

# Rekomendasi Buku Menggunakan K-Nearest Neighbor (KNN) dan Binary Particle Swarm Optimization (BPSO)

Hernawati Gohzali<sup>1</sup>, Sunario Megawan<sup>2</sup>, Erwin<sup>3</sup>, Jeffrey Onggo<sup>4</sup>

Program Studi Teknik Informatika, STMIK Mikroskil

<sup>1</sup>hernawati.gohzali@mikroskil.ac.id, <sup>2</sup>sunario@mikroskil.ac.id, <sup>3</sup>erwinhuank@gmail.com,

<sup>4</sup>Onggo.Jeffrey@gmail.com

## Abstrak

*Sebagian besar sistem rekomendasi yang berdasarkan konten dapat membantu dalam menemukan buku yang cocok untuk pembaca tetapi hanya mempertimbangkan buku yang disukai tanpa mempertimbangkan yang tidak disukai oleh pembaca. Untuk menyelesaikan permasalahan tersebut, maka rekomendasi berdasarkan konten buku yang disukai dan tidak disukai perlu dilakukan agar dapat memberikan hasil rekomendasi yang lebih baik kepada pembaca. Dalam penelitian ini, akan diterapkan Binary Particle Swarm Optimization (BPSO) untuk memilih fitur dari buku yang disukai oleh pembaca dan K-Nearest Neighbor (KNN) digunakan untuk mengklasifikasikan data buku yang memiliki jarak terdekat ke buku yang disukai dan tidak disukai oleh pembaca. Pengujian hasil akurasi rekomendasi dilakukan dengan cara membandingkan hasil rekomendasi antara data buku yang tidak menerapkan pemilihan fitur dengan data buku yang menggunakan pemilihan fitur dengan tujuan untuk menguji pengaruh penerapan pemilihan fitur pada rekomendasi buku. Hasil pengujian akurasi rekomendasi buku dengan pemilihan fitur memberikan rekomendasi yang lebih baik bagi pembaca daripada rekomendasi tanpa pemilihan fitur.*

**Kata Kunci :** *content based recommender system, text mining, feature selection, Binary Particle Swarm Optimization (BPSO), K-Nearest Neighbor (KNN)*

## Abstract

*Most of recommender systems are based on content can be helpful to find recommendation book suitable for reader but it only consider about a liked book by user without considered about the disliked one. For solving the problem, a recommendation based on content of liked and dislike book by user is done. In this research, we applied Binary Particle Swarm Optimization(BPSO) to select feature from book that the reader like and K-Nearest Neighbor(KNN) are use for classify book data which had the closest distance to the book that the reader liked and disliked. Testing the accuracy of the recommendations is done by comparing the results of recommendations in the book data that do not apply feature selection with book data that applies selective features to test the effect of application of feature selection on book recommendations. The result of book recommendation accuracy testing with feature selection give a better recommendation for user than recommendation without feature selection.*

**Keyword :** *content based recommender system, text mining, feature selection, Binary Particle Swarm Optimization (BPSO), K-Nearest Neighbor (KNN)*

## 1. PENDAHULUAN

Sistem rekomendasi merupakan salah satu fitur yang tersedia pada perangkat lunak dan sangat bermanfaat dalam memudahkan pengguna. Sistem rekomendasi sangat diperlukan karena terlalu banyak jenis dan jumlah data yang tersedia. Dengan adanya sistem rekomendasi, pengguna akan dimanjakan dengan rekomendasi – rekomendasi yang sesuai dengan preferensi masing – masing pengguna, sehingga pengguna tidak perlu mengalami kesulitan untuk menemukan item yang diinginkan. Pada sistem rekomendasi, kegiatan memproses data, memilih fitur data yang sesuai, dan metode klasifikasi selalu menjadi tantangan terhadap kinerja dari sistem rekomendasi [1].

Sistem rekomendasi buku akan memberikan target rekomendasi buku yang dipersonalisasi dengan menggunakan data yang diperoleh dari riwayat pembaca dalam memprediksi minat dan hobi potensial pembaca [2]. Sikap pembaca terhadap buku dapat mengekspresikan pendapatnya sendiri seperti suka, tidak suka, biasa, yang dalam hal ini biasanya dapat merujuk pada perasaan positif, perasaan negatif ataupun perasaan netral [3]. Sistem rekomendasi buku berdasarkan konten yang digunakan saat ini hanya mempertimbangkan buku yang disukai pembaca tanpa mempertimbangkan buku yang tidak disukai oleh pembaca. Salah satu parameter yang dapat digunakan untuk menentukan buku mana yang disukai atau tidak disukai adalah melalui peringkat yang diberikan oleh pembaca.

Dalam penelitian ini, dilakukan rekomendasi berbasis konten terhadap buku yang disukai dan tidak disukai oleh pembaca. Metode berbasis konten ini memberikan rekomendasi berdasarkan klasifikasi item yang disukai dan tidak disukai oleh pembaca di masa lalu. Adapun cara kerja dari metode ini adalah dengan menghitung kesamaan item yang disukai pembaca dengan item lain berdasarkan fitur yang ada [4]. Pemilihan fitur memiliki pengaruh besar pada klasifikasi. Pemilihan fitur yang relevan dan menghapus fitur yang tidak relevan dapat meningkatkan efisiensi dalam komputasi dan akurasi klasifikasi [5]. Algoritma Binary Particle Swarm Optimization (BPSO) digunakan untuk memilih fitur dan meningkatkan akurasi klasifikasi [6] dan metode klasifikasi yang digunakan adalah K-Nearest Neighbor (KNN).

## 2. METODE PENELITIAN

### 2.1 Kajian Pustaka

#### 2.1.1 Sistem Rekomendasi

Sistem rekomendasi dapat didefinisikan sebagai program yang mencoba untuk merekomendasikan item yang paling cocok (produk atau jasa) untuk pengguna tertentu (individu atau bisnis) dengan memprediksi minat pengguna pada sebuah item berdasarkan informasi terkait tentang item, pengguna dan interaksi antara item dan pengguna. Tujuan dari pengembangan sistem rekomendasi adalah untuk mengurangi informasi yang berlebihan dengan mengambil informasi dan layanan yang paling relevan dari sejumlah besar data, sehingga memberikan layanan pribadi. Fitur yang paling penting dari sebuah sistem rekomendasi adalah kemampuannya untuk “menebak” preferensi dan kepentingan pengguna dengan menganalisis perilaku pengguna dan / atau perilaku pengguna lain untuk menghasilkan rekomendasi pribadi [7].

Metode rekomendasi yang dapat dilakukan diantaranya mencakup collaborative filtering, content-based, knowledge-based, dan hybrid. Prinsip-prinsip dasar sistem rekomendasi content-based adalah: 1) Untuk menganalisis deskripsi item yang disukai oleh pengguna tertentu untuk menentukan atribut umum utama (preferensi) yang dapat digunakan untuk membedakan item-item ini. Preferensi ini disimpan dalam profil pengguna. 2) Untuk

membandingkan atribut setiap item dengan profil pengguna sehingga hanya item yang memiliki tingkat kesamaan yang tinggi dengan profil pengguna yang akan direkomendasikan. Dalam sistem rekomendasi content-based, dua teknik telah digunakan untuk menghasilkan rekomendasi. Salah satu teknik menghasilkan rekomendasi secara heuristik menggunakan metode pencarian informasi tradisional, seperti ukuran kesamaan cosinus. Teknik lain menghasilkan rekomendasi menggunakan pembelajaran statistik dan metode pembelajaran mesin, sebagian besar membangun model yang mampu mempelajari minat pengguna dari data historis (data pelatihan) pengguna [7].

### 2.1.2 Text Preprocessing

*Text preprocessing* merupakan suatu proses perubahan bentuk data tekstual yang belum terstruktur menjadi data yang terstruktur. Tahapan pada *text preprocessing* adalah sebagai berikut [8]:

1. *Case folding*  
*Case folding* bertujuan untuk menghilangkan semua karakter selain huruf didalam data dan mengubah semua huruf menjadi huruf kecil (*lower case*).
2. *Tokenization*  
*Tokenization* merupakan proses menghilangkan angka-angka dan tanda baca ( , . ? ! ) ( - \* \$ # ; : % ^ @ / & + = ' \n \t) dan pemotongan strip input berdasarkan tiap kata yang menyusunnya.
3. *Stemming*  
*Stemming* merupakan proses pemotongan imbuhan atau pengembalian kata berimbuhan menjadi kata dasar. Algoritma *stemming* yang digunakan dalam penelitian ini adalah algoritma *porter*.
4. *Filtering* atau *stopwords removal*  
*Stopwords removal* merupakan penghapusan terhadap kata-kata umum yang tidak berguna dalam klasifikasi. Daftar kata-kata umum yang digunakan pada tahap ini diambil dari sumber <https://www.ranks.nl/stopwords>.

### 2.1.3 Vector Space Model

*Vector Space Model* (VSM) merupakan metode yang digunakan untuk merepresentasikan data dalam vektor multidimensi[9]. Dalam *Vector Space Model*, koleksi dokumen direpresentasikan sebagai sebuah matriks term document (atau matriks *term frequency*). Setiap sel dalam matriks bersesuaian dengan bobot yang diberikan dari suatu *term* dalam dokumen yang ditentukan. Nilai nol berarti bahwa term tersebut tidak ada dalam dokumen.

*Term Frequency* (TF) adalah pembobotan yang sederhana dimana penting tidaknya sebuah kata dianggap sama atau sebanding dengan jumlah kemunculan kata tersebut dalam dokumen. Fungsi untuk menghitung nilai *Term Frequency* (TF) adalah sebagai berikut :

$$tf = tf_{ij} \quad (1)$$

Dimana :

$tf_{ij}$  = banyaknya kemunculan *term*  $t_i$  dalam dokumen  $d_j$ .

Pembobotan *Inverse Document Frequency* (IDF) menganggap bahwa bobot sebuah kata akan semakin besar jika kata tersebut semakin sering muncul dalam satu dokumen, tidak dalam

banyak dokumen. Rumus perhitungan *Inverse Document Frequency* (IDF) didefinisikan sebagai berikut :

$$idf_i = \log \frac{N}{df_i} \quad (2)$$

dimana :

$N$  = jumlah total dokumen yang ada dalam sistem.

$df_i$  = frekuensi dokumen dari *term* i (jumlah dokumen yang mengandung term i).

$\log$  = digunakan untuk memperkecil pengaruh relative untuk  $tf_{ij}$ .

Bobot dokumen dihitung dengan perhitungan *Term Frequency - Inverse Document Frequency* (TF-IDF) yang merupakan kombinasi antara *Term Frequency* (TF) dan *Inverse Document Frequency* (IDF). Rumus perhitungan TF-IDF adalah sebagai berikut:

$$w_{ij} = tf_i \cdot \log \left( \frac{N}{df_i} \right) \quad (3)$$

#### 2.1.4 Binary Particle Swarm Optimization

Pada *Binary Particle Swarm Optimization* (BPSO), partikel bergerak dengan membalik berbagai jumlah bit. Dengan demikian, kecepatan partikel dapat dijelaskan dengan jumlah bit-bit yang berubah di setiap iterasi. Dalam penelitian ini BPSO digunakan untuk pemilihan / seleksi fitur. Posisi setiap partikel direpresentasikan dalam bentuk biner dengan  $X_p = \{X_{p1}, X_{p2}, \dots, X_{pd}\}$  yang ditentukan secara random. Nilai bit 0 dan 1 merepresentasikan fitur yang tidak terpilih dan terpilih. Kecepatan setiap partikel direpresentasikan dengan  $V_p = \{V_{p1}, V_{p2}, \dots, V_{pd}\}$ , dimana p adalah partikel dan d adalah dimensi (fitur) dari kumpulan data yang diberikan. Kecepatan awal partikel adalah nilai acak pada kisaran 0 hingga 1. Kecepatan setiap partikel diperbaharui sesuai dengan persamaan berikut [10] :

$$V_{pd}^{new} = w \times V_{pd}^{old} + C_1 \times rand1 \times (pbest_{pd} - X_{pd}^{old}) + C_2 \times rand2 \times (gbest_{pd} - X_{pd}^{old}) \quad (4)$$

dimana :

$w$  = *inertia weight*

$C_1, C_2$  = parameter percepatan

$rand1, rand2$  = nilai acak diantara 0 hingga 1

$V_{pd}^{new}$  = nilai kecepatan partikel yang sudah diperbaharui

$V_{pd}^{old}$  = nilai kecepatan partikel sebelum diperbaharui

$X_{pd}^{new}$  = nilai posisi partikel yang sudah diperbaharui

$X_{pd}^{old}$  = nilai posisi partikel saat ini

Kecepatan partikel dari masing-masing dimensi dicoba ke kecepatan maksimum  $V_{max}$ . Jika jumlah percepatan menyebabkan kecepatan partikel itu melebihi  $V_{max}$ , maka kecepatan partikel tersebut dibatasi pada  $V_{max}$ .  $V_{max}$  dan  $V_{min}$  adalah parameter yang ditentukan pengguna, yang dalam penelitian ini digunakan  $V_{max} = 6$  dan  $V_{min} = -6$ . Adapun persamaannya dapat dilihat pada persamaan berikut [10] :

$$if V_{pd}^{new} \notin (V_{min}, V_{max}) \text{ then} \\ V_{pd}^{new} = \max(\min(V_{max}, V_{pd}^{new}), V_{min}) \quad (5)$$

Fitur yang sudah diperbaharui dihitung dengan fungsi  $S(V_{pd}^{new})$  seperti terlihat pada persamaan 6 berikut ini. Pada persamaan 7 dijelaskan bahwa jika nilai  $S(V_{pd}^{new})$  lebih besar dari nilai yang ditentukan secara acak (0 – 1) maka nilai posisi  $X_{pd}^{new}$  direpresentasikan sebagai 1, yang berarti fitur dipilih sebagai fitur yang diperlukan pada pembaharuan selanjutnya. Jika nilai  $S(V_{pd}^{new})$  lebih kecil dari nilai yang ditentukan secara acak (0 – 1) maka nilai posisi  $X_{pd}^{new}$  direpresentasikan sebagai 0, yang berarti fitur tidak dipilih sebagai fitur yang diperlukan pada pembaharuan selanjutnya [10].

$$S(V_{pd}^{new}) = \frac{1}{1+e^{-V_{pd}^{new}}} \quad (6)$$

$$\text{if } (rand < S(V_{pd}^{new})) \text{ then } X_{pd}^{new} = 1; \text{ else } X_{pd}^{new} = 0 \quad (7)$$

Proses *binary particle swarm optimization* pada seleksi fitur akan dimulai dengan menentukan jumlah partikel yang akan di tempatkan secara acak pada ruang pencarian, yaitu setiap partikel dimulai dengan satu posisi acak. Sebagai alternatif, jumlah partikel yang ditempatkan pada ruang pencarian dapat diatur sama dengan jumlah fitur dalam data, Setiap partikel mulai untuk mencari solusi dengan pergerakan. Dari posisi awal, partikel bergerak melewati solusi-solusi pada setiap iterasi [11]. Setiap *subset* yang terpilih akan dikumpulkan dan dievaluasi dengan menggunakan fungsi *fitness* (persamaan 8) [12].

$$\text{Fitness} = \alpha * \gamma(F_i(t)) + \beta * \frac{|N| - |F|}{|N|} \quad (8)$$

dimana :

$F(t)$  = subset fitur yang ditemukan oleh partikel i pada iterasi t

$\gamma(F_i(t))$  = kualitas klasifikasi dari fitur yang terpilih.

$\alpha$  dan  $\beta$  = parameter yang berhubungan dengan kualitas klasifikasi dan length subset fitur, nilai  $\alpha$  berada diantara 0 dan 1 dan nilai  $\beta = 1 - \alpha$ .

$|N|$  = total jumlah fitur.

$|F|$  = length subset fitur yang dipilih.

Jika *subset* terbaik telah ditemukan atau algoritma telah berjalan sesuai iterasi yang ditentukan, maka proses berhenti dan kembali ke subset terbaik yang ditemui. Jika kondisi pemberhentian tidak terpenuhi maka kecepatan akan diperbaharui menggunakan persamaan 4 dan 5, posisi partikel akan diperbaharui menggunakan persamaan 6 dan 7, dan prosesnya akan dilakukan perulangan lagi [11].

### 2.1.5 K-Nearest Neighbor

Algoritma *K-Nearest Neighbor* (KNN) secara luas diterapkan dalam pengenalan pola dan penggalian data (data mining) untuk klasifikasi. Algoritma ini terkenal dengan kesederhanaan dan tingkat kesalahan yang rendah. Prinsip algoritma KNN adalah jika sebagian besar sampel k paling mirip dengan titik *query*  $q_i$  dalam ruang fitur masuk dalam kategori tertentu, maka keputusan yang dapat dibuat adalah titik *query*  $q_i$  masuk dalam kategori ini. Kesamaan dapat diukur dengan jarak di ruang fitur, sehingga algoritma ini disebut algoritma *K Nearest Neighbor*. Dataset pelatihan dengan label klasifikasi yang akurat harus diketahui pada permulaan algoritma. Kemudian untuk *query* data  $q_i$ , yang labelnya tidak diketahui dan disajikan oleh vektor di ruang fitur, akan dihitung jaraknya dengan setiap data dalam dataset pelatihan. Setelah diperoleh hasil kalkulasi jarak, maka keputusan label kelas dari *query*  $q_i$  dapat dibuat sesuai dengan label k terdekat dalam dataset pelatihan [13].

### 2.1.6 Cosine Similarity

*Cosine similarity* adalah sebuah metode pengukuran kesamaan antar teks atau dokumen yang berbeda dengan mengukur sudut cosinus antar dokumen yang direpresentasikan dengan vektor. Nilai cosinus berada diantara 0 dan 1. Semakin tinggi nilai cosinus maka semakin mirip teks atau dokumen yang diukur. Nilai cosinus 1 menyatakan kesamaan 100%, sedangkan nilai cosinus 0 berarti 100% tidak sama [14]. Perhitungan *cosine similarity* menggunakan persamaan berikut [15]:

$$\text{Cosine}(q, d) = \frac{q \cdot d}{\|q\| \|d\|} \quad (9)$$

Bila dipaparkan lebih lanjut bisa dirumuskan sebagai berikut [18] :

$$\text{Cosine}(q, d) = \frac{\sum_{k=1}^m w_{qk} \cdot w_{dk}}{\sqrt{\sum_{k=1}^m (w_{qk})^2} \cdot \sqrt{\sum_{k=1}^m (w_{dk})^2}} \quad (10)$$

Dimana :

q = vektor dokumen q.

d = vektor dokumen q.

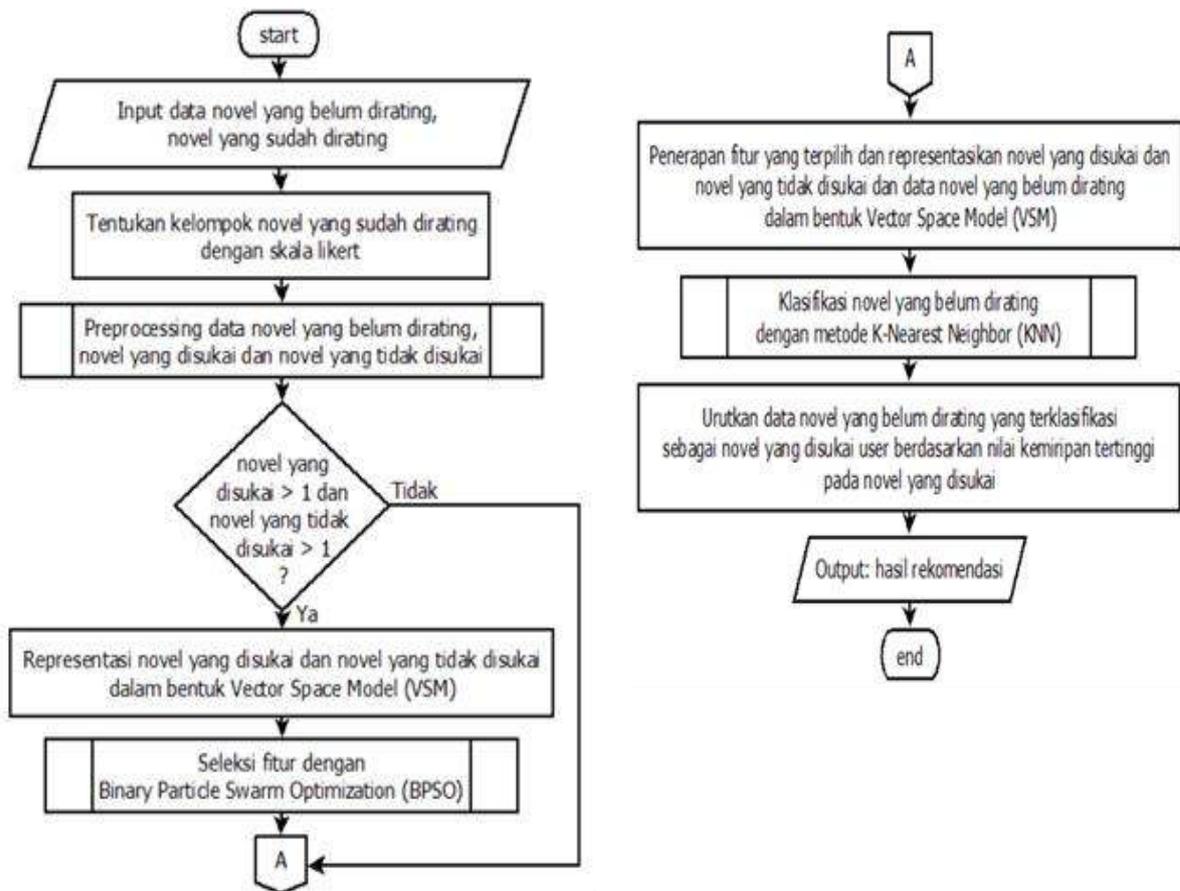
m = dimensi vektor q dan d.

$w_{qk}$  = bobot term k terhadap dokumen q.

$w_{dk}$  = bobot term k terhadap dokumen d.

### 2.2 Analisis Proses

Pada penelitian ini, sample data yang digunakan berupa buku dengan kategori novel berbahasa Inggris. Bagian ini akan menjelaskan tahapan proses kerja dari sistem rekomendasi buku yang dapat dilihat pada diagram alir Gambar 1 berikut .



Gambar 1. Diagram Alir Sistem Rekomendasi Buku

Setelah data novel yang belum di-rating, novel yang disukai dan novel yang tidak disukai di-*input*, maka dilakukan pemrosesan teks (*text preprocessing*) terhadap sinopsis setiap novel. Selanjutnya dilakukan pengecekan jumlah novel yang disukai apakah lebih besar dari 1 dan jumlah novel yang tidak disukai lebih besar 1. Jika hasil pengecekan tersebut benar maka lanjut ke tahap representasi data novel yang disukai dan novel yang tidak disukai dalam bentuk *Vector Space Model (VSM)*. Sebaliknya, jika tidak benar maka dilanjutkan ke tahap representasi data novel yang disukai dan novel yang tidak disukai dan data novel yang belum di-rating dalam bentuk *Vector Space Model (VSM)*. Setelah representasi dengan VSM dilakukan, maka akan dilanjutkan dengan seleksi fitur (*feature selection*) pada data novel yang disukai dan novel yang tidak disukai dengan BPSO. Selesai melakukan seleksi fitur, maka fitur yang terpilih akan diterapkan pada data novel yang belum di-rating, novel yang disukai dan novel yang tidak disukai. Berikutnya dilakukan klasifikasi data novel yang belum di-rating dengan metode *K-Nearest Neighbor (KNN)*. Urutkan data novel yang belum di-rating yang terklasifikasi sebagai novel yang disukai *user* berdasarkan nilai kemiripan tertinggi pada novel yang disukai, maka data novel yang belum di-rating dan terklasifikasi sebagai novel disukai selanjutnya akan direkomendasikan kepada pengguna.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Hasil

Berikut ini merupakan hasil tampilan antarmuka dari sistem rekomendasi buku yang telah berhasil diselesaikan.



Gambar 2. Tampilan Antarmuka Data Novel

Gambar 2 merupakan tampilan antarmuka yang dipergunakan untuk menambahkan, mengedit ataupun menghapus data novel dalam sistem rekomendasi.



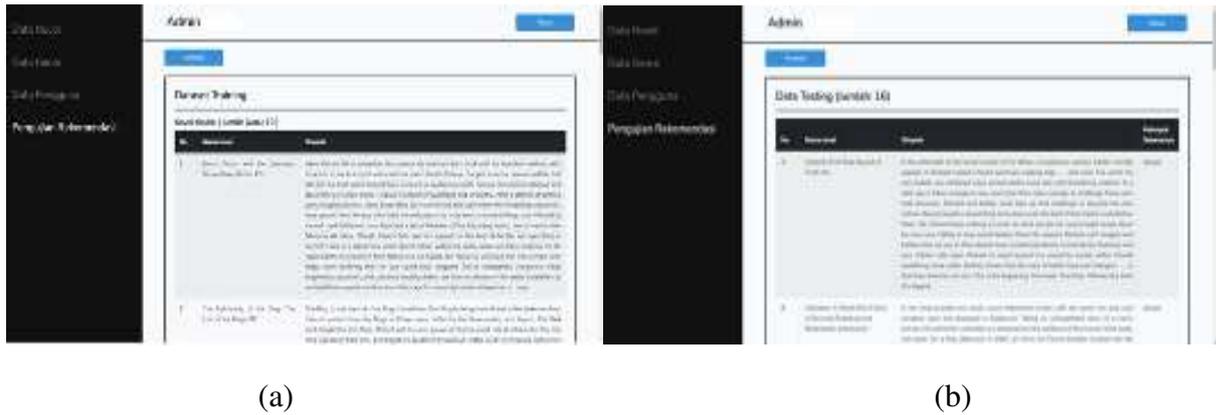
Gambar 3. Tampilan Antarmuka Genre Novel

Gambar 3 merupakan tampilan antarmuka yang dipergunakan untuk menambahkan, mengedit ataupun menghapus data genre novel dalam sistem rekomendasi.



Gambar 4. Tampilan Antarmuka Data Pengguna

Gambar 4 merupakan tampilan antarmuka yang dipergunakan oleh admin untuk mengelola data pengguna sistem. Melalui tampilan antarmuka tersebut admin dapat mengaktifkan maupun menonaktifkan akun pengguna sistem.



Gambar 5. (a) Tampilan Antarmuka DataTraining (b) Tampilan Antarmuka DataTesting

Gambar 5 (a) merupakan tampilan antarmuka untuk menampilkan data training yang digunakan dalam pengujian akurasi rekomendasi. Sementara Gambar 5 (b) merupakan tampilan antarmuka menampilkan data testing yang digunakan dalam pengujian akurasi rekomendasi.



Gambar 6. (a) Tampilan Antarmuka Rekomendasi KNN (b) Tampilan Antarmuka Rekomendasi BPSO – KNN

Gambar 6 (a) merupakan tampilan antarmuka yang menyajikan hasil rekomendasi buku novel dengan hanya menggunakan algoritma KNN untuk melakukan klasifikasi (tanpa seleksi fitur) yang digunakan dalam pengujian akurasi rekomendasi. Sementara Gambar 6 (b) merupakan tampilan antarmuka yang menyajikan hasil rekomendasi buku novel dengan menggunakan algoritma BPSO dalam melakukan seleksi fitur dan algoritma KNN untuk melakukan klasifikasi yang digunakan dalam pengujian akurasi rekomendasi.

### 3.2 Pembahasan

Pengujian terhadap akurasi sistem rekomendasi dilakukan dengan membandingkan akurasi yang dihasilkan dari rekomendasi buku novel menggunakan seleksi fitur dan akurasi yang dihasilkan dari rekomendasi buku novel tanpa menggunakan seleksi fitur. Mekanisme dalam menentukan nilai akurasi rekomendasi buku adalah sebagai berikut :

1. Tentukan data training dan data testing. Data training adalah kelompok buku novel yang disukai atau tidak disukai. Data buku novel yang termasuk dalam kelompok disukai didapatkan dari buku novel yang memiliki rating 4 ataupun 5, sedangkan data buku novel yang termasuk dalam kelompok tidak disukai didapatkan dari buku novel dengan rating 1 atau 2. Sementara data buku novel dengan rating 3 tidak dapat dimasukkan dalam kelompok disukai maupun dalam kelompok tidak disukai. Data testing yang dipergunakan merupakan gabungan dari buku novel yang disukai dan tidak disukai.

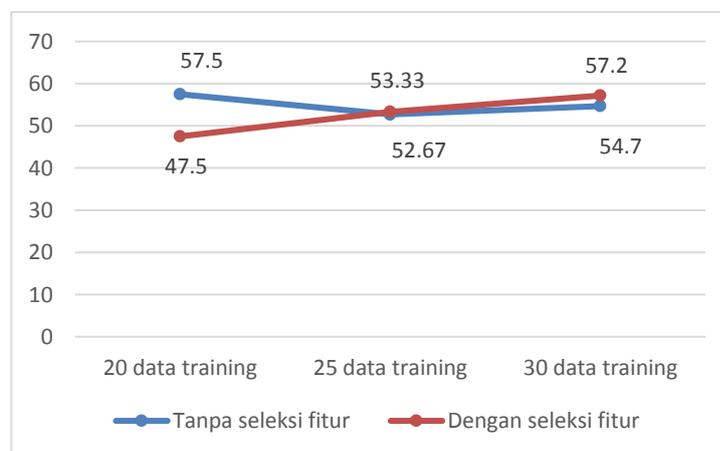
2. Klasifikasikan setiap data testing. Klasifikasikan setiap data testing ke dalam kelompok data buku novel disukai atau tidak disukai dengan menggunakan metode K-Nearest Neighbor (KNN).
3. Tentukan nilai akurasi. Nilai akurasi adalah persentase jumlah data testing yang diklasifikasikan dengan benar. Dalam hal ini, data testing yang diklasifikasikan dengan benar yaitu jika kelompok sebenarnya dari data testing adalah buku novel yang disukai diklasifikasikan sebagai buku novel disukai ataupun jika kelompok sebenarnya dari data testing adalah buku novel yang tidak disukai diklasifikasikan sebagai buku novel novel yang tidak disukai.

Banyaknya data yang digunakan dalam melakukan pengujian ditunjukkan pada Tabel 1 berikut ini :

Tabel 1. Banyak Data yang digunakan untuk Rekomendasi

	Data Training	Data Testing
Pengujian 1	20	80
Pengujian 2	25	75
Pengujian 3	30	70

Berdasarkan jumlah data yang digunakan untuk memperoleh rekomendasi tersebut, maka rata-rata akurasi pada pengujian akurasi rekomendasi dapat ditunjukkan pada Gambar 7 berikut ini.



Gambar 7. Hasil Pengujian Akurasi Rekomendasi Buku Novel

Hasil pengujian menunjukkan bahwa jumlah data training yang digunakan sangat berpengaruh terhadap akurasi hasil rekomendasi buku novel. Jika jumlah data training semakin banyak, maka akurasi rekomendasi dengan seleksi fitur akan semakin tinggi (bagus), akan tetapi hal ini berbanding terbalik dengan akurasi rekomendasi tanpa seleksi fitur yang semakin rendah.

Pengujian juga dilakukan untuk mengukur waktu yang dibutuhkan dalam mendapatkan rekomendasi buku novel dengan seleksi fitur. Data banyaknya buku novel yang disukai dan tidak disukai yang digunakan untuk pengujian bervariasi, hal ini terlihat pada tabel berikut ini:

Tabel 2. Waktu Eksekusi Rekomendasi

No.	Jumlah novel disukai	Jumlah novel tidak disukai	Jumlah novel yang belum di-rating	Waktu eksekusi (menit)
1	3	2	195	8.386
2	5	5	190	10.402
3	10	10	180	22.567

Berdasarkan pada Tabel 2, maka jumlah novel disukai dan jumlah novel yang tidak disukai sangat mempengaruhi waktu eksekusi yang dibutuhkan untuk memperoleh rekomendasi buku novel.

#### 4. KESIMPULAN

Kesimpulan yang dapat diperoleh setelah melakukan pengujian adalah sebagai berikut:

1. Banyaknya jumlah data training yang dipergunakan untuk memperoleh rekomendasi sangat berpengaruh terhadap akurasi hasil rekomendasi.
2. Hasil rekomendasi yang diperoleh dengan menggunakan seleksi fitur, akurasinya akan lebih baik bila dibandingkan dengan hasil rekomendasi tanpa menggunakan seleksi fitur.

Pada penelitian ini, algoritma seleksi fitur yang dipergunakan memiliki kinerja yang belum optimal dikarenakan dengan jumlah data training yang sedikit masih membutuhkan waktu proses eksekusi yang lama.

#### 5. SARAN

Penulis menyarankan agar pada penelitian selanjutnya dapat dilakukan penelitian dengan menggunakan buku – buku dengan teks berbahasa Indonesia. Selanjutnya algoritma Fast Correlation Based Filter dapat digunakan sebagai alternatif untuk menghilangkan fitur-fitur yang tidak relevan dan menyaring fitur-fitur yang redundant agar dapat meningkatkan performa algoritma klasifikasi.

#### DAFTAR PUSTAKA

- [1] Thi Thanh Sang Nguyen, "Model-Based Book Recommender Systems using Naïve Bayes enhanced with Optimal Feature Selection." In Proceedings of the 2019 8th International Conference on Software and Computer Applications (ICSCA '19), Feb 2019, Page 217-222.
- [2] Yuhe Gao and Can Huang and Mengqi Hu and Jianan Feng and Xiaoxue Yang, "Research on Book Personalized Recommendation Method Based on Collaborative Filtering Algorithm," IOP Conference Series: Earth and Environmental Science, July 2019.
- [3] H. Sun, Y. Li, Y. Zi and X. Han, "Research on ARIA of Affective Recommendation Index Algorithm Based on Topic Selection of Publications," 2018 International Conference on Big Data and Artificial Intelligence (BD AI), Beijing, 2018, pp. 62-66.

- [4] Ricci, F., Rokach, L., Shapira, B. & Kantor, P. B., "Recommender Systems Handbook," 2011, Springer.
- [5] Tang, J., Alelyani, S. & Liu, H., 2014, "Feature Selection for Classification: A Review," *Data Classification: Algorithms and Applications*, Jan 2014.
- [6] Xue, B., Zhang, M. & Browne, W. N., "New Fitness Functions in Binary Particle Swarm Optimisation for Feature Selection," *Appl. Soft Comput.* 18, May 2014, Page 261-276.
- [7] Lu, J., Wu, D. Mao, M., Wang, W., & Zhang, G., "Recommender System Application Developments: A Survey," pp. 1-30, 2015.
- [8] Rahman, A., Wiranto & Doewes, A., 2017, "Online News Classification Using Multinomial Naive Bayes," *Jurnal Ilmiah Teknologi dan Informasi*, Vol, 6, halaman 32-38.
- [9] Abdillah, A. A. & Muktyas I. B., 2013, "Implementasi Vector Space Model untuk Pencarian Dokumen," *Prosiding Seminar Nasional Matematika dan Pendidikan Matematika*, tanggal akses: 20 Juni 2018.
- [10] Chuang, L. Y., Tsai, S. W. & Yang, C. H., 2011, "Improved Binary Particle Swarm Optimization Using Catfish Effect for Feature Selection, *Expert Systems with Applications*," Vol. 38, halaman 12699-12707, tanggal akses : 15 Juni 2018..
- [11] Aghdam, M. H. & Heidari, S., 2015, "Feature Selection Using Particle Swarm Optimization in Text Categorization," *JAISCR*, Vol. 5, halaman 231-238, tanggal akses : 16 Juni 2018.
- [12] Zahran, B. M. & Kanaan, G., 2009, "Text Feature Selection Using Particle Swarm Optimization Algorithm," *World Applied Sciences Journal*, Vol. 7, halaman 69-74, tanggal akses : 16 Juni 2018.
- [13] Kuang, Q. & Zhao, L., 2009, "A Practical GPU Based KNN Algorithm," In *Proceedings of the Second Symposium International Computer Science and Computational Technology (ISCSCT '09)*, Dec 2009, pp. 151-155.
- [14] Fauzi, M. A., Utomo, D. C., Setiawan, B. D. & Pramukantoro, E. S., 2017, "Automatic Essay Scoring System Using N-Gram and Cosine Similarity for Gamification Based E-Learning," In *Proceedings of the International Conference on Advances in Image Processing*, page 151-155.
- [15] Sarno, R. & Rahutomo, F., 2008, "Penerapan Algoritma Weighted Tree Similarity Untuk Pencarian Semantik," *JUTI*, Vol. 7, halaman 39-46, tanggal akses : 20 Juni 2018.