset yang digunakan untuk studi ini adalah *Amazon Review* (dengan studi kasus *Amazon Product Review Electronic*) yang dapat diunduh pada link berikut (<http://jmcauley.ucsd.edu/data/amazon/>). Dalam dataset ini memiliki *attribute* dan *variable* yang diperlukan untuk penelitian. Berikut terdiri dari 4 Kolom yaitu (“*user\_id*”, “*produk\_id*”, “*rating*”, “*time*”) dan total memiliki 7.824.481 baris. Dengan baris *user\_id* memiliki 4.201.696 unik baris, lalu *product\_id* mempunyai 476.001 baris, Dapat dilihat pada Tabel 3.1.

TABEL 3.1

Rangkuman Data yang digunakan dalam percobaan

Nama Atribut	Jumlah Data
<i>user_id</i>	4.201.696
<i>product_id</i>	476.001
<i>rating</i>	7.824.482
<i>time</i>	5489

Pada kolom atribut yang terdapat pada dataset meliputi *user\_id*, *product\_id* dan *rating* memiliki jumlah baris yang unik.

B. Preprocessing Data

1. Data Cleansing

Pada tahap ini penulis melakukan proses pembersihan data seperti data yang duplikat dan nilai yang hilang, dua hal tersebut akan penulis *drop* dikarenakan untuk memastikan bahwa data yang digunakan dalam analisis menjadi akurat dan relevan.

2. Data Transformation

Dapat dilihat rincian pada *table 1* atribut yang dibutuhkan dalam penelitian ini adalah “*user\_id*”, “*product\_id*”, dan “*rating*”. Atribut “*time*” tidak diperlukan untuk permodelan berdasarkan algoritma *Matrix Factorization* (MF) *Penyaringan Kolaboratif*

3. Random Sampling

Dalam proses pembuatan model, penulis mengurangi data tersebut dengan metode *random sampling* yaitu 25% dari seluruh populasi dengan memanfaatkan *library pandas* yaitu *sample()*. Pada tahap ini dilakukan *random sampling* terhadap dataset (“*electronic\_Ratings.csv*”) karena dataset ini cukup besar yaitu total 7.824.481 baris, maka perlu dilakukannya pengambilan sample secara random dengan menjamin bahwa setiap pengguna mempunyai peluang yang sama. Tujuan melakukan *random sampling* agar mempermudah dalam pengolahan data, dan juga mengurangi beban komputasi.

C. Skenario Uji Coba

Pada tahap uji coba dalam mekanisme *k-fold cross-validation*. Mekanisme ini. Selama pengujian penulis melakukan inialisasi *k-fold* dengan *k=5*, *epoch=20* dengan variasi *n-factor* yang diuji adalah (10, 15, 20) dari total 1.956.120 data. ketiga model algoritma tersebut dilatih dalam satu proses pengujian sehingga kita dapat melihat perbedaan performansi dari setiap algoritma dalam memprediksi rating. Ada dua jenis evaluasi yang penulis gunakan dalam penelitian ini untuk mengukur kinerja algoritma yang sudah penulis sebutkan diatas yaitu MAE & RMSE Kedua evaluasi matriks tersebut dipilih karena sesuai dengan model yang penulis bangun.

D. Cross-Validation

Pada penelitian ini bertujuan untuk menguji dataset *Amazon Product Electronics* dengan ketiga turunan algoritma *matrix factorization* yaitu, SVD, SVD++ dan NMF. Kemudian membandingkan dengan algoritma *BaselineOnly*. Dataset ini terdiri atas kolom (“*user\_id*”, “*product\_id*”, “*rating*” dan “*time*”) atribut yang penulis butuhkan hanya 3 yaitu “*user\_id*”, “*product\_id*” dan “*rating*”.

dalam tahap *cross-validation*, dilakukan dengan metode *5-fold cross-validation* pada setiap eksperimen menggunakan 4 bagian sebagai *training set*, sementara 1 bagian sisanya digunakan sebagai *test set*, proses ini diulang sebanyak 5 kali, dengan setiap *fold* mendapatkan bagian-nya menjadi *test set* sekali, untuk setiap eksperimen, model diuji dengan parameter *n-epoch = 20* dan

variasi *n-factor* (10, 15, 20). Setelah semua iterasi selesai, nilai evaluasi dihitung sebagai rata-rata dari hasil validasi pada ke-5 *fold*. Penulis menggunakan metrik evaluasi yaitu MAE dan *RMSE* yang diimplementasikan dengan *library surprise*. Semakin kecil nilai yang dihasilkan MAE dan *RMSE*, semakin baik model dalam memberikan prediksi rating.

IV. HASIL DAN PEMBAHASAN

Seperti yang disebutkan diatas, Penulis menjalankan pengujian menggunakan *5-fold cross-validation* dan *n-epoch*=20 dari total 1.956.120 data. Dalam analisa ini yang penulis hitung adalah MAE dan *RMSE*, yang diimplementasikan dengan *library surprise*, Tabel 4.1 dan Tabel 4.2 menggambarkan rangkuman kinerja masing-masing dari ketiga algoritma.

A. MAE

TABEL 4.1

Hasil Kinerja Algoritma dengan jumlah epoch = 20, k-fold = 5 dan berbagai n-latent factors

Algoritma	N-Factors	1	2	Fold-k 3	4	5	Rata-Rata
BaselineOnly	-	1.0434	1.0434	1.0348	1.0431	1.0420	1.0433
	10	1.0391	1.0375	1.0397	1.0379	1.0380	1.0384
SVD	15	1.0390	1.0386	1.0398	1.0389	1.0382	1.0389
	20	1.0408	1.0394	1.0389	1.0386	1.0368	1.0389
	10	1.0396	1.0414	1.0375	1.0391	1.0408	1.0397
SVD++	15	1.0413	1.0429	1.0424	1.0408	1.0435	1.0422
	20	1.0431	1.0425	1.0435	1.0444	1.0446	1.0436
	10	1.2644	1.2637	1.2640	1.2639	1.2676	1.2647
NMF	15	1.1541	1.1522	1.1526	1.1516	1.1508	1.1523
	20	1.0957	1.0926	1.0934	1.0935	1.0918	1.0934

B. RMSE

TABEL 4.2

Hasil Kinerja Algoritma dengan jumlah epoch = 20, k-fold = 5 dan berbagai n-latent factors

Algoritma	N-Factors	1	2	Fold-k 3	4	5	Rata-Rata
BaselineOnly	-	1.3157	1.3175	1.3160	1.3156	1.3145	1.3158
	10	1.3146	1.3136	1.3154	1.3132	1.3128	1.3139
SVD	15	1.3141	1.3139	1.3157	1.3136	1.3138	1.3188
	20	1.3176	1.3145	1.3149	1.3139	1.3115	1.3145
	10	1.3208	1.3228	1.3176	1.3202	1.3224	1.3207
SVD++	15	1.3221	1.3247	1.3228	1.3220	1.3251	1.3234
	20	1.3242	1.3239	1.3245	1.3266	1.3268	1.3252
	10	1.5507	1.5500	1.5503	1.5502	1.5535	1.5509
NMF	15	1.4602	1.4589	1.4597	1.4587	1.4588	1.4593
	20	1.4360	1.4314	1.4329	1.4336	1.4327	1.4333

Dalam tabel 4.1 dan Tabel 4.2 menampilkan hasil prediksi pada setiap algoritma. *n-factor* menunjukan jumlah fitur laten untuk setiap algoritma. Kemudian, *fold-k* menunjukan setiap iterasi yang dilakukan pada setiap algoritma. Setelah itu pada kolom rata-rata menunjukan rata-rata dari semua iterasi. Dari tabel 4.1 dapat dilihat dari beberapa perbedaan *n-factor* dan *fold-k*. Pada Algoritma SVD dan SVD++ memiliki nilai MAE yang mirip, algoritma SVD mampu menghasilkan nilai prediksi terbaik pada *n-factor* = 10 iterasi pada *fold-k* = 5 dengan nilai rata-rata MAE yaitu 1.0384. Dibandingkan dengan SVD++ setiap *n-factor* ditambahkan hasil dari MAE semakin besar. Selanjutnya pada algoritma NMF nilai prediksi terbaik pada *n-factor* = 20 dan iterasi *k-fold* = 5 dengan nilai MAE yaitu 1.0918, Selain itu dapat lihat bahwa SVD & SVD++ yang di uji dalam penelitian ini mampu melampaui kinerja dari model *BaselineOnly*. Dalam *RMSE* yang dirangkum pada tabel 4.2, Setiap algoritma memiliki nilai yang berbeda dari hasil evaluasi sebelumnya. Algoritma SVD menghasilkan prediksi terbaik dengan *n-factor* = 20 dan iterasi *fold-k* = 5 dengan nilai *RMSE* sebesar 1.3115. Rata-rata nilai *RMSE* terbaik untuk semua iterasi pada algoritma SVD adalah 1.3139, Kemudian untuk algoritma SVD++, Nilai prediksi terbaik yaitu pada *n-factor* = 10 dan iterasi *fold-k* = 3 dengan nilai *RMSE* 1.3176. Dan yang terakhir nilai prediksi terbaik untuk algoritma NMF diperoleh pada *n-factor* = 20 dan *k-fold* = 2, dengan nilai *RMSE* 1.4314, dari semua iterasi nilai *RMSE* rata-rata

terbaik untuk algoritma NMF adalah 1.4433. Pada hasil *RMSE* hanya algoritma SVD yang dapat mengungguli kinerja model *BaselineOnly*.

Berdasarkan hasil rangkuman evaluasi *error* untuk setiap algoritma dalam Tabel 4.1 dan tabel 4.2, kita dapat melihat perbedaan perbandingan dalam kinerja algoritma yang memberikan prediksi *rating*. Berdasarkan tabel 4.1 algoritma dengan performa terbaik berdasarkan metrik MAE adalah SVD dengan nilai rata-rata 1.0384, Sementara itu, dalam tabel 4.2 menunjukkan bahwa algoritma dengan nilai *RMSE* terbaik juga adalah SVD dengan rata-rata 1.3139. hal ini terjadi karena kedua metrik evaluasi tersebut memiliki parameter yang berbeda dalam melakukan pengukuran, MAE mengukur nilai kesalahan absolut sedangkan *RMSE* mengukur kesalahan kuadrat. Dari MAE, algoritma SVD mampu menghasilkan nilai *error* rata-rata paling kecil jika dibandingkan algoritma lainnya.

Dapat dilihat pada tabel dibawah bahwa pada algoritma SVD dari *n-factor* 15 ke 20 hampir tidak ada perubahan, ini memperlihatkan bahwa adanya titik terbaik dimana penambahan faktor laten tidak lagi berpengaruh secara signifikan terhadap akurasi prediksi.

TABEL 4.3

Analisis SVD

n-factors	MAE	RMSE
10	1,0384	1,3139
15	1,0389	1,3145
20	1,0389	1,3145

Kemudian, NMF menghasilkan nilai yang cukup besar, terutama pada *n-factor* yang lebih kecil, namun pada *n-factor* = 20, performa NMF meningkat menjadi lebih baik dibandingkan *n-factor* sebelumnya, hal ini terjadi karena NMF bekerja dengan faktor laten non-negatif, sehingga membutuhkan dimensi faktor yang lebih besar untuk menangkap data dengan baik.

TABEL 4.4

Analisis NMF

n-factors	MAE	RMSE
10	1,2647	1,5509
15	1,1523	1,4593
20	1,0934	1,4333

Selain itu, SVD++, mirip dengan SVD, memiliki performa yang lebih rendah dibandingkan SVD karena algoritma ini menggunakan implisit *feedback* dalam perhitungannya, maksud dari implisit *feedback* yaitu interaksi pengguna dengan produk elektronik seperti melihat produk atau membeli tanpa memberikan rating, yang dimana dapat menyebabkan *noise* dalam data, hal ini bisa menjadi pengaruh terhadap nilai *error* yang sedikit lebih besar dibandingkan algoritma SVD.

TABEL 4.5

Analisis SVD++

n-factors	MAE	RMSE
10	1,0397	1,3207
15	1,0422	1,3234
20	1,0436	1,3252

Hasil pengujian dapat dilihat bahwa semakin tinggi *n factor* yang ditambahkan performa algoritma cenderung membaik, tetapi peningkatan *n factor* bukan berarti peningkatan performa secara signifikan. Sebagai contoh, pada algoritma SVD, perbedaan antara *n-factor* = 10 dan *n-factor* = 20 tidak berbeda jauh, hal ini memperlihatkan bahwa ada sebuah titik terbaik dalam pemilihan, *n factor*, yang dimana setelah batas tertentu, penambahan faktor laten tidak lagi memberikan peningkatan performa yang signifikan. Dari hasil pengujian, dapat dilihat bahwa *BaselineOnly* memiliki nilai MAE dan *RMSE* yang lebih besar dibandingkan ketiga algoritma *Matrix Factorization*. Hal tersebut menunjukkan bahwa penggunaan faktor laten dalam *matrix factorization* mampu menangkap pola tersembunyi dalam data *rating*.

## V. KESIMPULAN

Berdasarkan dari hasil evaluasi dan perbandingan nilai kesalahan prediksi yang telah penulis lakukan, penulis menyimpulkan bahwa sistem rekomendasi dari ketiga algoritma yang penulis usulkan mampu melakukan prediksi dengan baik. Dari hasil evaluasi yang menggunakan *5 fold cross-validation*, ketiga algoritma yang penulis usulkan mampu mengungguli performa prediksi algoritma *BaselineOnly*. Dari hasil evaluasi MAE, algoritma prediksi yang menghasilkan performa terbaik dalam memberikan rekomendasi produk elektronik adalah (SVD), dengan nilai rata-rata MAE yaitu 1.0384, sedangkan algoritma yang memberikan hasil prediksi RMSE terbaik adalah SVD juga dengan nilai 1.3139, sebaliknya nilai SVD++ cenderung menghasilkan nilai yang lebih tinggi seiring *n-factor* bertambah, karena algoritma ini menggunakan implisit *feedback* dalam perhitungannya, maksud dari implisit *feedback* yaitu interaksi pengguna dengan produk elektronik seperti melihat produk atau membeli tanpa memberikan rating. Sedangkan NMF dapat menghasilkan MAE terbaik pada *n-factor* = 20 dengan nilai rata-rata MAE yaitu 1.0934, hal ini terjadi karena NMF bekerja dengan faktor laten non-negatif, sehingga membutuhkan dimensi faktor yang lebih besar untuk menangkap pola dengan baik. Dari pengujian algoritma ini penulis menyimpulkan bahwa algoritma SVD dapat memberikan nilai evaluasi prediksi paling rendah, diantara algoritma lain tetapi masih belum dapat mencapai nilai dibawah 1 (satu).

## REFERENSI

- [1] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," *IEEE Access*, vol. 6, pp. 69009–69022, 2018, doi: 10.1109/ACCESS.2018.2880197.
- [2] C. Channarong, C. Paosirikul, S. Maneeroj, and A. Takasu, "HybridBERT4Rec: A Hybrid (Content-Based Filtering and Collaborative Filtering) Recommender System Based on BERT," *IEEE Access*, vol. 10, pp. 56193–56206, 2022, doi: 10.1109/ACCESS.2022.3177610.
- [3] D. Jannach and M. Zanker, *Value and Impact of Recommender Systems*. 2022. doi: 10.1007/978-1-0716-2197-4\_14.
- [4] P. Sharma and L. Yadav, "Movie Recommendation System Using Item Based Collaborative Filtering," *International Journal of Innovative Research in Computer Science & Technology*, vol. 8, no. 4, pp. 8–12, 2020, doi: 10.21276/ijircst.2020.8.4.2.
- [5] X. Guan, C. T. Li, and Y. Guan, "Matrix Factorization with Rating Completion: An Enhanced SVD Model for Collaborative Filtering Recommender Systems," *IEEE Access*, vol. 5, pp. 27668–27678, 2017, doi: 10.1109/ACCESS.2017.2772226.
- [6] I. Saifudin and T. Widiyaningtyas, "Systematic Literature Review on Recommender System: Approach, Problem, Evaluation Techniques, Datasets," *IEEE Access*, vol. 12, no. November 2023, pp. 19827–19847, 2024, doi: 10.1109/ACCESS.2024.3359274.
- [7] D. Seng, G. Chen, and Q. Zhang, "Item-Based Collaborative Memory Networks for Recommendation," *IEEE Access*, vol. 8, pp. 213027–213037, 2020, doi: 10.1109/ACCESS.2020.3039380.
- [8] Y. Zhang, "An Introduction to Matrix factorization and Factorization Machines in Recommendation System, and Beyond," pp. 1–15, 2022, [Online]. Available: <http://arxiv.org/abs/2203.11026>
- [9] Y. Koren, "Factorization meets the neighborhood," pp. 426–434, 2008, doi: 10.1145/1401890.1401944.
- [10] N. Hug, "Surprise: A Python library for recommender systems," *J Open Source Softw*, vol. 5, no. 52, p. 2174, 2020, doi: 10.21105/joss.02174.
- [11] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems," *IEEE Trans Industr Inform*, vol. 10, no. 2, pp. 1273–1284, 2014, doi: 10.1109/TII.2014.2308433.
- [12] H. Hartatik, S. D. Nurhayati, and W. Widayani, "Sistem Rekomendasi Wisata Kuliner di Yogyakarta dengan Metode Item-Based Collaborative Filtering," *Journal Automation Computer Information System*, vol. 1, no. 2, pp. 55–63, 2021, doi: 10.47134/jacis.v1i2.8.