

Perbandingan Cosine Similarity dan Mean Squared Difference dalam Rekomendasi Buku Fiksi berbasis Item

Lucyta Qutsyaning Rosyidah¹, Triyanna Widiyaningtyas^{1,*}

¹ Program Studi Teknik Informatika, Universitas Negeri Malang, Indonesia

* Correspondence: triyannaw.ft@um.ac.id

Copyright: © 2024 by the authors

Received: 12 Oktober 2024 | Revised: 18 Oktober 2024 | Accepted: 7 November 2024 | Published: 19 Desember 2024

Abstrak

Kebutuhan akan rekomendasi semakin krusial di era digital, terutama dengan melimpahnya data buku fiksi dari *platform e-book* dan pustaka digital. Penelitian ini bertujuan untuk mengevaluasi efektivitas *item-based collaborative filtering* menggunakan metrik *cosine similarity* dan *Mean Squared Difference* (MSD) dalam memberikan rekomendasi buku. Metode *knowledge discovery in databases* diterapkan, meliputi pemilihan data, pra-pemrosesan, transformasi, *data mining*, dan evaluasi. Data yang digunakan mencakup 100.000 *rating* pengguna yang diperoleh dari Kaggle dengan judul '*Book Recommendation Dataset*'. Hasil temuan kami menunjukkan bahwa *mean absolute error* untuk MSD adalah 0,152307, sedikit lebih baik dibandingkan *cosine similarity* dengan nilai 0,152406. *Root Mean Squared Error* MSD lebih rendah, yaitu 0,185551, dibandingkan *cosine similarity* sebesar 0,185636. Namun, dari segi efisiensi waktu, *cosine similarity* lebih cepat dengan 0,50 detik, sedangkan MSD memerlukan 0,59 detik. Kedua metrik ini penting untuk dipahami lebih dalam karena diantaranya menunjukkan perbedaan akurasi dan efisiensi dalam merekomendasikan buku. Hasil penelitian menunjukkan bahwa metode MSD lebih unggul dalam akurasi rekomendasi buku fiksi dibandingkan *cosine similarity*, sehingga lebih cocok untuk aplikasi yang mengutamakan ketepatan rekomendasi, sementara *cosine* lebih efisien dalam pemrosesan waktu untuk data yang besar.

Kata kunci: *cosine similarity; item-based collaborative filtering; msd; rekomendasi buku*

Abstract

The need for recommendations is increasingly crucial in the digital era, especially with the abundance of fiction book data from e-book platforms and digital libraries. This study aims to evaluate the effectiveness of item-based collaborative filtering using cosine similarity and Mean Squared Difference (MSD) metrics for book recommendations. The knowledge discovery in databases method was applied, encompassing data selection, pre-processing, transformation, data mining, and evaluation. The dataset includes 100,000 user ratings obtained from Kaggle's "Book Recommendation Dataset." Our findings show that the Mean Absolute Error for MSD is 0.152307, slightly better than cosine similarity at 0.152406. The Root Mean Squared Error for MSD is lower at 0.185551, compared to cosine similarity's 0.185636. However, Cosine Similarity is more efficient in processing time, with 0.50 seconds compared to 0.59 seconds for MSD. Understanding these metrics is crucial, as they reveal differences in accuracy and efficiency in book recommendation. The results indicate that MSD performs better in the accuracy of fiction book recommendations compared to cosine similarity, making it more suitable for applications prioritizing recommendation precision, while Cosine is more efficient for large data processing.

Keywords: *cosine similarity; item-based collaborative filtering; msd; book recommendation*



PENDAHULUAN

Sejak kemunculan era digital, teknologi menjadi kebutuhan penting dan transformasi informasi telah mengubah cara manusia berinteraksi dengan buku dan media lainnya (Daulay et al., 2020). Hal ini membawa dampak yang signifikan dalam kehidupan kita, baik dari segi positif maupun negatif (Addenan & Susanti, 2021). Dengan pesatnya perkembangan ilmu pengetahuan dan teknologi, era informasi menjadi lebih maju yang memungkinkan akses cepat ke data yang terus meningkat jumlahnya (Nasution & Zulaikha, 2023). Namun, melimpahnya data ini menjadi tantangan baru, terutama dalam pengelolaan data besar yang melibatkan pencarian konten relevan bagi pengguna (Yudistira, 2021). Tanpa rekomendasi yang efisien, pengguna berisiko menerima konten yang tidak sesuai, yang dapat berdampak negatif pada kepuasan mereka dalam mencari informasi (Suharya et al., 2021). Misalnya, dalam konteks buku, pembaca sering kali kebingungan memilih buku di antara ribuan pilihan, yang dapat mengakibatkan kesalahan dalam memilih konten yang sesuai dengan preferensi mereka (Widjaja & Palit, 2022).

Rekomendasi menjadi solusi penting dalam memecahkan masalah ini, dengan menyediakan daftar produk atau konten berdasarkan riwayat data pengguna sebelumnya (Putra & Santika, 2020). Dalam bidang literatur, seperti buku, rekomendasi membantu pembaca menemukan buku yang sesuai dengan selera dan minat mereka, sehingga dapat memperkaya pengalaman membaca (Hersani et al., 2022). Meningkatnya minat terhadap buku, terutama buku elektronik, semakin memperjelas pentingnya pengembangan model rekomendasi yang efektif (Nurbaiti & Mariah, 2020). Namun, rekomendasi yang ada saat ini belum sepenuhnya memuaskan pengguna karena hasil rekomendasi sering kali tidak relevan.

Salah satu metode yang banyak digunakan dalam rekomendasi adalah *Collaborative Filtering* (CF) (Muliadi & Lestari, 2019). CF memiliki dua pendekatan utama, yaitu *User-Based CF* (UBCF) dan *Item-Based CF* (IBCF) (Ardiansyah et al., 2022). UBCF merekomendasikan *item* berdasarkan preferensi pengguna yang mirip satu sama lain (Mahendra et al., 2024), sedangkan IBCF mengukur kesamaan antar-*item*, yang lebih efektif ketika ada lebih banyak data *item* daripada pengguna atau ketika pengguna tidak terlalu aktif memberikan *rating*. Dalam konteks rekomendasi buku, IBCF lebih unggul karena berfokus pada kesamaan antar buku berdasarkan riwayat pembaca sebelumnya, yang dapat menghasilkan rekomendasi yang lebih akurat (Widiyaningtyas et al., 2022).

Penelitian-penelitian sebelumnya telah menguji beberapa metrik untuk mengukur kesamaan *item* dalam IBCF, seperti *Jaccard Similarity* yang menghasilkan nilai *Root Mean Squared Error* (RMSE) 1,504 menekankan kesamaan berdasarkan jumlah pengguna yang memiliki preferensi serupa (Rana & Deeba, 2019). Namun, penelitian lain telah menunjukkan bahwa *Cosine Similarity* lebih efektif dalam mengukur kesamaan antar *item* dalam konteks buku (Ajeng et al., 2023; Rosita et al., 2022; Rosyad et al., 2023). Penelitian oleh Rosita et al. (2022) mencatat rata-rata nilai *Mean Absolute Error* (MAE) tertinggi mencapai 0,568 dan nilai *Mean Absolute Percentage Error* (MAPE) sebesar 22%. Sementara itu, penelitian Rosyad et al. (2023) menunjukkan bahwa sistem yang dikembangkan memperoleh nilai MAE mencapai 1,29, menandakan tingkat akurasi yang cukup baik. Penelitian lain oleh Ajeng et al. (2023) dihasilkan nilai MAE 0,89 dan mengkonfirmasi bahwa penggunaan *cosine similarity* dalam mengukur kemiripan antar buku efektif dalam menghasilkan rekomendasi yang akurat, tanpa kesalahan dalam proses rekomendasi.

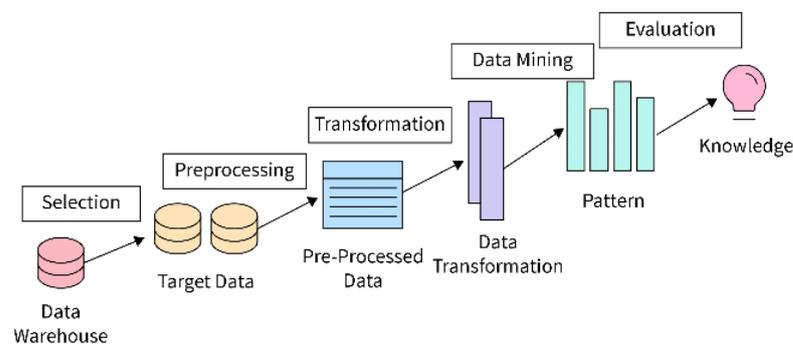
Cosine similarity memiliki keunggulan dalam mengevaluasi sudut antar-vektor *rating*, menghasilkan nilai antara 0 dan 1, di mana 1 menunjukkan kecocokan sempurna (Syah et al., 2021). Selain itu, *Mean Squared Difference* (MSD) juga sering digunakan untuk mengukur perbedaan rata-rata kuadrat antar-*item*, yang fokus pada nilai *rating* absolut. Meskipun *cosine similarity* lebih fokus pada pola kemiripan, MSD memberikan informasi lebih tentang

perbedaan antar *rating* yang berguna dalam rekomendasi. Kedua metrik ini memiliki kelebihan masing-masing dan telah diakui efektif dalam rekomendasi.

Penelitian ini berfokus pada analisis perbandingan antara *cosine similarity* dan MSD dalam konteks IBCF untuk rekomendasi buku fiksi. Fokus utama analisis adalah pada tiga metrik evaluasi: MAE, RMSE, dan waktu pengujian. Tujuan penelitian ini adalah untuk mengevaluasi efektivitas IBCF menggunakan metrik *cosine similarity* dan MSD dalam memberikan rekomendasi buku fiksi. Hasil penelitian ini diharapkan dapat memberikan kontribusi signifikan bagi pengembangan literatur terkait performa metrik dalam rekomendasi buku.

METODE

Penelitian ini menggunakan metode *Knowledge Discovery in Databases (KDD)* yang merupakan tahapan untuk menemukan informasi yang sebelumnya tidak diketahui dari data yang ada (Fauziah & Purnamasari, 2023). Gambar 1 menunjukkan proses KDD meliputi beberapa tahapan, yaitu pemilihan data, pra-pemrosesan, transformasi, *data mining*, hingga evaluasi hasil (Zai, 2022). Penelitian ini dimulai dengan tahap pemilihan data dari ‘*Book Recommendation Dataset*’ yang tersedia di situs Kaggle, berisi 278.857 pengguna, 1.048.575 *rating*, dan 271.282 buku. *Rating* dipilih sebagai fitur utama sebagai dasar perhitungan similaritas antar *item*.



Gambar 1. Tahapan KDD

Pada tahap *preprocessing*, beberapa langkah penting dilakukan untuk memastikan kualitas data. Dilakukan penghapusan duplikasi untuk menghindari bias pada model. Penggantian nilai kosong pada kolom ‘*Book-Rating*’, ‘*User-ID*’, dan ‘*ISBN*’ dengan 0 dilakukan agar tidak mengganggu analisis. Penghilangan baris dengan nilai ‘*Book-Rating*’ 0 juga penting, karena nilai ini tidak merepresentasikan *rating* valid dari pengguna. Selanjutnya, data diubah menjadi tipe numerik untuk memfasilitasi proses penambahan data, dan baris yang masih memiliki nilai NaN dihapus.

Sementara itu, pada tahap *transformation* data diubah ke dalam format yang lebih sesuai untuk *data mining*, termasuk normalisasi kolom ‘*Book-Rating*’ menggunakan *MinMaxScaler* agar nilai *rating* berada dalam rentang 0 hingga 1, sehingga algoritma similaritas dapat bekerja lebih optimal tanpa distorsi skala. Selanjutnya *outlier* dihapus menggunakan metode IQR untuk meningkatkan kualitas model, karena *outlier* dapat menyebabkan distorsi dalam perhitungan jarak atau similaritas antar-*item*. Metode IQR dipilih karena mampu mengidentifikasi *outlier* dengan baik berdasarkan distribusi data. Pengambilan sampel acak sebanyak 100.000 baris data dilakukan untuk menjaga efisiensi komputasi. Ini penting karena semakin besar ukuran data, semakin tinggi pula kompleksitas komputasinya.

Selanjutnya pada tahap *data mining* bertujuan untuk mengidentifikasi pola dalam data menggunakan metode tertentu, di tahap ini diterapkan *cosine similarity* dan MSD sebagai

metrik similaritas. *Cosine similarity* mengukur kemiripan antara dua item atau pengguna berdasarkan arah vektor rating mereka menggunakan persamaan 1. Persamaan 1 mengukur tingkat kemiripan antara dua item atau pengguna u dan v berdasarkan kesamaan arah vektor rating mereka. Bagian pembilang, $\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}$ menghitung jumlah perkalian dari rating yang diberikan oleh pengguna u dan v untuk setiap item yang sama-sama dirating oleh mereka. Sementara itu, bagian penyebut menghitung akar kuadrat dari jumlah kuadrat rating yang diberikan oleh masing-masing pengguna. Sementara MSD menghitung perbedaan rata-rata kuadrat antara nilai, guna mengevaluasi kesamaan dengan mempertimbangkan kesalahan antar-*item* yang dibandingkan menggunakan persamaan 2. Persamaan 2 mengukur rata-rata kuadrat selisih rating antara dua pengguna u dan v pada item yang sama. Bagian $(r_{ui} - r_{vi})^2$ menghitung kuadrat selisih rating, dan rata-rata diperoleh dengan membagi totalnya dengan $|I_{uv}|$.

$$\text{cosine_sim}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}} \quad (1)$$

$$\text{msd}(u, v) = \frac{1}{|I_{uv}|} \cdot \sum_{i \in I_{uv}} (r_{ui} - r_{vi})^2 \quad (2)$$

Cosine similarity dan MSD dipilih karena banyak digunakan dalam rekomendasi, dengan *cosine similarity* menghitung kemiripan antar-vektor berdasarkan sudut kosinus, yang cocok untuk data *rating* yang *sparse*, sementara MSD dengan mempertimbangkan perbedaan kuadrat antara dua *item*. Kinerja kedua metode dievaluasi dengan MAE yang mengukur rata-rata selisih absolut, dan RMSE yang memberikan penekanan lebih besar pada kesalahan yang lebih besar. Implementasi algoritma dilakukan menggunakan pustaka *Pandas* untuk manipulasi data dan *Surprise* untuk membangun model rekomendasi.

HASIL DAN PEMBAHASAN

Hasil

Hasil temuan kami pada tahap *Selection*, dataset ‘*Book Recommendation Dataset*’ dari Kaggle yang diunduh pada 27 September 2024 dipilih untuk penelitian ini. Dataset ini mencakup 95.522 pengguna dan 322.112 buku. Tabel 1 ditampilkan data dari *Ratings.csv* yang di dalamnya mencakup tiga kolom: *User-ID*, *ISBN*, dan *Book-Rating*. Kolom *User-ID* merepresentasikan pengguna yang memberikan *rating* pada buku. Kolom *ISBN* menunjukkan kode unik untuk buku yang di-*rating*. Sementara itu, kolom *Book-Rating* menunjukkan skor yang diberikan oleh pengguna untuk buku tersebut, dengan nilai antara 0 hingga 10.

Tabel 1. Dataset *rating* buku

<i>User-ID</i>	<i>ISBN</i>	<i>Book-Rating</i>
276725	034545104X	0
276726	155061224	5
276727	446520802	0
276729	052165615X	3
276729	521795028	6

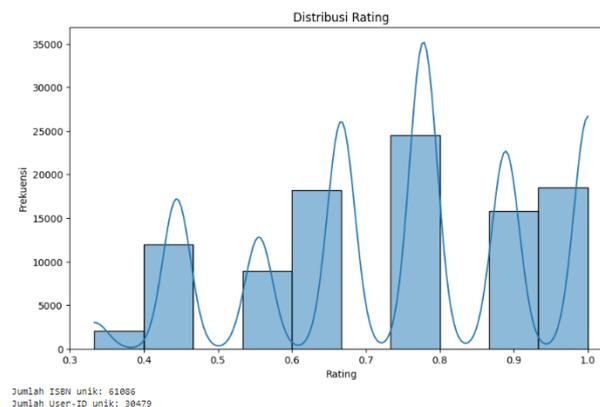
Setelah pemilihan dataset, tahap berikutnya adalah *preprocessing*, yang merupakan proses penting dalam mempersiapkan data sebelum analisis lebih lanjut. Langkah awal adalah melakukan pemeriksaan duplikasi. Terdapat 146 baris data yang merupakan duplikat, yang kemudian dihapus untuk menjaga kebersihan data. Selanjutnya, jika terdapat nilai kosong, mereka akan digantikan dengan nilai '0'. Penggunaan nilai '0' sebagai pengganti bertujuan untuk menjaga konsistensi tipe data serta mencegah potensi error saat dilakukan proses perhitungan dalam algoritma. Nilai *rating* buku '0' pada dataset ini digunakan untuk menunjukkan buku yang tidak diberi *rating* oleh pengguna. Oleh karena itu, hanya baris dengan nilai *rating* lebih dari 0 yang digunakan dalam analisis lanjutan, dengan tujuan agar rekomendasi berbasis *rating* dapat lebih bermakna. Setelah itu, setiap kolom dikonversi menjadi tipe numerik menggunakan *pd.to_numeric()* untuk memastikan konsistensi tipe data, dan semua baris yang masih memiliki nilai kosong setelah konversi dihapus menggunakan *dropna()*, untuk memastikan bahwa dataset siap untuk tahap analisis berikutnya.

	User-ID	ISBN	Book-Rating
1	276726.0	155061224.0	5
4	276729.0	521795028.0	6
6	276736.0	3257224281.0	8
7	276737.0	600570967.0	6
9	276745.0	342310538.0	10

Gambar 2. Hasil *preprocessing* data

Gambar 2 menunjukkan hasil dari proses *preprocessing* data untuk rekomendasi buku. Data tersebut mencakup tiga kolom, yaitu *User-ID*, *ISBN*, dan *Book-Rating*. Data di setiap kolomnya sudah numerik. Kolom *Book-Rating* berisi skor penilaian yang diberikan oleh pengguna terhadap buku, dengan nilai berkisar antara 1 hingga 10. Sudah tidak ada *rating* dengan isian 0. Tahap *preprocessing* ini sangat penting untuk memastikan data bersih dan siap digunakan dalam model rekomendasi berbasis CF.

Tahap *Transformation* melibatkan normalisasi data menggunakan *MinMaxScaler*. Normalisasi sangat penting dalam konteks algoritma KNN agar perhitungan jarak tidak terpengaruh oleh perbedaan skala antar variabel. Hasil normalisasi menempatkan nilai *rating* dalam rentang 0 hingga 1, seperti yang divisualisasikan pada Gambar 3. Selain itu, deteksi dan penghapusan *outlier* dilakukan menggunakan metode *Interquartile Range* (IQR) untuk meningkatkan akurasi. *Outlier* diidentifikasi berdasarkan jarak antara kuartil pertama (Q1) dan kuartil ketiga (Q3) dari distribusi data. *Outlier* dapat memengaruhi performa algoritma rekomendasi, karena nilai ekstrem dapat mendistorsi perhitungan jarak atau kemiripan. Selanjutnya, jika dataset memiliki lebih dari 100.000 baris, dilakukan pengambilan sampel acak sebanyak 100.000 baris untuk menjaga efisiensi proses tanpa mengurangi kualitas hasil.



Gambar 3. Grafik distribusi *rating*

Gambar 3 menunjukkan distribusi *rating* setelah dilakukan normalisasi menggunakan *MinMaxScaler*, yang mengubah skala *rating* ke dalam rentang 0 hingga 1. Grafik tersebut memperlihatkan bahwa setelah normalisasi, nilai *rating* terdistribusi dengan variasi yang signifikan di berbagai interval *rating*. Rentang nilai yang lebih tinggi, seperti sekitar 0,7 dan 0,8, memiliki frekuensi yang paling banyak, sementara beberapa nilai *rating* yang lebih rendah memiliki frekuensi yang lebih kecil.

Tahap *Data Mining* menerapkan algoritma *KNN Basic* dengan dua metode pengukuran similaritas, yaitu *MSD* dan *cosine similarity*. Algoritma *KNN* digunakan untuk menghitung kemiripan antar *item* buku, bukan antar pengguna. Gambar 4 dan gambar 5 menunjukkan dua implementasi *KNN* menggunakan *MSD* dan *Cosine Similarity*. Masing-masing menggunakan konfigurasi yang disesuaikan untuk perhitungan kemiripan. Evaluasi performa dilakukan dengan menggunakan metrik *MAE* dan *RMSE* untuk memastikan akurasi rekomendasi yang dihasilkan oleh kedua metode tersebut.

```
# Menghitung Similaritas menggunakan MSD
sim_options = {
    'name': 'msd',
    'user_based': False
}

# Menggunakan KNN
algo = KNNBasic(sim_options=sim_options)

# Perform cross validation
cv_results = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
results.append({
    'Algorithm': 'KNNBasic (msd)',
    'MAE': cv_results['test_mae'].mean(),
    'RMSE': cv_results['test_rmse'].mean()
})
```

Gambar 4. Pemodelan menggunakan *msd*

Hasil pada gambar 4 menunjukkan implementasi algoritma *KNN* dengan *MSD* sebagai metrik kemiripan. Opsi `sim_options` mengatur perhitungan kemiripan, dengan `msd` sebagai metrik dan `user_based: False` yang berarti kemiripan dihitung antar *item*, bukan antar pengguna. Algoritma `KNNBasic` diinisialisasi dengan opsi kemiripan ini. Selanjutnya, dilakukan *cross-validation* menggunakan dataset (`data`), dengan evaluasi performa menggunakan metrik *MAE* dan *RMSE* dalam 5 lipatan (`cv=5`). Rata-rata hasil *MAE* dan *RMSE* ditambahkan ke dalam daftar `results` bersama dengan nama algoritmanya.

```
# Menghitung Similaritas menggunakan Cosine Similarity
sim_options = {
    'name': 'cosine',
    'user_based': False
}

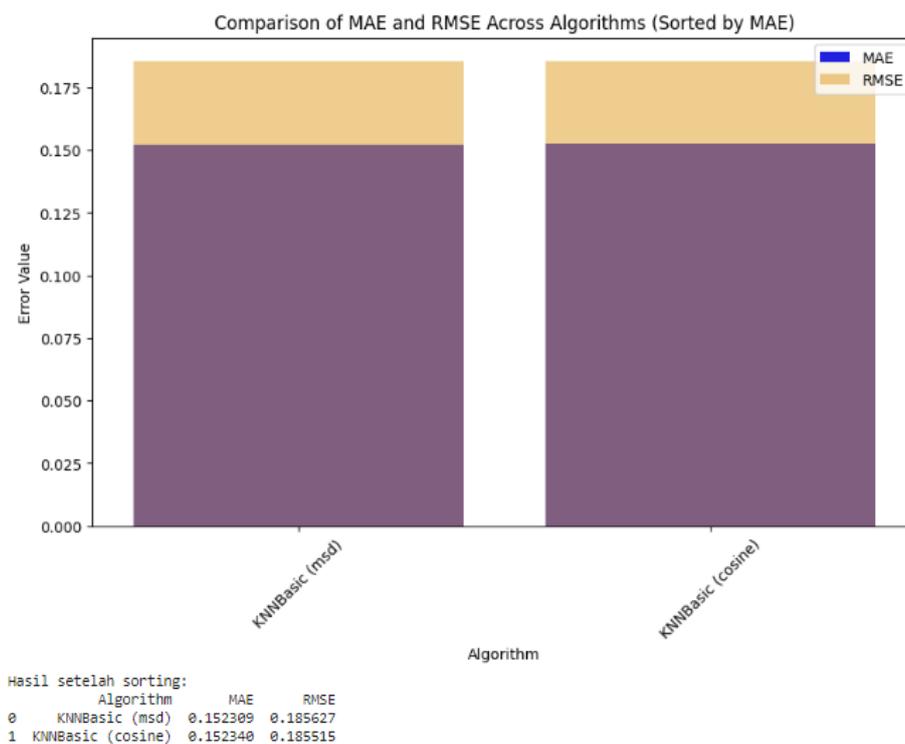
# Menggunakan KNN
algo = KNNBasic(sim_options=sim_options)

#perform cross validation
cv_results = cross_validate(algo, data, measures=['RMSE', 'MAE'], cv=5, verbose=True)
results.append({
    'Algorithm': 'KNNBasic (cosine)',
    'MAE': cv_results['test_mae'].mean(),
    'RMSE': cv_results['test_rmse'].mean()
})
```

Gambar 5. Pemodelan menggunakan *cosine similarity*

Kode pada gambar 5 menunjukkan implementasi algoritma KNN dengan *cosine similarity* sebagai metrik kemiripan. Opsi ``sim_options`` digunakan untuk mengatur perhitungan kemiripan, di mana ``cosine`` digunakan sebagai metrik, dan ``user_based: False`` menunjukkan bahwa kemiripan dihitung antar *item*, bukan antar pengguna. Algoritma ``KNNBasic`` diinisialisasi menggunakan opsi kemiripan ini. Kemudian, dilakukan *cross-validation* dengan dataset (``data``), menggunakan metrik evaluasi MAE dan RMSE dalam 5 lipatan (``cv=5``). Hasil rata-rata MAE dan RMSE ditambahkan ke daftar ``results`` bersama dengan nama algoritmanya.

Tahap ini telah menerapkan algoritma KNN *basic* dengan dua metode pengukuran similaritas, yaitu MSD dan *cosine similarity*. Dengan menggunakan MSD, diharapkan dapat memberikan prediksi yang lebih akurat dengan mempertimbangkan perbedaan *rating* antar buku, yang seharusnya menghasilkan nilai MAE dan RMSE yang lebih rendah. Sementara itu, *cosine similarity* dirancang untuk menangkap kemiripan dalam pola penilaian, yang diharapkan juga dapat menghasilkan metrik evaluasi yang baik.



Gambar 6. Komparasi MSD dan *Cosine Similarity*

Tahapan akhir didapatkan hasil evaluasi yang dilihat dari MAE, RMSE, dan *testing time*. Gambar 6 menunjukkan hasil perbandingan antara dua algoritma, MSD dan *cosine similarity*, berdasarkan nilai MAE dan RMSE. Grafik memperlihatkan bahwa kedua algoritma memiliki kinerja yang sangat mirip, dengan sedikit perbedaan pada nilai *error*. Nilai MAE dan RMSE untuk MSD sedikit lebih rendah dibandingkan dengan *cosine similarity*. Hasil ini menunjukkan bahwa MSD sedikit lebih unggul dalam hal akurasi prediksi.

Tabel 2. Rekapitulasi hasil perhitungan *data mining*

	<i>MAE</i>	<i>RMSE</i>	<i>Test Time</i>
MSD	0,152307	0,185551	0,59
<i>Cosine Similarity</i>	0,152406	0,185636	0,50

Hasil evaluasi yang diperoleh dari kedua algoritma ditampilkan dalam tabel 2. Kedua metode menunjukkan performa yang hampir sama, dengan nilai MAE dan RMSE yang sangat mirip. MAE menunjukkan rata-rata selisih absolut antara nilai *rating* prediksi dan aktual, sedangkan RMSE menekankan pada selisih yang lebih besar. Waktu pengujian juga diukur, yang memberikan *insight* penting mengenai kecepatan eksekusi algoritma dalam konteks dunia nyata. Dalam aplikasi rekomendasi nyata, waktu pengujian yang lebih cepat dapat memengaruhi pengalaman pengguna, terutama saat harus memberikan rekomendasi secara *real-time*.

Pembahasan

Hasil eksperimen menunjukkan bahwa pengujian rekomendasi menggunakan MSD dan *cosine similarity* menghasilkan nilai evaluasi yang hampir serupa, dengan MAE untuk MSD tercatat sebesar 0,152307, sedangkan untuk *cosine similarity* adalah 0,152406. RMSE untuk MSD adalah 0,185551, sedikit lebih baik dibandingkan *cosine similarity* yang menghasilkan 0,185636. Perbedaan utama antara MAE dan RMSE terletak pada konteks penggunaannya. MAE lebih penting pada aplikasi yang membutuhkan akurasi tinggi untuk pengalaman pengguna, seperti rekomendasi buku, sementara RMSE, yang sensitif terhadap *outlier*, dapat lebih informatif dalam konteks di mana kesalahan besar berpengaruh pada keputusan.

Perbedaan hasil ini dipengaruhi oleh karakteristik dari masing-masing metode. MSD mengukur kesamaan berdasarkan perbedaan absolut dalam *rating* antar pengguna, sehingga lebih efektif dalam menangkap keseragaman pola *rating*. Sementara itu, *cosine similarity* mempertimbangkan sudut antar vektor *rating*, yang lebih mengutamakan arah preferensi dibandingkan nilai absolut *rating*. Dalam konteks dataset ini, MSD sedikit lebih unggul karena mampu mencerminkan stabilitas pola preferensi pengguna yang memiliki kemiripan *rating*, sedangkan *Cosine Similarity* yang cenderung lebih sensitif terhadap perbedaan kecil dalam nilai *rating* antar pengguna.

Meskipun banyak penelitian sebelumnya mengunggulkan *cosine similarity*, hasil penelitian ini menunjukkan bahwa MSD lebih efektif dalam prediksi kesamaan antar *item*. Pada penelitian sebelumnya, seperti yang dilakukan oleh Ajeng et al. (2023) menghasilkan MAE sebesar 0.83. Sementara penelitian oleh Rosita et al. (2022) mencatatkan MAE sebesar 0.568, Lalu penelitian lain oleh Rosyad et al. (2023) menunjukkan MAE sebesar 1.291. Penelitian tersebut umumnya berfokus pada pola kesamaan arah preferensi antar pengguna, tanpa mempertimbangkan pengaruh variasi absolut dalam nilai *rating*. Dalam penelitian ini, MSD lebih diutamakan karena kemampuannya dalam menangkap kesamaan berdasarkan perbedaan absolut antar *rating*. Perhitungan berdasarkan MSD memungkinkan sistem untuk menangkap keseragaman preferensi pengguna dengan lebih stabil, tanpa terpengaruh oleh perbedaan kecil yang biasanya dapat mengurangi akurasi pada *cosine similarity*.

Perbedaan kecil antara MAE dan RMSE kedua metode ini, meskipun tidak signifikan secara statistik, tetap memiliki makna praktis. Dalam konteks pengguna, prediksi yang lebih akurat dari MSD dapat meningkatkan pengalaman rekomendasi, terutama dalam aplikasi di mana pengguna mencari referensi berkualitas tinggi, seperti dalam rekomendasi buku untuk keperluan akademis. Pada situasi tersebut, akurasi lebih diutamakan dibandingkan dengan efisiensi waktu pemrosesan, meskipun *cosine similarity* lebih cepat dengan waktu 0,50 detik dibandingkan 0,59 detik untuk MSD. Namun, selisih waktu 0,09 detik mungkin tidak terlalu signifikan pada skala besar, tetapi tetap berpengaruh jika jutaan prediksi dilakukan secara bersamaan.

Kontribusi penelitian ini adalah penemuan bahwa MSD lebih andal dalam konteks rekomendasi buku fiksi, mendukung argumen bahwa algoritma ini lebih baik untuk akurasi dibandingkan efisiensi waktu. Temuan ini memiliki relevansi praktis untuk meningkatkan rekomendasi di bidang pendidikan dan literatur. Selain itu, perbedaan hasil antara MSD dan

cosine similarity memberikan wawasan tentang pengaruh distribusi *rating* dan densitas data terhadap performa algoritma, terutama buku fiksi. Buku fiksi memiliki karakteristik unik seperti dalam preferensi pengguna, seperti genre atau gaya penulisan, sehingga penting mempertimbangkan variasi data tersebut saat memilih metode rekomendasi yang tepat untuk memberikan hasil yang lebih akurat dan relevan.

SIMPULAN

Penelitian ini menunjukkan bahwa metode IBCF dengan MSD menghasilkan akurasi lebih tinggi dengan MAE 0,152307 dibandingkan *cosine similarity*, namun *cosine* unggul dalam waktu eksekusi 0,50 detik untuk 100.000 data *rating*. Keunggulan akurasi MSD ini disebabkan oleh pendekatannya yang mengukur kesamaan berdasarkan perbedaan absolut antar *rating*, sehingga lebih efektif dalam menangkap pola keseragaman preferensi pengguna pada dataset yang variatif. Kontribusi utama penelitian ini adalah bukti bahwa MSD lebih cocok untuk skenario yang mengutamakan akurasi, terutama dalam rekomendasi buku di bidang pendidikan dan literatur. Untuk penelitian selanjutnya, disarankan menguji algoritma lain seperti SVD atau NMF, serta menggunakan dataset yang lebih besar untuk menguji skalabilitas dan generalisasi model.

REFERENSI

- Addenan, R., & Susanti, W. (2021). Penerapan Metode Rank Order Centroid dan Additive Ratio Assessment Pada Aplikasi Rekomendasi Supplier. *Edumatic: Jurnal Pendidikan Informatika*, 5(1), 31–40. <https://doi.org/10.29408/edumatic.v5i1.3252>
- Ajeng, P., Sukmawati, S., Hiryanto, L., Viny,), & Mawardi, C. (2023). Implementasi Metode Collaborative Filtering Based Untuk Sistem Rekomendasi Buku Fiksi. *Jurnal Ilmu Komputer Dan Sistem Informasi*, 11(2), 1–5. <https://doi.org/https://doi.org/10.24912/jiksi.v11i2.25999>
- Ardiansyah, A. H., Widiyanto, A., & Nugroho, S. (2022). Implementation of the Item-Based Collaborative Filtering Method on a Web-Based Culinary Tourism Recommendation System (Case Study: Magelang City). *Borobudur Informatics Review*, 2(2), 47–60. <https://doi.org/10.31603/binr.6731>
- Daulay, R. S., Pulungan, H., Noviana, A., & Hurhaliza, S. (2020). Manfaat Teknologi Smartphone Di Kalangan Pelajar Sebagai Akses Pembelajaran Di Masa Pandemi Corona-19. *Jurnal Pendidikan Islam*, 1(1), 29–43. <https://doi.org/https://doi.org/10.56114/al-ulum.v1i1.7>
- Fauziah, R., & Purnamasari, A. I. (2023). Implementasi Algoritma K-Means pada Kasus Kekerasan Anak dan Perempuan Berdasarkan Usia. *Hello World Jurnal Ilmu Komputer*, 2(1), 34–41. <https://doi.org/10.56211/helloworld.v2i1.232>
- Hersani, R., Kurniawan, Pangestu, S. L., Edi, M., & Wahyunita, N. N. (2022). Perancangan Sistem Informasi Rekomendasi Buku Dengan Metode Extreme Programming. *Jurnal Teknologi Sistem Informasi Dan Sistem Komputer TGD*, 5(2), 293–300. <https://ojs.trigunadharma.ac.id/index.php/jsk/index>
- Mahendra, R. R., Anggraeny, F. T., & Wahanani, H. E. (2024). Implementasi Item-Based Collaborative Filtering Untuk Rekomendasi Film. *Repeater: Publikasi Teknik Informatika Dan Jaringan*, 2(3), 213–221. <https://doi.org/10.62951/repeater.v2i3.140>
- Muliadi, K. H., & Lestari, C. C. (2019). Rancang Bangun Sistem Rekomendasi Tempat Makan Menggunakan Algoritma Typicality Based Collaborative Filtering. *Techno.Com*, 18(4), 275–287. <https://doi.org/https://doi.org/10.33633/tc.v18i4.2515>
- Nasution, C. F., & Zulaikha, S. R. (2023). Penggunaan Konsep User Experience Terhadap Layanan Situs Web Perpustakaan Universitas Islam Sumatra Utara. *Jurnal Pustaka Budaya*, 10(2), 2442–7799. <https://doi.org/https://doi.org/10.31849/pb.v10i2.14683>

- Nurbaiti, D., & Mariah. (2020). Pengaruh Sikap pada Ebook dan Sikap pada Buku Fisik terhadap Minat Baca Masyarakat di Era Industri 4.0. *Jurnal Logistik Indonesia*, 4(1), 74–80. <https://doi.org/https://doi.org/10.31334/logistik.v4i1.875>
- Putra, A. I., & Santika, R. R. (2020). *Implementasi Machine Learning dalam Penentuan Rekomendasi Musik dengan Metode Content-Based Filtering*. 4(1), 121–130. <https://doi.org/10.29408/edumatic.v4i1.2162>
- Rana, A., & Deeba, K. (2019). Online Book Recommendation System Using Collaborative Filtering (with Jaccard Similarity). *Journal of Physics: Conference Series*, 1362(1), 1–8. <https://doi.org/10.1088/1742-6596/1362/1/012130>
- Rosita, A., Puspitasari, N., & Kamila, V. Z. (2022). Rekomendasi Buku Perpustakaan Kampus dengan Metode Item-Based Collaborative Filtering. *Sebatik*, 26(1), 340–346. <https://doi.org/10.46984/sebatik.v26i1.1551>
- Rosyad, S., Mahendra, D., & Azizah, N. (2023). Sistem Rekomendasi Buku di Perpustakaan Daerah Jepara Menggunakan Metode Item-Based Collaborative Filtering. *Biner: Jurnal Ilmiah Informatika Dan Komputer*, 2(2), 76–81. <https://doi.org/https://doi.org/10.32699/biner.v2i2.3934>
- Suharya, Y., Herdiana, Y., Putri, N. I., & Munawar, Z. (2021). Sistem Rekomendasi untuk Toko Online Kecil dan Menengah. *Tematik: Jurnal Teknologi Informasi Komunikasi*, 8(2), 176–185.
- Syah, T. H., Syah, S., & Nurmalasari, N. (2021). Minat Masyarakat Terhadap Karakteristik Informasi Teksual dari Kementerian Lingkungan Hidup dan Kehutanan di Media Sosial. *Jurnal Komunika: Jurnal Komunikasi, Media Dan Informatika*, 10(2), 62–72. <https://doi.org/10.31504/komunika.v10i2.4442>
- Widiyaningtyas, T., Ardiansyah, M. I., & Adji, T. B. (2022). Recommendation Algorithm Using SVD and Weight Point Rank (SVD-WPR). *Big Data and Cognitive Computing*, 6(4), 1–15. <https://doi.org/10.3390/bdcc6040121>
- Widjaja, A. A., & Palit, H. N. (2022). Hybrid Recommendation System untuk Peminjaman Buku Perpustakaan dengan Collaborative dan Content-Based Filtering. *Jurnal Infra*, 10(2), 1–6.
- Yudistira, N. (2021). Peran Big Data dan Deep Learning untuk Menyelesaikan Permasalahan Secara Komprehensif. *EXPERT: Jurnal Manajemen Sistem Informasi Dan Teknologi*, 11(2), 78–89. <https://doi.org/10.36448/expert.v11i2.2063>
- Zai, C. (2022). Implementasi Data Mining Sebagai Pengolahan Data. *Jurnal Portal Data*, 2(3), 1–12.