

# **LAPORAN TUGAS BESAR DASAR KECERDASAN ARTIFICIAL**

## **Perancangan Sistem Rekomendasi Buku Menggunakan Logika Fuzzy Berdasarkan Dataset *Goodbooks***



Disusun Oleh:

- |                           |              |
|---------------------------|--------------|
| - Velisia Nihan Rahmawati | 103012300203 |
| - Nabila Putri Aulia      | 103012330531 |

**PROGRAM STUDI S1 INFORMATIKA  
TELKOM UNIVERSITY  
2024/2025**

# Daftar Isi

<b>A. Pendahuluan.....</b>	<b>3</b>
Latar Belakang.....	3
Permasalahan.....	3
<b>B. Block Diagram.....</b>	<b>5</b>
<b>C. Metode.....</b>	<b>6</b>
Paparan, statistik, dan sumber dari dataset.....	6
Pre-processing Dataset.....	7
Rancangan dua metode yang digunakan.....	9
Metode Fuzzy.....	9
Metode Matrix Factorization.....	15
<b>D. Hasil dan Analisis.....</b>	<b>19</b>
Hasil Sistem Rekomendasi Fuzzy.....	19
Hasil Perbandingan dengan Matrix Factorization.....	19
Analisis.....	20
<b>E. Kesimpulan.....</b>	<b>21</b>
<b>F. Referensi.....</b>	<b>22</b>

# A. Pendahuluan

## Latar Belakang

Di era digital, pembaca dihadapkan pada banyak pilihan bacaan yang tersedia secara daring, seperti pada platform Goodreads. Banyaknya pilihan ini membuat proses seleksi buku yang sesuai menjadi semakin kompleks dan membingungkan, terutama ketika preferensi pengguna bersifat subjektif dan tidak terstruktur (Ricci et al., 2011). Sistem rekomendasi menjadi salah satu solusi penting untuk membantu pengguna menemukan buku yang sesuai dengan minat dan kebutuhannya, dengan memanfaatkan data historis dan perilaku pengguna.

Namun, sistem rekomendasi konvensional seperti collaborative filtering atau content-based filtering sering kali tidak cukup efektif dalam menangani ketidakpastian, ambiguitas, dan subjektivitas dari preferensi pengguna (Han et al., 2012). Di sinilah pendekatan logika fuzzy menjadi relevan. Logika fuzzy, yang diperkenalkan oleh Zadeh (1965), memungkinkan sistem untuk menangani informasi yang tidak pasti atau samar dan memodelkan proses pengambilan keputusan dalam bentuk linguistik, seperti “tinggi”, “sedang”, atau “rendah”, yang lebih menyerupai cara berpikir manusia.

Dengan demikian, penerapan logika fuzzy dalam sistem rekomendasi buku diharapkan dapat meningkatkan akurasi dan interpretabilitas rekomendasi, terutama dalam konteks preferensi pengguna yang tidak selalu bisa dijelaskan secara matematis atau eksplisit.

## Permasalahan

Masalah utama yang ingin diselesaikan dalam penelitian ini adalah memprediksi atau merekomendasikan buku yang sesuai untuk pengguna berdasarkan informasi yang tersedia seperti rating, jumlah pembaca, dan jumlah rating bagus. Selain permasalahan utama tersebut, terdapat pula sejumlah isu tambahan yang mendasari penggunaan logika fuzzy dalam proses evaluasi kualitas buku dan penentuan rekomendasi, yaitu:

### 1. Kombinasi multi-kriteria yang kompleks

Sistem rekomendasi tidak hanya bergantung pada satu nilai seperti rating saja, melainkan juga faktor lain seperti *number of reviews*, *publication year*, dan *ratings count*. Fuzzy logic memungkinkan integrasi berbagai kriteria tersebut ke dalam satu kerangka penilaian yang logis dan dapat dikustomisasi sesuai skenario

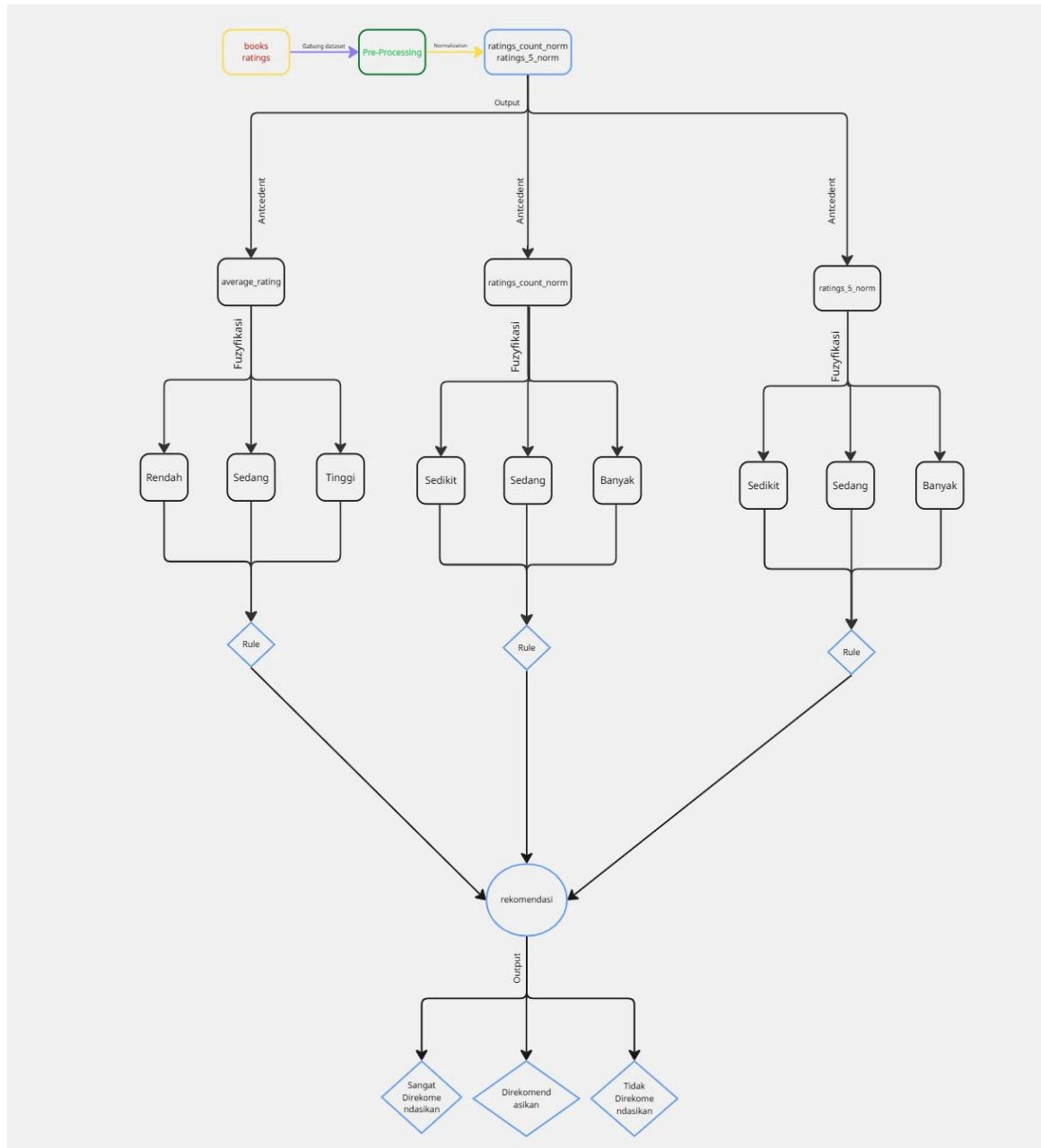
## **2. Ketidakpastian dan ambiguitas dalam preferensi pengguna**

Beberapa indikator seperti average rating, number of ratings, atau rating count tidak memiliki batasan tegas untuk menentukan apakah sebuah buku itu “bagus”, “sedang”, atau “kurang”. Sebagai contoh, rating 4.0 bisa dianggap bagus oleh sebagian pengguna, tetapi biasa saja bagi pengguna lain. Logika fuzzy membantu menangani ketidakpastian ini dengan representasi linguistik seperti “tinggi”, “sedang”, dan “rendah”.

## **3. Kebutuhan untuk mentransformasikan data numerik menjadi kategori linguistik yang lebih mudah dipahami**

Agar sistem rekomendasi dapat digunakan oleh pengguna awam atau dikembangkan lebih lanjut ke sistem berbasis antarmuka (misal chatbot atau aplikasi mobile), maka nilai-nilai numerik seperti “average rating = 4.2” perlu dikonversi menjadi istilah linguistik seperti “sangat bagus”.

## B. Block Diagram



## C. Metode

### Paparan, statistik, dan sumber dari dataset

#### a. Dataset

Dataset yang digunakan adalah Goodbooks-10K, yang tersedia di platform Kaggle. Dataset ini dikompilasi berdasarkan data publik dari Goodreads dan mencakup lebih dari 20.000 entri buku dari total sekitar 10.000 karya (works) yang paling populer. Dataset ini cocok digunakan untuk membangun dan mengevaluasi sistem rekomendasi karena memiliki berbagai atribut numerik dan kategorik yang lengkap.

Sumber dataset:

<https://www.kaggle.com/datasets/zygmunt/goodbooks-10k>

#### b. Statistik Umum Dataset

Dataset yang kami pilih dari Kaggle berisi informasi lebih dari 20.000 buku, termasuk atribut seperti:

Atribut	Deskripsi
id	ID internal unik untuk setiap entri buku di dataset.
book_id	ID buku dari sistem Goodreads. Ini adalah kunci utama yang sering digunakan untuk penggabungan data.
best_book_id	ID dari versi “terbaik” atau utama dari buku tersebut di Goodreads, sering digunakan untuk menyatukan berbagai edisi.
work_id	ID dari “work” Goodreads, yang mengelompokkan berbagai edisi dari sebuah karya buku.
books_count	Jumlah total edisi dari buku tersebut di Goodreads.
isbn	ISBN (International Standard Book Number), identifikasi edisi cetakan buku.
isbn13	Versi ISBN yang terdiri dari 13 digit.

authors	Nama penulis buku. Bisa lebih dari satu penulis, dipisah dengan koma.
original_publication_year	Tahun pertama kali buku diterbitkan.
original_title	Judul asli buku, tanpa tambahan edisi atau subtitle.
title	Judul lengkap buku (bisa termasuk subtitle, edisi, atau keterangan lainnya).
language_code	Kode bahasa dari buku
average_rating	Nilai rata-rata rating dari pengguna Goodreads.
ratings_count	Total jumlah rating yang diberikan oleh pengguna untuk buku tersebut.
work_ratings_count	Jumlah rating untuk seluruh “work” (semua edisi dari buku tersebut).
work_text_reviews_count	Jumlah ulasan berbentuk teks untuk seluruh edisi.
ratings_1, ratings_2, ratings_3, ratings_4, ratings_5	Jumlah rating bintang 1 hingga 5 yang diberikan ke buku tersebut.

## Karakteristik Data

- Tipe data campuran: numerik (average\_rating, ratings\_count, dll) dan kategorik (title, authors, dll).
- Terdapat beberapa atribut yang redundant atau jarang digunakan seperti work\_text\_reviews\_count, books\_count, dan original\_title.
- Distribusi rating cenderung bias ke nilai tinggi (4.0 ke atas), yang umum dalam sistem ulasan daring.

## Pre-processing Dataset

Dalam tahap ini, dilakukan pembersihan dan penyesuaian data mentah agar siap digunakan dalam proses penalaran fuzzy dan pembelajaran model pembandingan. Langkah-langkah pre-processing yang dilakukan adalah sebagai berikut:

### 1. Menghapus Atribut yang Tidak Diperlukan

Beberapa kolom dalam dataset tidak relevan terhadap sistem rekomendasi berbasis preferensi pengguna, sehingga dihapus untuk menyederhanakan data. Atribut-atribut yang dihapus antara lain:

- isbn dan isbn13: Nomor unik buku, tetapi tidak memiliki pengaruh langsung terhadap proses rekomendasi.
- language\_code: Hampir seluruh buku menggunakan bahasa Inggris, sehingga kurang variatif dan tidak memberi kontribusi signifikan.
- work\_ratings\_count dan work\_text\_reviews\_count: Redundan terhadap kolom ratings\_count, sehingga bisa diabaikan untuk menghindari duplikasi informasi.

## 2. Menggabungkan Dataset

Dataset *Goodbooks-10K* terdiri dari dua file utama:

- books.csv → Berisi metadata buku (judul, penulis, rating, dsb.)
- ratings.csv → Berisi rating yang diberikan oleh pengguna terhadap buku

Kedua dataset ini digabungkan menggunakan kunci book\_id untuk membentuk satu dataframe terpadu. Hasil penggabungan memungkinkan kita untuk:

- Menghubungkan data rating pengguna dengan informasi buku terkait
- Melakukan agregasi nilai rating seperti ratings\_count dan ratings\_5 per buku

## 3. Normalisasi Nilai ratings\_count dan ratings\_5

Nilai asli dari atribut ratings\_count dan ratings\_5 sangat besar dan bervariasi antar buku. Untuk memastikan konsistensi skala dan mendukung proses fuzzifikasi, dilakukan normalisasi ke rentang [0–100].

Tujuannya adalah:

- Menyamakan skala antar atribut agar bisa dipetakan ke fungsi keanggotaan fuzzy dengan tepat
- Menghindari dominasi satu atribut karena skala angka yang terlalu besar

Hasil dari proses ini memungkinkan nilai ratings\_count dan ratings\_5 diklasifikasikan ke dalam kategori linguistik fuzzy seperti sedikit, sedang, atau banyak secara konsisten.



# Rancangan dua metode yang digunakan

## Metode Fuzzy

### a. Jumlah nilai lingustik untuk setiap atribut

Sistem rekomendasi ini menggunakan tiga atribut utama dari dataset Goodbooks-10K, yaitu:

1. `average_rating` – Rata-rata penilaian terhadap buku
2. `ratings_count` – Jumlah total penilaian dari semua bintang
3. `ratings_5` – Jumlah penilaian dengan bintang 5 (paling tinggi)

Masing-masing atribut diberi representasi linguistik (label bahasa alami) untuk memodelkan ketidakpastian dalam preferensi pengguna.

Atribut	Nilai Linguistik	Batas Nilai
Average_rating	3 (rendah, sedang, tinggi)	[1.0, 1.0, 3.5], [1.0, 3.5, 5.0], [3.5, 5.0, 5.0]
Rating_Count	3 (sedikit, sedang, banyak)	[0, 0, 30], [0, 30, 80], [30, 100, 100]
Rating_5	3 (Sedikit, Sedang, Banyak)	[0, 0, 40], [0, 40, 70], [40, 100, 100]

#### - Atribut `average_rating`

Atribut ini merepresentasikan rata-rata rating yang diberikan oleh pengguna terhadap sebuah buku, berkisar antara 1.0 hingga 5.0. Nilai ini mencerminkan persepsi umum terhadap kualitas buku.

#### - Atribut `ratings_count`

Atribut ini menunjukkan jumlah total rating yang diberikan oleh pengguna terhadap suatu buku. Nilai ini dinormalisasi ke dalam skala 0–100 agar seragam dengan atribut lain.

#### - Atribut `ratings_5`

Atribut ini mencerminkan seberapa banyak pengguna yang memberikan **rating sempurna (bintang 5)** terhadap buku. Juga dinormalisasi ke dalam rentang 0–100.

## b. Fungsi keanggotaan

Fungsi keanggotaan digunakan untuk memetakan nilai numerik ke dalam derajat keanggotaan fuzzy (antara 0 hingga 1). Pada sistem rekomendasi buku ini, semua fungsi keanggotaan untuk atribut `average_rating`, `ratings_count`, dan `ratings_5` menggunakan fungsi keanggotaan segitiga (triangular membership function).

Fungsi keanggotaan segitiga merupakan salah satu fungsi keanggotaan yang paling sederhana dan sering digunakan dalam sistem fuzzy. Fungsi ini membentuk sebuah bentuk segitiga pada grafik derajat keanggotaan, dengan tiga parameter utama, yaitu:

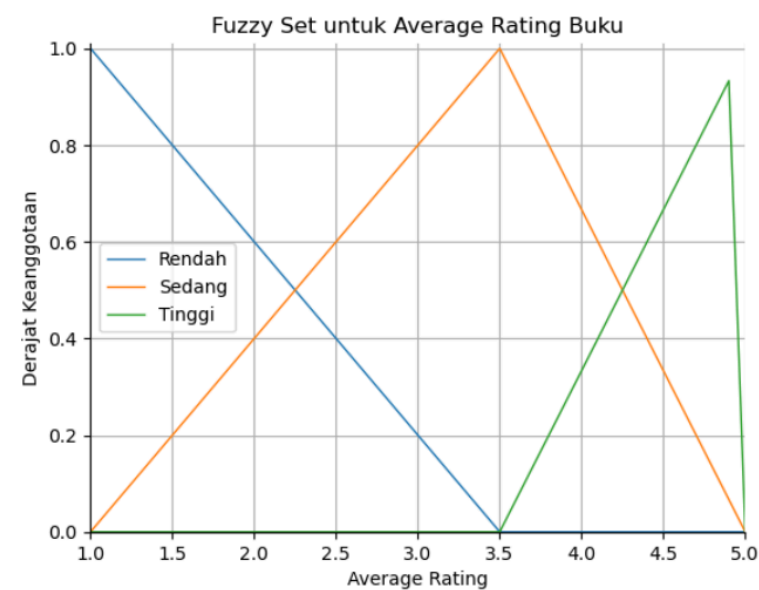
- a: titik awal di mana derajat keanggotaan mulai naik dari 0,
- b: titik puncak dengan derajat keanggotaan maksimum 1,
- c: titik akhir di mana derajat keanggotaan turun kembali ke 0.

Secara matematis, fungsi keanggotaan segitiga untuk sebuah nilai input  $x$  dapat dinyatakan sebagai berikut:

$$\mu(x) = \begin{cases} 0 & \text{jika } x \leq a \text{ atau } x \geq c \\ \frac{x-a}{b-a} & \text{jika } a < x < b \\ \frac{c-x}{c-b} & \text{jika } b \leq x < c \end{cases}$$

Fungsi ini menghasilkan derajat keanggotaan antara 0 dan 1 yang menunjukkan tingkat keanggotaan suatu nilai ke dalam kategori linguistik tertentu.

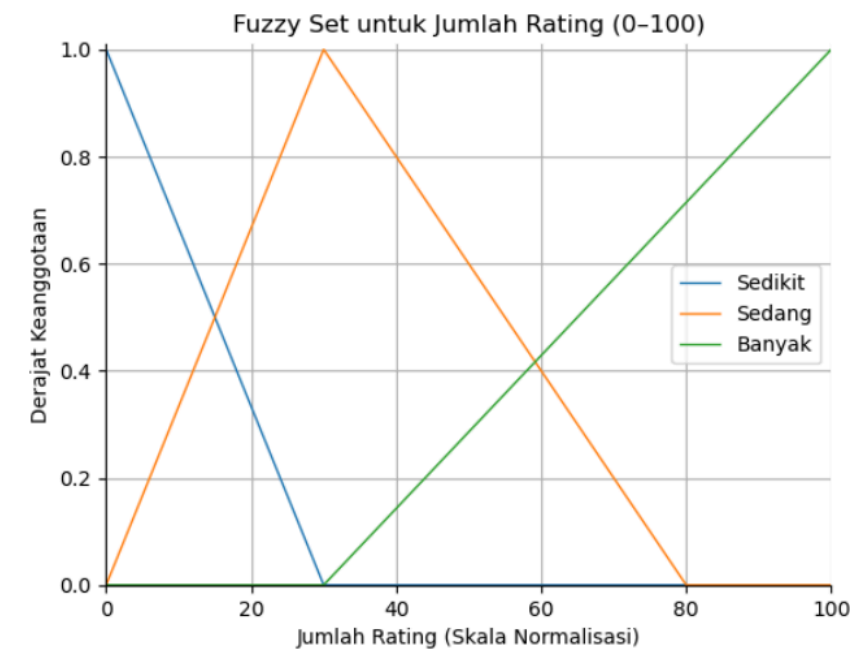
Average\_Rating



Tabel Interpretasi Fungsi Keanggotaan Fuzzy untuk Average Rating:

	Label	Range	Arti Fuzzy
0	Rendah	[1.0, 1.0, 3.5]	Mulai dari 1.0 (penuh), turun ke 0 di 3.5
1	Sedang	[1.0, 3.5, 5.0]	Mulai naik dari 1.0, puncak di 3.5, turun ke 0 di 5.0
2	Tinggi	[3.5, 5.0, 5.0]	Mulai dari 3.5, puncak di 5.0, tetap tinggi

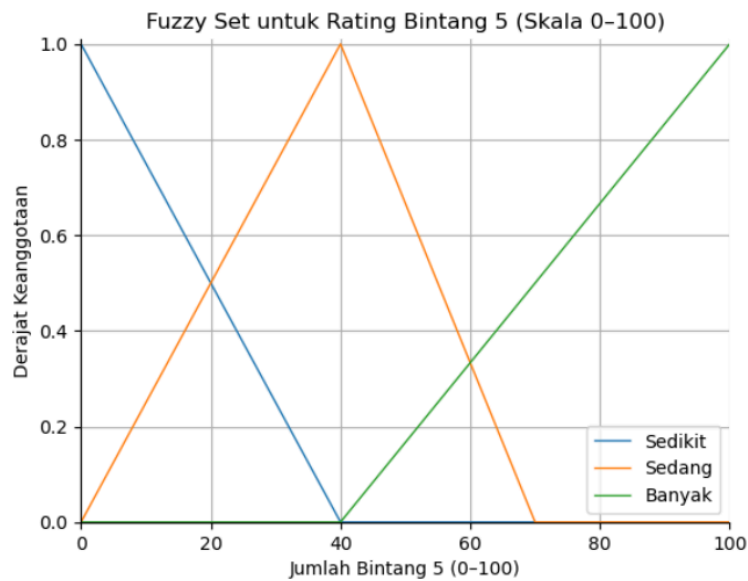
Rating\_Count



Tabel Interpretasi Fuzzy untuk Jumlah Rating (0-100):

	Label	Range	Arti Fuzzy
0	Sedikit	[0, 0, 30]	Jumlah rating rendah, penuh keanggotaan di 0, menurun ke 0 di 30
1	Sedang	[0, 30, 80]	Keanggotaan mulai naik dari 0, puncak di 30, turun ke 0 di 80
2	Banyak	[30, 100, 100]	Keanggotaan mulai dari 30, penuh keanggotaan dari 100 ke atas

## Rating\_5



Tabel Interpretasi Fungsi Keanggotaan Fuzzy untuk Rating Bintang 5 (Skala 0-100):

	Label	Range	Arti Fuzzy
0	Sedikit	[0, 0, 40]	Jumlah rating bintang 5 rendah, keanggotaan penuh di 0, turun ke 0 di 40
1	Sedang	[0, 40, 70]	Naik dari 0, puncak di 40, turun ke 0 di 70
2	Banyak	[40, 100, 100]	Mulai dari 40, penuh keanggotaan dari 100

### c. Batas-batas nilai linguistik

Untuk membangun sistem rekomendasi berbasis logika fuzzy yang akurat dan representatif, diperlukan definisi nilai linguistik (fuzzy linguistic values) terhadap atribut-atribut utama pada dataset. Nilai linguistik ini menggambarkan persepsi atau kategori manusia terhadap suatu kuantitas, misalnya “tinggi”, “sedang”, atau “rendah”. Nilai-nilai ini kemudian dikaitkan dengan fungsi keanggotaan fuzzy untuk memungkinkan sistem menilai tingkat kepastian terhadap setiap input.

Berikut adalah estimasi rentang nilai linguistik untuk masing-masing atribut, ditentukan berdasarkan eksplorasi statistik data Goodbooks-10K:

Atribut	Linguistik	Rentang Nilai
Average_rating	Rendah	[1.0, 1.0, 3.5]
	Sedang	[1.0, 3.5, 5.0]
	Tinggi	[3.5, 5.0, 5.0]
Rating_count	Sedikit	[0, 0, 30]
	Sedang	[0, 30, 80]

	Banyak	[30, 100, 100]
Rating_5	Sedikit	[0, 0, 40]
	Sedang	[0, 40, 70]
	Banyak	[40, 100, 100]

#### Average\_rating:

- Digunakan fungsi **segitiga** karena distribusi rating cenderung berbentuk normal (bell-shaped), di mana sebagian besar buku memiliki nilai rating di sekitar rata-rata (sekitar 3.5–4.2), dan hanya sedikit buku yang memiliki rating sangat rendah ( $<3.0$ ) atau sangat tinggi ( $>4.8$ ).
- **Kategori linguistik:**
  - **Rendah:** [1.0, 1.0, 3.5]  
→ Buku cenderung kurang disukai pembaca; bisa karena kualitas isi, gaya bahasa, atau topik yang kurang menarik.
  - **Sedang:** [1.0, 3.5, 5.0]  
→ Buku dengan penerimaan sedang; bisa cocok untuk sebagian pembaca, tidak terlalu buruk namun juga bukan favorit umum.
  - **Tinggi:** [3.5, 5.0, 5.0]  
→ Buku sangat disukai, mendapat apresiasi tinggi dari pengguna Goodreads.

#### Ratings\_count

- Fungsi **segitiga** dipilih untuk mencerminkan distribusi jumlah rating yang memiliki puncak pada nilai tertentu, dengan transisi nilai antar kategori yang halus.
- **Kategori linguistik:**
  - **Sedikit:** [0, 0, 30]  
→ Buku yang baru dirilis atau kurang dikenal, sehingga mendapat sedikit perhatian dari pembaca.
  - **Sedang:** [0, 30, 80]  
→ Buku dengan jumlah rating yang cukup, menunjukkan adanya minat dan perhatian yang moderat dari pembaca.
  - **Banyak:** [30, 100, 100]  
→ Buku yang sangat populer dan banyak mendapatkan perhatian pembaca, biasanya best-seller atau dari penulis terkenal.

### Ratings\_5:

- Fungsi **segitiga** digunakan untuk menggambarkan proporsi rating bintang 5 yang menunjukkan tingkat kepuasan pembaca secara spesifik.
- Kategori linguistik:
  - **Sedikit**: [0, 0, 40]  
→ Sedikit pembaca memberikan rating tertinggi, mungkin menunjukkan buku kurang memuaskan atau kurang sesuai selera.
  - **Sedang**: [0, 40, 70]  
→ Buku dengan ulasan positif yang cukup banyak, menunjukkan penerimaan yang baik namun tidak dominan.
  - **Banyak**: [40, 100, 100]  
→ Buku dengan banyak ulasan bintang 5, menunjukkan tingkat kepuasan yang sangat tinggi di kalangan pembaca.

### d. Fuzzy rule

Setelah input dikonversi menjadi nilai fuzzy, sistem melakukan inferensi berdasarkan kumpulan aturan fuzzy dalam bentuk logika "IF-THEN". Aturan ini disusun berdasarkan logika umum dalam penilaian buku: semakin tinggi rating rata-rata, jumlah rating, dan jumlah rating bintang 5, maka semakin direkomendasikan pula buku tersebut.

Sistem ini menggunakan metode Mamdani sebagai teknik inferensi fuzzy. Mamdani dipilih karena mampu menangani pernyataan linguistik yang kompleks dan mendekati logika manusia dalam pengambilan keputusan.

Berikut adalah contoh-contoh aturannya

```
rule1 = ctrl.Rule(rating['Tinggi'] & ratings_count_norm['Banyak'] & ratings_5_norm['Banyak'], rekomendasi['Ya'])
rule2 = ctrl.Rule(rating['Tinggi'] & ratings_count_norm['Sedang'] & (ratings_5_norm['Sedang'] | ratings_5_norm['Banyak']), rekomendasi['Ya'])
rule3 = ctrl.Rule(rating['Tinggi'] & ratings_count_norm['Sedikit'] & ratings_5_norm['Banyak'], rekomendasi['Ya'])
rule4 = ctrl.Rule(rating['Sedang'] & ratings_count_norm['Banyak'] & (ratings_5_norm['Sedang'] | ratings_5_norm['Banyak']), rekomendasi['Ya'])
rule5 = ctrl.Rule(rating['Sedang'] & ratings_count_norm['Sedang'] & ratings_5_norm['Banyak'], rekomendasi['Ya'])
rule6 = ctrl.Rule(rating['Sedang'] & ratings_count_norm['Sedikit'], rekomendasi['Tidak'])
rule7 = ctrl.Rule(rating['Rendah'] | ratings_5_norm['Sedikit'], rekomendasi['Tidak'])
rule8 = ctrl.Rule(rating['Rendah'] & ratings_count_norm['Banyak'], rekomendasi['Tidak'])
```

- IF rating Tinggi AND ratings\_count\_norm Banyak AND ratings\_5\_norm Banyak THEN → rekomendasi Ya
- IF rating Tinggi AND (ratings\_count\_norm Sedang OR ratings\_5\_norm Banyak) THEN → rekomendasi Ya
- IF rating Tinggi and ratings\_count\_norm Sedikit AND ratings\_5\_norm Banyak THEN → rekomendasi Ya
- IF rating Sedang AND ratings\_count\_norm Banyak AND ratings\_5\_norm Sedang OR ratings\_5\_norm Banyak THEN → rekomendasi Ya

- IF rating Sedang AND ratings\_count\_norm Sedang AND ratings\_5\_norm Banyak THEN → rekomendasi Ya
- IF rating Sedang AND ratings\_count\_norm Sedikit THEN → rekomendasi Tidak
- IF rating Rendah AND ratings\_5\_norm Sedikit THEN → rekomendasi Tidak
- IF rating Rendah AND ratings\_count\_norm Banyak THEN → rekomendasi Tidak

#### Operator Logika:

- **AND:** minimum ( $\wedge$ )
- **OR:** maksimum ( $\vee$ )

Output sistem rekomendasi fuzzy kemudian diklasifikasikan ke dalam 3 kategori:

- **Tidak Direkomendasikan** (nilai output rendah)
- **Direkomendasikan** (nilai output sedang)
- **Sangat Direkomendasikan** (nilai output tinggi)

ID :		title	average_rating	ratings_count_norm	ratings_5_norm	Rekomendasi
0		The Hunger Games (The Hunger Games, #1)	4.34	100.0	57.0	Sangat Direkomendasikan
1		Harry Potter and the Sorcerer's Stone (Harry Potter, #1)	4.44	96.0	63.0	Sangat Direkomendasikan
2		Twilight (Twilight, #1)	3.57	81.0	28.0	Tidak Direkomendasikan
3		To Kill a Mockingbird	4.25	67.0	36.0	Sangat Direkomendasikan
4		The Great Gatsby	3.89	56.0	20.0	Tidak Direkomendasikan
5		The Fault in Our Stars	4.26	49.0	27.0	Direkomendasikan
6		The Hobbit	4.25	43.0	23.0	Direkomendasikan
7		The Catcher in the Rye	3.79	43.0	15.0	Tidak Direkomendasikan
8		Angels & Demons (Robert Langdon, #1)	3.85	42.0	14.0	Tidak Direkomendasikan
9		Pride and Prejudice	4.24	43.0	24.0	Direkomendasikan

## Metode Matrix Factorization

Pada penelitian ini, digunakan metode **Matrix Factorization (MF)** untuk memprediksi rating pengguna terhadap buku berdasarkan dataset rating pengguna terhadap buku.

## 1. Pengumpulan dan Persiapan Data

Data yang digunakan dalam eksperimen ini berasal dari file CSV yang berisi informasi rating pengguna terhadap berbagai buku. Dataset ini memiliki ukuran yang cukup besar, sehingga untuk mempercepat proses pelatihan model dan mempermudah analisis, diambil subset sebanyak 10.000 baris pertama saja. Langkah ini bertujuan untuk menjaga keseimbangan antara representasi data yang cukup dan waktu komputasi yang efisien.

```
# Load data
ratings = pd.read_csv('C:\\Users\\Velisia Nihan\\Downloads\\ratings.csv')
ratings = ratings.head(10000) # subset kecil
```

## 2. Inisialisasi Parameter dan Matriks Faktor Laten

Parameter utama dalam model Matrix Factorization meliputi jumlah pengguna, jumlah buku, jumlah faktor laten, learning rate, jumlah epoch pelatihan, dan nilai regularisasi. Dalam penelitian ini, jumlah faktor laten dipilih sebesar 30, yang berarti setiap pengguna dan buku akan direpresentasikan oleh vektor berdimensi 30 yang merefleksikan karakteristik laten yang mempengaruhi rating.

Matriks faktor laten pengguna dan buku diinisialisasi secara acak dengan distribusi normal yang terpusat pada nol dan skala yang berbanding terbalik dengan jumlah faktor laten. Pendekatan ini umum dilakukan agar pembelajaran dapat dimulai dari keadaan yang netral dan menghindari bias awal.

## 3. Proses Pelatihan Model dengan Stochastic Gradient Descent (SGD)

Proses pelatihan model dilakukan menggunakan algoritma **Stochastic Gradient Descent (SGD)**, yang secara iteratif memperbarui matriks faktor laten untuk meminimalkan error prediksi rating pengguna terhadap buku. Error dihitung sebagai selisih antara rating asli dengan rating yang diprediksi oleh model. Prediksi rating diperoleh dengan melakukan perkalian dot product antara vektor faktor laten pengguna dan buku.

Pada setiap iterasi, bobot matriks P dan Q disesuaikan berdasarkan gradien error dan parameter regularisasi untuk menghindari overfitting. Regularisasi berfungsi sebagai penalti agar nilai faktor laten tidak terlalu besar, sehingga model menjadi lebih generalisasi.

Secara matematis, proses pembaruan bobot mengikuti rumus berikut:



$$P_u \leftarrow P_u + \alpha \times ((r_{ui} - \hat{r}_{ui}) \times Q_i - \lambda \times P_u)$$

$$Q_i \leftarrow Q_i + \alpha \times ((r_{ui} - \hat{r}_{ui}) \times P_u - \lambda \times Q_i)$$

di mana:

- $r_{ui}$  adalah rating asli pengguna  $u$  terhadap buku  $i$ ,
- $\hat{r}_{ui} = P_u \cdot Q_i^T$  adalah prediksi rating,
- $\alpha$  adalah learning rate,
- $\lambda$  adalah parameter regularisasi.

Performa menggunakan metrik **Mean Squared Error (MSE)** memungkinkan evaluasi seberapa baik model belajar dari data pada setiap epoch, serta membantu menentukan apakah model sudah cukup konvergen atau perlu penyesuaian parameter.

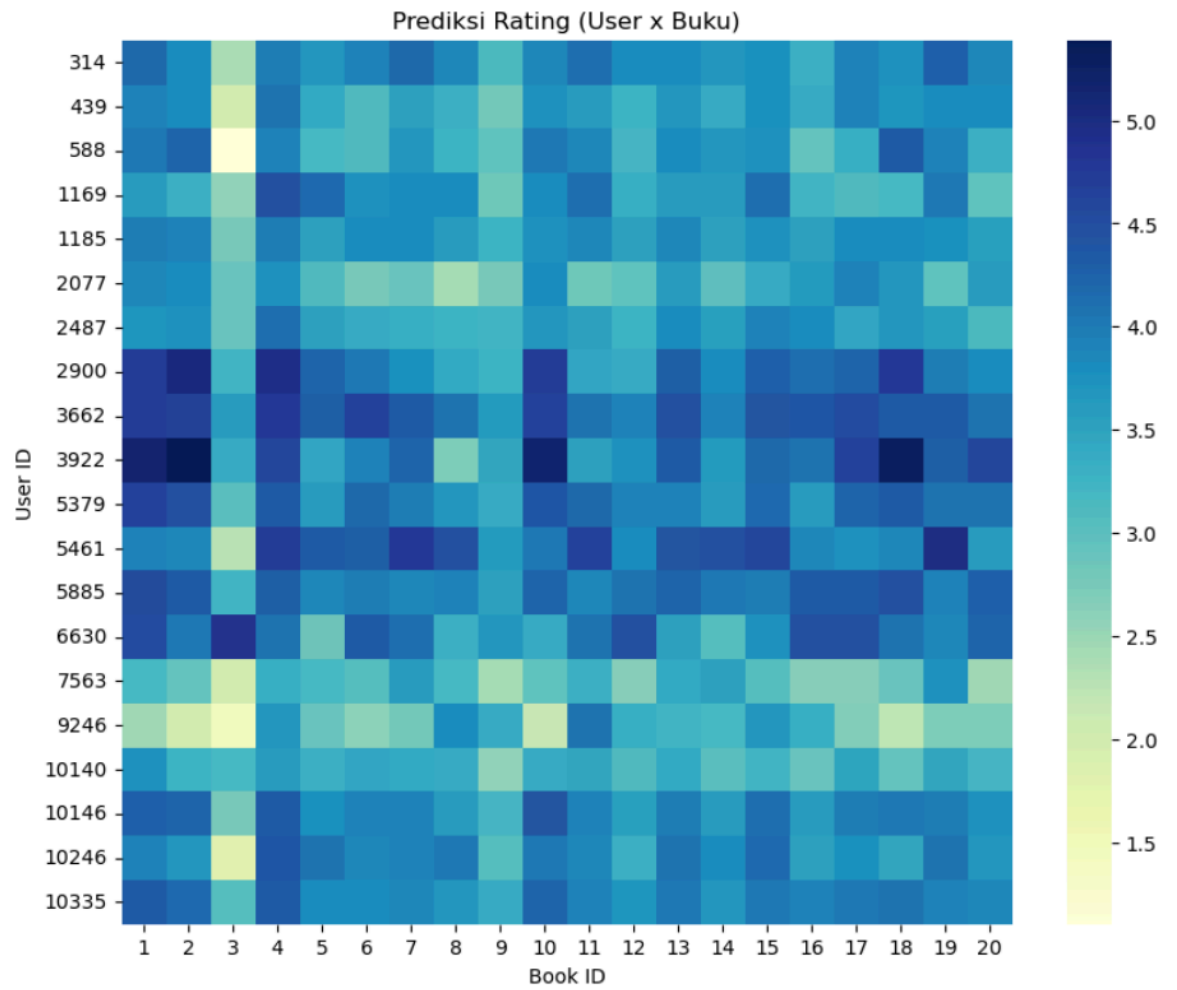
#### 4. Hasil Prediksi dan Visualisasi Matriks Rating

Setelah proses pelatihan selesai, matriks prediksi rating lengkap dapat dihitung dengan mengalikan matriks faktor laten pengguna dan matriks faktor laten buku:

$$\hat{R} = P \times Q^T$$

Matriks ini berisi nilai prediksi rating untuk seluruh kombinasi pengguna dan buku dalam dataset. Nilai-nilai ini nantinya dapat digunakan untuk merekomendasikan buku kepada pengguna berdasarkan rating tertinggi yang diprediksi.

Untuk memberikan gambaran hasil prediksi secara visual, dibuat sebuah heatmap menggunakan 20 pengguna pertama dan 20 buku pertama. Visualisasi ini menunjukkan intensitas prediksi rating dalam skala warna, dimana warna lebih gelap menandakan rating prediksi yang lebih tinggi.



## D. Hasil dan Analisis

### Hasil Sistem Rekomendasi Fuzzy

Sistem rekomendasi buku yang dibangun menggunakan logika fuzzy dengan tiga input utama yaitu *average rating*, *ratings count*, dan *ratings 5* telah diuji untuk mengklasifikasikan buku ke dalam dua kategori rekomendasi: Ya (direkomendasikan) dan Tidak (tidak direkomendasikan). Nilai-nilai input dinormalisasi ke dalam skala 0–100 sebelum diproses dalam sistem fuzzy.

Terdapat delapan aturan fuzzy (rule 1–8) yang digunakan untuk menentukan output rekomendasi berdasarkan kombinasi berbagai tingkat dari ketiga variabel input tersebut.

Hasil pengujian sistem fuzzy menunjukkan tingkat akurasi sebesar 59,89%, dengan performa yang paling baik pada kelas "Direkomendasikan". Detail performa kelas "Direkomendasikan" adalah sebagai berikut:

- Precision: 0.60
- Recall: 1.00

Hal ini menunjukkan bahwa sistem fuzzy cukup baik dalam mengenali buku yang memang direkomendasikan, namun masih memiliki kelemahan dalam mengklasifikasikan buku yang seharusnya tidak direkomendasikan.

Akurasi Sistem Fuzzy: 59.89%

Classification Report:

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1912
1	0.60	1.00	0.75	5989
2	0.00	0.00	0.00	2099
accuracy			0.60	10000
macro avg	0.20	0.33	0.25	10000
weighted avg	0.36	0.60	0.45	10000

---

### Hasil Perbandingan dengan Matrix Factorization

Sebagai pembandingan, dilakukan juga metode rekomendasi berbasis Matrix Factorization, yang memanfaatkan prediksi rating antar pengguna dan buku. Metode ini menghasilkan akurasi klasifikasi yang jauh lebih tinggi, yaitu sebesar 86,99%. Adapun performa pada kelas "Direkomendasikan" adalah:

- Precision: 0.88

- Recall: 0.94

Model matrix factorization terbukti lebih andal dalam memprediksi buku yang disukai oleh pengguna, meskipun prediksi untuk buku yang tidak disukai masih bisa ditingkatkan.

## Analisis

Dari hasil pengujian tersebut, dapat disimpulkan bahwa:

- Sistem fuzzy memiliki keunggulan dalam menjelaskan proses rekomendasi secara transparan melalui aturan-aturan logika, namun terbatas dalam akurasi ketika diterapkan pada data kompleks.
- Matrix factorization menunjukkan performa prediktif yang lebih tinggi karena mampu menangkap pola-pola tersembunyi antar pengguna dan item.

Akurasi klasifikasi berdasarkan threshold 3.5: 86.99%

Classification Report:

	precision	recall	f1-score	support
0	0.85	0.73	0.79	3294
1	0.88	0.94	0.91	6706
accuracy			0.87	10000
macro avg	0.86	0.83	0.85	10000
weighted avg	0.87	0.87	0.87	10000

## E. Kesimpulan

Berdasarkan hasil pengujian sistem rekomendasi buku menggunakan logika fuzzy, diperoleh akurasi sebesar 59,89%, dengan performa terbaik pada kelas *Direkomendasikan* (precision 0.60 dan recall 1.00). Hal ini menunjukkan bahwa sistem cukup baik dalam mengidentifikasi buku-buku yang layak direkomendasikan, namun masih kurang akurat dalam mendeteksi buku yang seharusnya tidak direkomendasikan.

Hal ini bisa terjadi karena sistem fuzzy sangat bergantung pada aturan IF–THEN yang bersifat statik dan ditentukan secara manual, sehingga kurang fleksibel dalam menangkap pola kompleks dari data pengguna secara keseluruhan.

Sebagai pembandingan, metode Matrix Factorization mampu mencapai akurasi yang jauh lebih tinggi yaitu 86,99%, dengan precision dan recall yang juga unggul pada kelas yang sama. Ini menandakan bahwa pendekatan berbasis pembelajaran mesin lebih efektif dalam menangkap preferensi pengguna secara keseluruhan.

Secara keseluruhan, sistem fuzzy menawarkan transparansi dan interpretabilitas dalam proses pengambilan keputusan, tetapi masih memiliki keterbatasan dalam hal akurasi. Oleh karena itu, untuk pengembangan sistem rekomendasi yang lebih optimal, disarankan untuk menggabungkan kelebihan logika fuzzy dengan metode berbasis machine learning seperti Matrix Factorization guna memperoleh hasil yang lebih akurat dan dapat dijelaskan secara logis.

## F. Referensi

Kaggle, "Goodbooks-10k Dataset."

<https://www.kaggle.com/datasets/zygmunt/goodbooks-10k>

Zadeh, L. A. (1965). "Fuzzy sets." *Information and Control*, 8(3), 338–353.

Jang, J.-S. R., Sun, C.-T., & Mizutani, E. (1997). *Neuro-Fuzzy and Soft Computing*. Prentice Hall.

Han, J., Kamber, M., & Pei, J. (2012). *Data Mining: Concepts and Techniques*. Morgan Kaufmann.

Ricci, F., Rokach, L., & Shapira, B. (2011). *Introduction to Recommender Systems Handbook*. Springer.

Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix Factorization Techniques for Recommender Systems. *Computer*, 42(8), 30-37.  
<https://doi.org/10.1109/MC.2009.263>

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.). Springer.