Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

Optimasi K-Means Menggunakan Algoritma Genetika pada Metode User-based Collaborative Filtering

Axl Adilla[1], Affi Nizar Suksmawati[2,*] ,Affifah Mutiara Pertiwi[3], Kharis Suryandaru Pratama[4]

[1], Informatika, Universitas Jenderal Soedirman, Purwokerto, Indonesia
[2], Sistem dan Teknologi Informasi, Universitas Widya Gama, Malang, Indonesia
[3], Informatika, Universitas Telkom, Surabaya, Indonesia
[4], Perumdam Tirta Satria, Purwokerto, Indonesia
Email: [1]axl.adilla@unsoed.ac.id, [2,*]affi.nizar@widyagama.ac.id, [3]affifahmutiara@telkomuniversity.ac.id, [4]kharis.suryandaru@mail.ugm.ac.id
Email Penulis Korespondensi: affi.nizar@widyagama.ac.id[*]
Submitted: **28/11/2025**; Accepted: **07/12/2025**; Published: **31/12/2025**

Abstrak– Collaborative filtering merupakan teknik sistem rekomendasi yang menggunakan informasi rating dari beberapa pengguna untuk memprediksi rating suatu item bagi pengguna tertentu. Namun, tidak semua pengguna memberikan rating pada seluruh item. Hal ini menyebabkan ketidakmampuan sistem dalam menentukan nearest neighborhood, sehingga rekomendasi yang dihasilkan menjadi lemah. Penelitian ini mengusulkan penggunaan Algoritma K-Means untuk mengelompokkan neighborhood yang sesuai. Penentuan awal titik pusat klaster pada Algoritma K-Means dioptimalkan menggunakan Algoritma Genetika. Evaluasi dilakukan dengan memvariasikan jumlah klaster optimal pada beberapa metode pengukuran yang digunakan, yaitu Jaccard Similarity Coefficient, Sørensen–Dice Coefficient, dan Hamming Coefficient. Hasil pengujian menggunakan pengukuran Jaccard Similarity Coefficient, Sørensen–Dice Coefficient, dan Hamming Coefficient memperoleh nilai fitness masing-masing sebesar 4.490, 4.979, dan 4.964 untuk jumlah klaster optimal 4 dan 6. Sementara itu, nilai MAPE rata-rata untuk ketiga metode pengukuran kemiripan tersebut sebesar 60%.

Kata Kunci: Algoritma Genetika; Algoritma K-Means; User-Based Collaborative Filtering; Sistem Rekomendasi; Jaccard Similarity Coefficient, Sørensen–Dice Coefficient, Hamming Coefficient

Abstract– Collaborative filtering is a recommendation system technique that uses rating information from several users to predict the rating of a particular user item. However, not all users rate the entire item. This causes the system's inability to determine the nearest neighborhood, resulting in weak recommendations. This study proposes the K-Means Algorithm to group the appropriate neighborhood. The initial determination of the cluster center point in the K-Means Algorithm is optimized using the Genetic Algorithm. The evaluation was carried out by varying the optimal number of clusters on several measuring methods used, namely Jaccard Similarity Coefficient, Sørensen–Dice Coefficient, and Hamming Coefficient. The test results using Jaccard Similarity Coefficient, Sørensen–Dice Coefficient, and Hamming Coefficient measurements obtained fitness values of 4,490, 4,979 and 4,964 for the optimal number of clusters 4 and 6. While the average MAPE value for the three similarity measurement methods is 60%.

Keywords: Genetic Algorithm; K-Means Algorithm; User-Based Collaborative Filtering; Recommender System; Jaccard Similarity Coefficient, Sørensen–Dice Coefficient, Hamming Coefficient

1. PENDAHULUAN

Sistem rekomendasi merupakan teknik yang dapat menyarankan suatu informasi yang berguna dan membantu pengguna dalam proses pengambilan keputusan [1]. Informasi yang diberikan sistem rekomendasi dapat mengarahkan pengguna pada *item* yang sesuai kebutuhan dan keinginan [2]. Secara garis besar sistem rekomendasi dibagi menjadi dua pendekatan, yaitu *Content-based Filtering* dan *Collaborative Filtering* [3]. Pendekatan *Content-based* mengukur keterkaitan *item-item* yang memiliki kesamaan konten. *Content-based Filtering* merupakan pendekatan paling tepat untuk membangun sistem rekomendasi dengan data berupa teks atau dokumen [4]. Sedangkan pendekatan *Collaborative Filtering* menggunakan informasi rating dari beberapa pengguna untuk memprediksi rating pengguna tertentu [5].

Collaborative Filtering dibagi menjadi dua pendekatan, yaitu *Item-based Collaborative Filtering* dan *User-based Collaborative Filtering*. Menggunakan *Item-based Collaborative Filtering* prediksi untuk rekomendasi didasarkan oleh kemiripan antar *item*. Sedangkan menggunakan *User-based Collaborative Filtering* prediksi untuk rekomendasi didasarkan oleh kemiripan antar pengguna. Pada *User-based Collaborative Filtering*, vektor pengguna aktif dibandingkan dengan vektor pengguna lain dengan ukuran kemiripan tertentu. Nilai kemiripan antar pengguna didapatkan dengan metode pengukuran kemiripan seperti *Jaccard Similarity Coefficient*, *Sørensen–Dice Coefficient*, dan *Hamming Coefficient*.

Beberapa penelitian mencoba mengembangkan sistem rekomendasi dengan metode pengukuran kemiripan tertentu. Penelitian Hadi (2020) menggunakan *Jaccard Similarity Coefficient* untuk menentukan nilai similaritas pengguna pada sistem rekomendasi film [6]. Hasil Penelitian diukur menggunakan *Mean Reciprocal Rank* (MRR) memperoleh nilai rata-rata sebesar 0.1507 dengan tingkat ketepatan cukup baik. Penelitian Ariyanto (2020) membandingkan metode pengukuran kemiripan *Jaccard Similarity Coefficient* dan *Sørensen–Dice Coefficient* untuk sistem rekomendasi artikel berita [7]. Hasil pengujian menunjukkan bahwa metode pengukuran kemiripan *Jaccard Similarity Coefficient* lebih baik dari sisi waktu komputasi.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

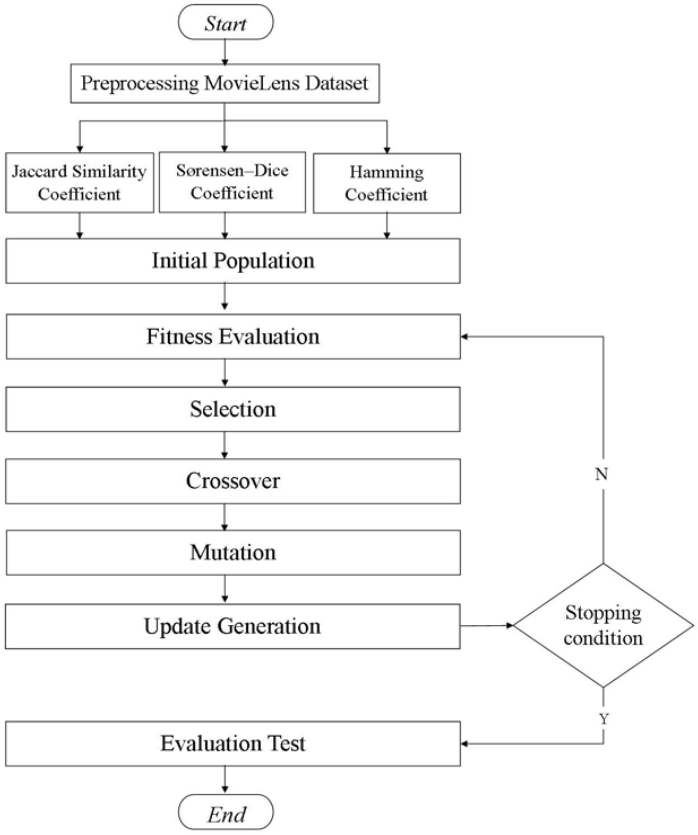
Permasalahan yang terjadi pada *User-based Collaborative Filtering* adalah kurangnya informasi yang cukup karena keterbatasan pengguna untuk memberikan *rating* pada keseluruhan *item*. Hal ini menyebabkan matriks antar pengguna-*item* yang jarang dan nilai kemiripan antar pengguna yang rendah [8]. Sehingga sistem tidak mampu menemukan tetangga yang terdekat dan menghasilkan sistem rekomendasi yang lemah. Permasalahan ini disebut dengan *data sparsity* [9]. Algoritma *K-Means* dapat digunakan untuk mengatasi permasalahan *data sparsity* dengan mengelompokkan nilai kemiripan pengguna sehingga *neighborhood* dapat ditentukan dengan tepat. Namun Algoritma *K-Means* memiliki kelemahan dalam penentuan awal titik pusat *cluster* yang ditentukan secara acak [10]. Penentuan titik pusat *cluster* di awal iterasi proses sangat berpengaruh pada kecepatan pemrosesan dan akurasi penggolongan jenis *cluster* [11]. Kecepatan pemrosesan dipengaruhi oleh jumlah iterasi pada proses komputasi algoritma *K-Means* [12].

Algoritma Genetika dapat digunakan untuk mengoptimasi titik pusat *cluster* yang dipilih secara acak oleh Algoritma *K-Means*. Beberapa penelitian yang menggunakan Algoritma *K-Means* dengan optimasi Algoritma Genetika diantaranya penelitian Taslim (2021) menggunakan Algoritma *K-Means* dengan optimasi Algoritma Genetika untuk target pemanfaatan air bersih [13]. Hasil validitas *cluster* dengan optimasi diukur dengan menggunakan metode DBI sebesar 2.068 lebih baik dari *cluster* tanpa optimasi dengan nilai DBI sebesar 2.614. Penelitian Istianto (2021) menggunakan *K-Means* dengan optimasi titik pusat *cluster* menggunakan Algoritma Genetika untuk klasifikasi jumlah produk makanan [14]. Hasil pengujian menggunakan *cross validation* untuk lima *fold* berhasil mendapatkan rata-rata akurasi sebesar 50.58%.

Sebagian besar penelitian terdahulu hanya mengandalkan perhitungan kemiripan langsung antar pengguna tanpa mekanisme struktural untuk memperbaiki pemilihan *neighborhood* pada kondisi data *sparsity*. Selain itu, meskipun *K-Means* berpotensi membantu pengelompokan pengguna, proses penentuan titik pusat *cluster* secara acak masih menjadi kelemahan karena menghasilkan kualitas klaster yang tidak stabil. Dengan demikian, diperlukan pendekatan yang mampu mengoptimalkan pembentukan *cluster* serta meningkatkan stabilitas dan akurasi proses identifikasi *neighborhood*. Pada penelitian ini dilakukan pengembangan sistem rekomendasi metode *User-based Collaborative Filtering* dengan membandingkan beberapa metode pengukuran kemiripan yang terdiri dari *Jaccard Similarity Coefficient*, *Sørensen–Dice Coefficient*, dan *Hamming Coefficient*. Algoritma *K-Means* digunakan untuk pengelompokan *neighborhood* yang sesuai. Sedangkan optimasi Algoritma Genetika digunakan untuk penentuan awal titik pusat *cluster* yang dipilih secara acak oleh Algoritma *K-Means*.

2. METODOLOGI PENELITIAN

Secara umum penelitian yang dilakukan memiliki alur seperti pada gambar 1.



Gambar 1. Alur Penelitian

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

Alur penelitian pada Gambar 1 dimulai dengan proses *preprocessing dataset MovieLens* 100K, yaitu mengubah data *rating* menjadi matriks pengguna. Matriks ini kemudian ditransformasi menjadi matriks kemiripan antar pengguna menggunakan tiga metode pengukuran kemiripan, yaitu *Jaccard*, *Sørensen–Dice*, dan *Hamming*. Nilai kemiripan tersebut menjadi *input* untuk proses klasterisasi *K-Means*, di mana Algoritma Genetika digunakan untuk mengoptimalkan penentuan pusat *cluster* awal agar kualitas klaster lebih stabil. Setiap individu GA mewakili kandidat pusat *cluster*, kemudian dievaluasi melalui proses *K-Means* dan perhitungan prediksi *rating* menggunakan *Pearson Correlation*. Nilai *error* prediksi diukur menggunakan *Mean Absolute Percentage* Error (MAPE). Proses seleksi, *crossover*, mutasi, dan elitisme diulang hingga mencapai jumlah generasi yang ditentukan. Individu terbaik dari GA digunakan untuk melakukan prediksi pada data testing. Alur tersebut memastikan bahwa model melalui tahapan *preprocessing*, perhitungan kemiripan, optimasi klaster, dan evaluasi performa prediksi secara berurutan dan terstruktur.

Pada tahap *preprocessing* dataset. *Dataset* yang digunakan adalah *dataset MovieLens* 100K. *Dataset* ini terdiri dari data *rating user* terhadap *item* dan data atribut *item*, yaitu *genre*. Jumlah data *rating* sebanyak 100.000, terdiri dari 943 *user* dan 1682 *item*. Atribut pada *dataset MovieLens* seperti terlihat pada Tabel 1.

Tabel 1. Format Dataset *MovieLens*

UserId	*MovieId*	*Rating*
1	1	4
1	3	3
…	…	…

Berdasarkan Tabel 1, *UserId* merupakan id pengguna, *MovieId* merupakan id film, dan *Rating* merupakan nilai preferensi dari pengguna terhadap film tertentu. Nilai *rating* berada pada rentang nilai 0 sampai 5. *Preprocessing* data dilakukan pada atribut *dataset MovieLens* dengan merubah menjadi matriks 2D yang merepresentasikan *rating user* dengan ukuran (jumlah *user* x jumlah film) seperti pada Tabel 2.

Tabel 2. Matriks *Pengguna-Item*

Movie Id / *User Id*	1	2	3	4	…
1	4		3		…
2		3	4	2	…
…	…	…	…	…	…

Berdasarkan Tabel 2, pengguna dengan *UserId* 1 memberikan *rating* untuk film pada *MovieId* 1 sebesar 4, dan *MovieId* 3 sebesar 3. Tahap selanjutnya setelah dilakukan *preprocessing* adalah mentransformasi matriks 2D sehingga merepresentasikan relasi antar *user* dengan ukuran (jumlah *user* x jumlah *user*) seperti pada Tabel 3.

Tabel 3. Matriks Kemiripan Pengguna

User Id	1	2	3	…
1	1	0.4	0.8	…
2	0.4	1	0.6	…
3	0.8	0.6	1	…
…	…	…	…	…

Berdasarkan Tabel 3, nilai kemiripan antar pengguna diperoleh dari perhitungan nilai *rating* yang diberikan pengguna terhadap *item* (pada Tabel 2) menggunakan metode pengukuran kemiripan tertentu. Metode pengukuran kemiripan yang digunakan terdiri dari *Jaccard Similarity Coefficient*, *Sørensen–Dice Coefficient*, dan *Hamming*

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

Coefficient. Menggunakan *Jaccard Similarity Coefficient* nilai kemiripan selalu berada diantara 0 sampai 1. Persamaan 1 berikut merupakan persamaan untuk menghitung nilai kemiripan menggunakan *Jaccard Similarity Coefficient* [15].

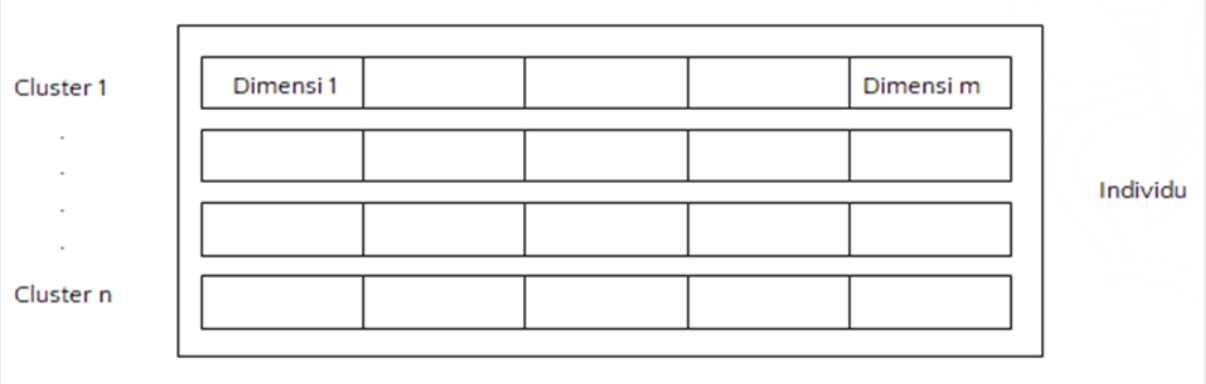
$$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A|+|B|-|A \cap B|} \quad (1)$$

Berdasarkan persamaan 1, J(A, B) menghitung kemiripan pengguna A dan B. A merupakan jumlah fitur yang dimiliki pengguna A sedangkan B merupakan jumlah fitur yang dimiliki pengguna B. |A∩B| merupakan jumlah fitur pengguna A yang sama dengan jumlah fitur pengguna B. |A∪B| merupakan jumlah fitur pengguna A ditambah dengan jumlah fitur pengguna B dikurangi dengan jumlah fitur pengguna A yang sama dengan jumlah fitur pengguna B. Metode pengukuran kemiripan *Jaccard Similarity Coefficient* akan dibandingkan dengan metode *Sørensen–Dice Coefficient*. Persamaan 2 berikut merupakan persamaan untuk menghitung nilai kemiripan menggunakan *Sørensen–Dice Coefficient* [16].

$$D(A,B) = \frac{2|A \cap B|}{|A|+|B|} \quad (2)$$

Berdasarkan persamaan 2, D(A, B) menghitung kemiripan pengguna A dan B dimana pada perhitungan kemiripan dilakukan dengan cara dua kali jumlah fitur yang sama pada kedua pengguna dibagi dengan jumlah fitur pada pengguna A dan pengguna B. Metode pengukuran kemiripan *Jaccard Similarity Coefficient* dan *Sørensen–Dice Coefficient*, akan dibandingkan dengan metode *Hamming Coefficient*. Menggunakan *Hamming Coefficient* pengukuran jarak antara dua objek yang ukurannya sama dilakukan dengan membandingkan nilai-nilai yang terdapat pada kedua objek pada posisi yang sama. Dimana penjumlahan dilakukan pada nilai yang memiliki perbedaan pada kedua objek. Penerapan metode *Hamming Coefficient* dapat diilustrasikan sebagai berikut, pengguna A memberikan rating sebagai berikut (2, 1, 2, 5, 0) pengguna B memberikan rating sebagai berikut (2, 0, 1, 2, 0), maka nilai *Hamming Coefficient* antara pengguna A dan B adalah sebesar 3.

Setelah didapatkan matriks kemiripan pengguna, kemudian dilakukan proses pengelompokan nilai kemiripan dengan Algoritma *K-Means*. Proses pengelompokan nilai kemiripan bertujuan untuk mendapatkan *neighborhood* yang sesuai. Pada tahap ini digunakan Algoritma Genetika untuk optimasi penentuan awal titik pusat *cluster* yang dilakukan secara acak oleh Algoritma *K-Means*. Inisialisasi individu pada Algoritma Genetika merupakan titik pusat *cluster* pada Algoritma *K-Means*. Gambar 2 merupakan inisialisasi individu pada Algoritma Genetika.



Gambar 2. Inisialisasi Individu Algoritma Genetika

Berdasarkan Gambar 2, proses pengkodean menggunakan *real encoding*, dimana tiap individu merepresentasikan solusi terhadap inisialisasi nilai vektor *K-Means* agar tidak terjebak pada *local minima*, kromosom merepresentasikan nilai inisial pada *cluster* 1 hingga *cluster* ke-n, dan gen merepresentasikan fitur 1 hingga fitur ke-m. Pada tahap ini dilakukan inisialisasi parameter Algoritma Genetika yang terdiri dari jumlah populasi, generasi, probabilitas *crossover* dan probabilitas mutasi. Tahap evaluasi pada Algoritma Genetika adalah setiap solusi dari individu diimplementasikan kedalam *K-Means* untuk mendapatkan *cluster* dari pengguna, untuk kemudian memprediksi *rating* dengan *Pearson Correlation*. Persamaan untuk *Pearson Correlation* adalah sebagai berikut [17].

$$w = \frac{\sum^{n}_{i=1}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum^{n}_{i=1}(x_i - \bar{x})^2}\ \sqrt{\sum^{n}_{i=1}(y_i - \bar{y})^2}} \quad (3)$$

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}} \quad (4)$$

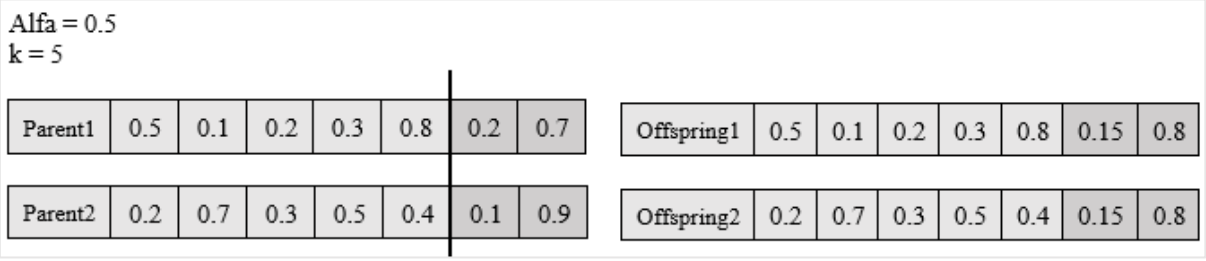
Berdasarkan persamaan 3 dilakukan perhitungan bobot *w*, n merupakan jumlah *item*, pada penelitian ini adalah *movie*, x_i merupakan penilaian *movie* dari *user x* dan nilai $\bar{x}$ merupakan rata-rata rating dari *user x*. Nilai y_i merupakan penilaian *movie* dari user y dan nilai $\bar{y}$ merupakan penilaian rata-rata dari *user y*. Hasil perkalian rating *user x* dan *y* kemudian dibagi dengan nilai *standar deviasi x* dan *y*. Perhitungan bobot *w* dilakukan pada setiap *user*. Persamaan 4 menghitung $P_{a,i}$ yang merupakan prediksi nilai *rating a* untuk *user i*, r_u yang merupakan penilaian pengguna u terhadap *movie* ke-i akan dikurangkan dengan rata-rata *rating* dari *user u* ($\bar{r_u}$). Hasilnya dikalikan dengan bobot *w* yang telah dihitung sebelumnya kemudian dibagi dengan total bobot $w_{a,u}$. Nilai tersebut ditambah dengan rata-rata *rating a* ($\bar{r_a}$). *User u* adalah anggota K yang merupakan tetangga dalam satu *cluster*. Setelah prediksi *rating* ditentukan kemudian dilakukan perhitungan selisih nilai *error* antara nilai *rating* yang diprediksi dengan nilai *rating* yang sebenarnya. Selisih nilai *error* dihitung menggunakan *Mean Absolute Percentage Error* (MAPE). Tujuan fungsi *fitness* adalah untuk memaksimalkan nilai sehingga fungsi *fitness* dapat didefinisikan pada persamaan 5.

$$Fitness = \frac{1}{\sum_{t=1}^{n} |(A_t - F_t)/A_t| + 0.0001} \times 100\% \quad (5)$$

Tahap selanjutnya pada Algoritma Genetika adalah proses seleksi. Pada penelitian ini dilakukan proses seleksi menggunakan operator seleksi *Roulette Wheel*. Menggunakan metode *Roulette Wheel*, setiap individu dipetakan menurut nilai *fitness*. Jika fi merupakan nilai *fitness* untuk individu dalam suatu i populasi. Maka probabilitas individu terpilih dapat didefinisikan pada persamaan 6.

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \quad (6)$$

Berdasarkan persamaan 6, nilai P_i merupakan *fitness* dari individu i, sedangkan N merupakan jumlah total individu dalam populasi tersebut. Pada proses seleksi akan dipilih kromosom dalam populasi sebagai induk untuk proses *crossover* [18]. Pada penelitian ini, operator *crossover* yang digunakan adalah *Simple Arithmetic Crossover*. Nilai gen yang mengalami perhitungan secara *arithmetic crossover* merupakan nilai gen yang dimulai dari titik k hingga nilai gen terakhir secara keseluruhan. Sehingga penentuan nilai k menjadi titik dimulainya *arithmetic crossover*. Penerapan proses *crossover* menggunakan *Simple Arithmetic Crossover* dapat diilustrasikan seperti terlihat pada gambar 3 berikut.



Gambar 3. Ilustrasi simple arithmetic crossover

Berdasarkan gambar 3, diperoleh persamaan 7 dan 8 untuk proses perhitungan *Simple Arithmetic Crossover* sebagai berikut.

$$Offspring1 = (x1, \dots, x(k-1), \alpha \cdot y(k+1) + (1-\alpha) \cdot x(k+1), \dots, \alpha \cdot y\_n + (1-\alpha) \cdot x\_n) \quad (7)$$
$$Offspring2 = (y1, \dots, y(k-1), \alpha \cdot x(k+1) + (1-\alpha) \cdot y(k+1), \dots, \alpha \cdot x\_n + (1-\alpha) \cdot y\_n) \quad (8)$$

$$Offspring1_6 = (0.5 \times 0.1 + (1-0.5) \times 0.2) = 0.15$$
$$Offspring1_7 = (0.5 \times 0.9 + (1-0.5) \times 0.7) = 0.8$$
$$Offspring2_6 = (0.5 \times 0.2 + (1-0.5) \times 0.1) = 0.15$$
$$Offspring2_7 = (0.5 \times 0.7 + (1-0.5) \times 0.9) = 0.8$$

Tahap selanjutnya adalah proses mutasi menggunakan operator *uniform mutation*. Mutasi merupakan tahap perubahan nilai gen pada suatu kromosom [19]. Menggunakan *uniform mutation*, gen-gen pada kromosom didapatkan dari pembangkitan bilangan secara acak dengan distribusi seragam (*uniform distribution*). Pada tahap ini dibangkitkan bilangan acak sejumlah banyak gen pada setiap individu, ketika bilangan acak yang mewakili gen bernilai lebih kecil dari probabilitas mutasi. Maka nilai dari gen yang bersangkutan akan diganti dengan bilangan acak dengan rentang 0 sampai 1.

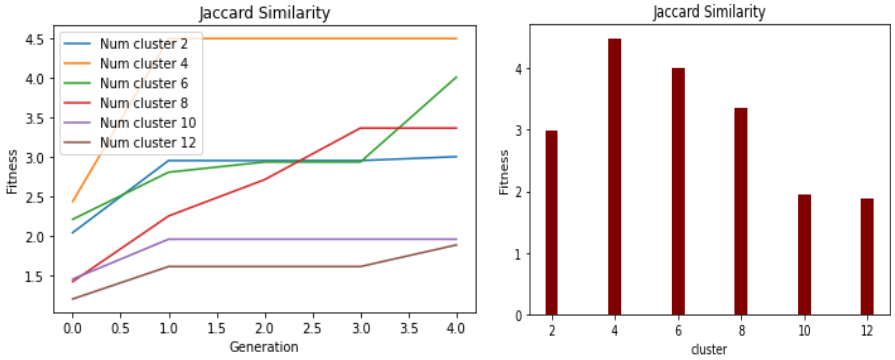
Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

Setelah proses mutasi selesai, kemudian dilanjutkan pada proses *update generation*. *Update generation* merupakan tahapan untuk menentukan individu / kromosom yang masih bertahan (*survive*) dalam populasi. Proses *update generation* pada penelitian ini menggunakan metode *Elitism*, yaitu memilih individu yang memiliki nilai *fitness* terbaik, baik dari individu *offspring* maupun dari individu *parent* [20]. Individu dengan nilai *fitness* tinggi sejumlah *population size* akan menjadi generasi selanjutnya, sedangkan individu dengan nilai *fitness* lebih rendah dari *population size* akan dibuang. Proses *evaluation*, *selection*, *crossover*, *mutation* dan *update generation* akan terus berulang hingga mencapai *stopping criteria*. Pada penelitian ini *stopping criteria* yang digunakan berdasarkan jumlah generasi. Hasil individu terbaik dari Algoritma Genetika akan digunakan untuk prediksi pada data *testing*. Pada tahap evaluasi, *dataset* dibagi menjadi data *training* dan data *testing* dengan rasio perbandingan 80:20 dari keseluruhan jumlah data. Data yang digunakan yaitu sebanyak 1K. Data *training* yang digunakan sebanyak 80% dari total jumlah data sedangkan data *testing* yang digunakan sebanyak 20% dari total jumlah data. Parameter Algoritma Genetika untuk proses evaluasi setiap metode pengukuran kemiripan adalah sama, yaitu menggunakan jumlah populasi = 5, jumlah generasi = 5, probabilitas *crossover* = 0.8, dan probabilitas mutasi = 0.1.

3. HASIL DAN PEMBAHASAN

3.1 Evaluasi Data Training

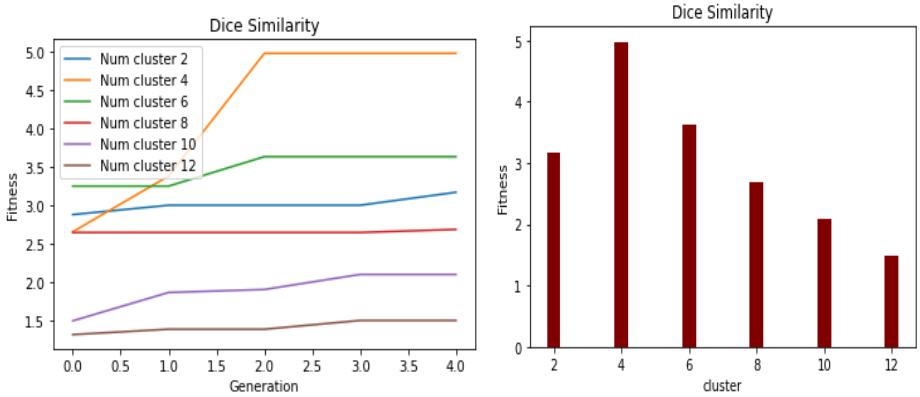
Proses evaluasi dilakukan dengan mencoba variasi jumlah *cluster* yang optimal terhadap beberapa metode pengukuran kemiripan yang digunakan, yaitu *Jaccard Similarity Coefficient*, *Sørensen–Dice Coefficient*, dan *Hamming Coefficient*. Hasil proses evaluasi pada data *training* menggunakan metode pengukuran kemiripan *Jaccard Similarity Coefficient* adalah seperti pada gambar 4 berikut.



Gambar 4. Hasil evaluasi data *training Jaccard Similarity Coefficient*

Berdasarkan gambar 4, pada percobaan variasi jumlah cluster 2, 4, 6, 8, 10, dan 12 diperoleh jumlah cluster optimal sebanyak 4 cluster untuk metode pengukuran kemiripan *Jaccard Similarity Coefficient*. Jumlah cluster sebanyak 4, mampu mendapatkan nilai *fitness* terbaik yaitu sebesar 4.490. Sedangkan jumlah cluster lebih dari 4 akan semakin menurunkan nilai *fitness*. Kemudian hasil evaluasi data *training* menggunakan pengukuran kemiripan *Jaccard Similarity Coefficient* akan dibandingkan dengan hasil evaluasi data *training* menggunakan metode pengukur kemiripan *Sørensen–Dice Coefficient*.

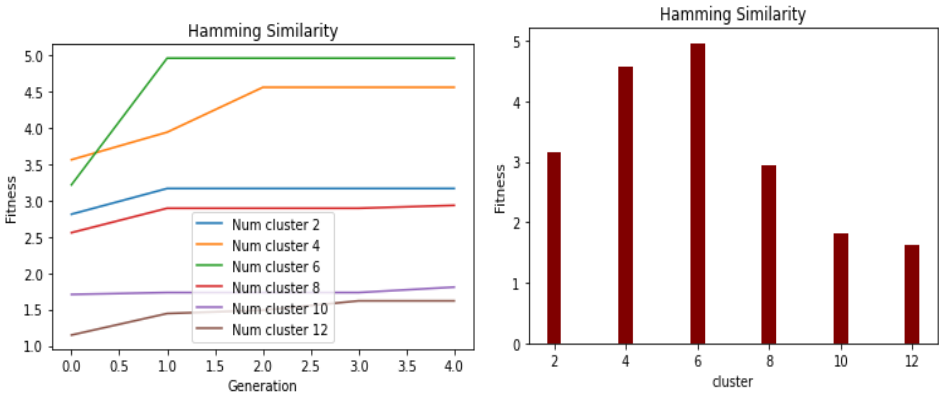
Hasil proses evaluasi pada data *training* menggunakan metode pengukuran kemiripan *Sørensen–Dice Coefficient* adalah seperti pada gambar 5 berikut.



Gambar 5. Hasil evaluasi data *training Sørensen–Dice Coefficient*

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

Berdasarkan gambar 5, pada percobaan variasi jumlah *cluster* 2, 4, 6, 8, 10, dan 12 diperoleh jumlah *cluster* optimal sebanyak 4 *cluster* untuk metode pengukuran kemiripan *Sørensen–Dice Coefficient*. Jumlah *cluster* sebanyak 4, mampu mendapatkan nilai *fitness* terbaik yaitu sebesar 4.979. Sedangkan jumlah *cluster* lebih dari 4 akan semakin menurunkan nilai *fitness*. Kemudian hasil evaluasi data *training* menggunakan pengukuran kemiripan *Jaccard Similarity Coefficient* dan *Sørensen–Dice Coefficient* akan dibandingkan dengan hasil evaluasi data *training* menggunakan metode pengukur kemiripan *Hamming Coefficient*.
Hasil proses evaluasi pada data *training* menggunakan metode pengukuran kemiripan *Hamming Coefficient* adalah seperti pada gambar 6 berikut.



Gambar 6. Hasil evaluasi data *training Hamming Coefficient*

Berdasarkan gambar 6, pada percobaan variasi jumlah cluster 2, 4, 6, 8, 10, dan 12 diperoleh jumlah *cluster* optimal sebanyak 6 *cluster* untuk metode pengukuran kemiripan *Hamming Coefficient*. Jumlah cluster sebanyak 6, mampu mendapatkan nilai *fitness* terbaik yaitu sebesar 4.964. Dari hasil pengujian terlihat kecenderungan evaluasi yang semakin meningkat seiring generasi bertambah dengan jumlah *cluster* optimal sejumlah 4 dan 6. Kemudian semakin banyak variasi jumlah *cluster* yang digunakan menyebabkan pengurangan akurasi.

3.2 Evaluasi Data Testing

Berdasarkan hasil evaluasi data *training* pada siklus Algoritma Genetika masing-masing metode pengukuran kemiripan memiliki individu yang terbaik. Individu terbaik tersebut akan digunakan untuk memprediksi nilai *rating* pada data *testing*. Hasil proses evaluasi pada data *testing* menggunakan metode pengukuran kemiripan *Jaccard Similarity Coefficient*, *Sørensen–Dice Coefficient*, dan *Hamming Coefficient* ditunjukkan pada tabel 4 berikut.

Tabel 4. Hasil Evaluasi Data *Testing*

Jaccard Similarity Coefficient	Sørensen–Dice Coefficient	Hamming Coefficient
0.6	0.59	0.61

Berdasarkan tabel 4, hasil evaluasi data *testing* pada individu terbaik yang dihasilkan Algoritma Genetika diukur menggunakan *Mean Absolute Percentage Error* (MAPE). Nilai MAPE untuk metode *Jaccard Similarity Coefficient* sebesar 0.6, metode *Sørensen–Dice Coefficient* sebesar 0.59 dan metode *Hamming Coefficient* sebesar 0.61. Rata-rata nilai MAPE ketiga metode pengukuran kemiripan adalah sebesar 0.6 atau 60%.

3.3 Pembahasan

Hasil penelitian ini menunjukkan bahwa kombinasi *K-Means* dan Algoritma Genetika efektif dalam meningkatkan kualitas pemilihan *neighborhood* pada *User-based Collaborative Filtering*. *Sørensen–Dice Coefficient* terbukti menjadi metode kemiripan yang paling stabil baik pada *training* maupun *testing*, melampaui *Jaccard* dan *Hamming* yang sebelumnya dianggap unggul pada beberapa studi. Temuan ini berbeda dengan penelitian [6] yang menyatakan bahwa *Jaccard* cukup efektif pada rekomendasi *film*, serta penelitian [7] yang menilai *Jaccard* lebih efisien secara komputasi dibandingkan *Sørensen–Dice*. Namun, kedua penelitian tersebut tidak menerapkan mekanisme klasterisasi dan optimasi, sehingga efektivitas metode kemiripan sangat bergantung pada struktur pemodelan yang digunakan. Di sisi lain, hasil penelitian ini sejalan dengan [13] dan [14] yang menunjukkan bahwa Algoritma Genetika mampu memperbaiki stabilitas centroid dan kualitas cluster pada *K-Means*. Perbedaannya, penelitian ini tidak hanya mengevaluasi kualitas cluster tetapi juga secara langsung mengukur dampaknya terhadap akurasi prediksi rating. Dengan demikian pemilihan metode kemiripan terbaik bergantung pada integrasi penuh antara perhitungan kemiripan, klasterisasi, dan optimasi.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

4. KESIMPULAN

Dari penelitian yang dilakukan dapat disimpulkan bahwa sistem rekomendasi berbasis *user-based collaborative filtering* yang digabungkan dengan proses pengelompokan menggunakan algoritma *K-Means* dan optimasi titik pusat (*centroid*) menggunakan Algoritma Genetika mampu meningkatkan kualitas pemilihan *neighborhood*. Tiga metode pengukuran kemiripan yang digunakan, yaitu *Jaccard Similarity Coefficient*, *Sørensen–Dice Coefficient*, dan *Hamming* Coefficient menunjukkan performa yang berbeda pada proses identifikasi kesesuaian *neighborhood*. Hasil optimasi menunjukkan bahwa nilai fitness terbaik diperoleh pada metode *Sørensen–Dice Coefficient* dengan nilai fitness sebesar 4,979, diikuti oleh *Hamming Coefficient* sebesar 4,964 dan *Jaccard Similarity Coefficient* sebesar 4,490. Evaluasi menggunakan data *testing* menunjukkan bahwa metode *Sørensen–Dice Coefficient* menghasilkan nilai MAPE terendah, yaitu 0,59 (59%), dibandingkan *Jaccard* sebesar 0,60 dan *Hamming* sebesar 0,61. Dengan demikian, *Sørensen–Dice Coefficient* dapat diidentifikasi sebagai metode pengukuran kemiripan yang paling efektif dalam penelitian ini karena menghasilkan kesalahan prediksi paling rendah sekaligus nilai *fitness* tertinggi. Pada penelitian ini parameter Algoritma Genetika yang digunakan meliputi jumlah populasi = 5, jumlah generasi = 5, probabilitas *crossover* = 0,8, dan probabilitas *mutation* = 0,1 dengan ukuran *dataset* 1K. Rata-rata waktu komputasi untuk tiap metode berada pada kisaran ±20 menit. Rata-rata nilai MAPE keseluruhan sebesar 60% menunjukkan bahwa akurasi prediksi masih dapat ditingkatkan. Kekurangan ini membuka peluang untuk pengembangan model pada penelitian berikutnya, misalnya dengan peningkatan ukuran populasi, jumlah generasi, metode optimasi tambahan, ataupun eksplorasi fungsi *fitness* yang lebih adaptif.

REFERENCES

[1] A. N. Khusna, K. P. Delasano, and D. C. E. Saputra, "Penerapan User-Based Collaborative Filtering Algorithm," *MATRIK : Jurnal Manajemen, Teknik Informatika dan Rekayasa Komputer*, vol. 20, no. 2, pp. 293–304, May 2021, doi: 10.30812/matrik.v20i2.1124.

[2] F. Ramadhan and A. Musdholifah, "Online Learning Video Recommendation System Based on Course and Sylabus Using Content-Based Filtering," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 3, p. 265, Jul. 2021, doi: 10.22146/ijccs.65623.

[3] C. Wibisono, L. S. Haryadi, J. E. Widyaya, and S. L. Liliawati, "Sistem Rekomendasi Suku Cadang Berdasarkan Item Based Filtering," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr. 2021, doi: 10.28932/jutisi.v7i1.3036.

[4] H. S. Kusuma and A. Musdholifah, "Recommendation System for Thesis Topics Using Content-based Filtering," *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, vol. 15, no. 1, p. 65, Jan. 2021, doi: 10.22146/ijccs.62716.

[5] I. W. Jepriana and R. Wardoyo, "Algoritme Genetika untuk Mengurangi Galat Prediksi Metode Item-based Collaborative Filtering," *Journal of Mathematics and Natural Sciences (BIMIPA)*, vol. 25, no. 2, 2018.

[6] Hadi, Ichwanto, L. W. Santoso, and A. N. Tjondrowiguno, "Sistem Rekomendasi Film menggunakan User-based Collaborative Filtering dan K-modes Clustering," *Infra*, vol. 8, no. 1, pp. 228–234, 2020.

[7] Masykur and M. Iqbal Rofikurrahman, "IMPLEMENTASI METODE DICE SIMILARITY DALAM PERANCANGAN SISTEM REKOMENDASI ARTIKEL BERITA," in *Seminar Informatika Aplikatif Polinema*, 2020, pp. 532–536.

[8] I. Ishak, A. Yahya, and Y. Yusri, "Book Recommendation System Based on Collaborative Filtering: User-Based, Item-Based, and Singular Value Decomposition Analysis," *Journal of System and Computer Engineering (JSCE)*, vol. 6, no. 4, pp. 329–342, Oct. 2025, doi: 10.61628/jsce.v6i4.2201.

[9] B. D. Okkalioglu, "A Novel Hybrid Item-Based Similarity Method to Mitigate the Effects of Data Sparsity in Multi-Criteria Collaborative Filtering," *IEEE Access*, vol. 13, pp. 64660–64686, 2025, doi: 10.1109/ACCESS.2025.3559398.

[10] Y. Zhu, J. Zhang, Y. Bai, W. Wang, and D. Ma, "Research on movie recommendation algorithm based on K-means++ collaborative filtering," *IET Conference Proceedings*, vol. 2025, no. 20, pp. 301–304, Sep. 2025, doi: 10.1049/icp.2025.2767.

[11] H. J. Suryanto, A. R. C, and Y. Lukito, "Indoor Positioning System dengan Algoritma K-Means dan KNN," *Jurnal Teknik Informatika dan Sistem Informasi (JuTISI)*, vol. 2, no. 3, 2016.

[12] R. Pormes and D. H. F. Manongga, "Penggunaan Algoritma Clustering K-means Untuk Melihat Daerah-Daerah Penyuplai Mahasiswa Di Universitas Kristen Satya Wacana, Salatiga.," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 5, no. 3, Jan. 2020, doi: 10.28932/jutisi.v5i3.1968.

[13] Taslim, D. Toresa, D. Jollyta, D. Suryani, and E. Sabna, "Optimasi K-Means dengan Algoritma Genetika untuk Target Pemanfaat Air Bersih Provinsi Riau," *Indonesian Journal of Computer Science*, vol. 10, no. 1, Jul. 2022, doi: 10.33022/ijcs.v10i1.3064.

[14] Y. Istianto and S. 'Uyun, "Klasifikasi Kebutuhan Jumlah Produk Makanan Customer Menggunakan K-Means Clustering dengan Optimasi Pusat Awal Cluster Algoritma Genetika," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 8, no. 5, p. 861, Oct. 2021, doi: 10.25126/jtiik.2021842990.

[15] M. Abdi, G. Okeyo, and R. Mwangi, "Improved Collaborative Filtering Recommender System Based on Hybrid Similarity Measures," *The International Arab Journal of Information Technology*, vol. 22, no. 1, Jan. 2025, doi: 10.34028/iajit/22/1/8.

Jurnal Sistem Komputer dan Informatika (JSON)
Volume 7, Nomor 2, Desember 2025
e-ISSN 2685-998X
DOI 10.30865/json.v7i2.9382

[16] I. Syafii, C. Edi Widodo, and J. Endro Suseno, "Expert System Diagnose Broiler Chicken Diseases using Case Based Reasoning Method and Sorensen Dice Coefficient," *E3S Web of Conferences*, vol. 448, p. 02067, Nov. 2023, doi: 10.1051/e3sconf/202344802067.
[17] F. Yuniardini and T. Widiyaningtyas, "Analisis Perbandingan Pearson Correlation dan Cosine Similarity pada Rekomendasi Musik berbasis Collaborative Filtering," *Edumatic: Jurnal Pendidikan Informatika*, vol. 8, no. 2, pp. 555–564, Dec. 2024, doi: 10.29408/edumatic.v8i2.27781.
[18] H. F. Mustika and A. Musdholifah, "Book Recommender System Using Genetic Algorithm and Association Rule Mining," *Computer Engineering and Applications Journal*, vol. 8, no. 2, pp. 85–92, Jun. 2019, doi: 10.18495/comengapp.v8i2.305.
[19] S. H. Novianti, E. C. Djamal, and A. Komarudin, "Optimalisasi Distribusi Harga Tiket Pesawat berdasarkan Kepadatan Rute Menggunakan Algoritma Genetika," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 5, no. 2, Sep. 2019, doi: 10.28932/jutisi.v5i2.1756.
[20] C. Rozikin and A. Solichin, "Implementasi Algoritma Genetika dan Regresi Linier Berganda Untuk Prediksi Persediaan Bahan Makanan Pada Restoran Cepat Saji," in *Prosiding Seminar Nasional Multidisiplin Ilmu*, 2017.