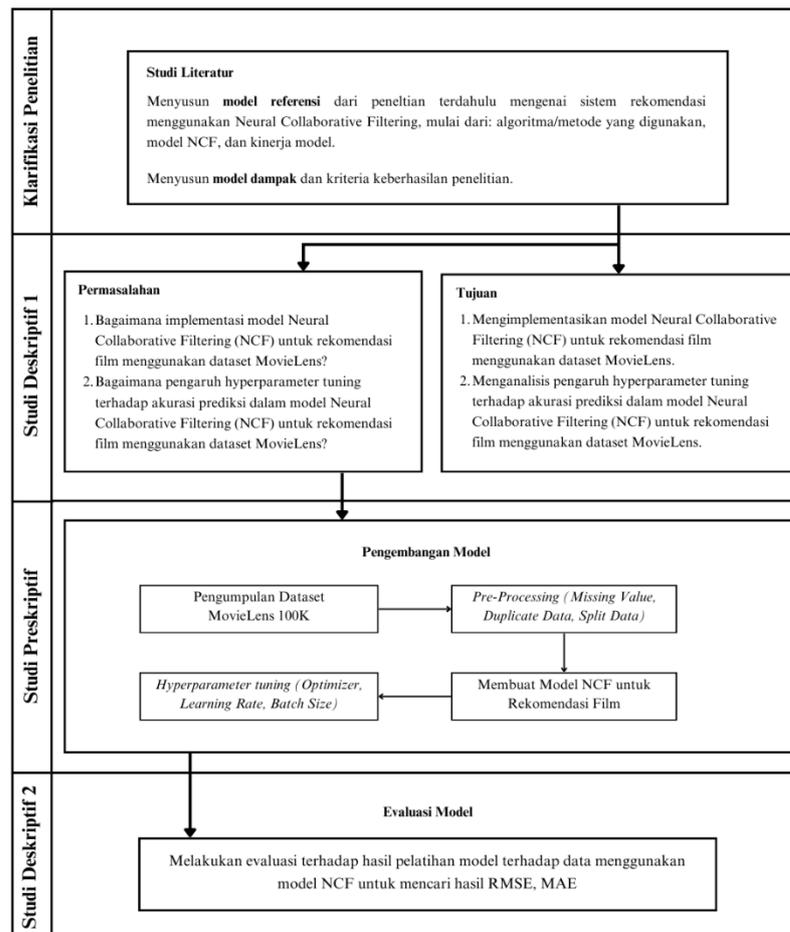


## BAB III METODE PENELITIAN

### 3.1. Desain Penelitian

Pada metodologi penelitian, dibutuhkan desain penelitian agar proses atau alur penelitian berjalan dengan terstruktur dan lancar. Penelitian ini menggunakan metode *Design Research Methodology* (DRM), yaitu metodologi preskriptif yang berfokus pada pengembangan artefak untuk menyelesaikan kelas masalah dalam suatu domain. DRM sangat cocok untuk penelitian berorientasi solusi di bidang teknologi informasi, karena memadukan aktivitas pemecahan masalah yang kreatif dengan pengembangan teknologi baru sebagai produk utamanya (Ebneyamini, 2022). Desain penelitian yang digunakan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alur Penelitian

Model penelitian lainnya yaitu ADDIE lebih berfokus pada desain instruksional dalam konteks pembelajaran, yang tidak sesuai dengan tujuan penelitian ini yang berorientasi pada pengembangan solusi teknologi (Nugraha dkk., 2024; Dermawa, 2023). Sedangkan DSRM berfokus pada perancangan solusi praktis untuk permasalahan nyata (McKenney & Handley, 2020), namun pendekatan ini kurang cocok dibandingkan dengan DRM, yang mendukung penelitian dengan tujuan teoritis dan praktis. Berdasarkan desain penelitian yang digambarkan, berikut penjelasan tahapan atau proses yang dilakukan:

### **3.1.1. Klarifikasi Penelitian**

Penjelasan awal dalam desain penelitian menggunakan DRM adalah proses klarifikasi penelitian, yang bertujuan untuk merangkum dan menganalisis literatur terkait guna memberikan landasan untuk penelitian. Dalam tahap ini, bertujuan untuk memahami permasalahan yang berkaitan dengan sistem rekomendasi menggunakan model NCF. Referensi untuk analisis teori didapat dari berbagai jurnal untuk mendapatkan pemahaman lebih mendalam mengenai sistem rekomendasi menggunakan model NCF, metode penelitian yang akan digunakan, hingga pengimplementasian model NCF. Hasil kajian literatur ini kemudian menjadi dasar bagi peneliti untuk merumuskan permasalahan dan tujuan penelitian.

### **3.1.2 Studi Deskriptif 1**

Tahap kedua dari desain penelitian DRM, berbagai permasalahan dirumuskan berdasarkan hasil studi literatur yang telah dilakukan sebelumnya. Rumusan masalah yang dihasilkan menjadi acuan utama dalam menentukan solusi yang akan dikembangkan dalam penelitian. Selanjutnya, penelitian berfokus pada penggambaran dan penjelasan permasalahan yang ditemukan, sekaligus merumuskan tujuan penelitian berdasarkan tinjauan literatur tersebut. Analisis masalah ini menjadi landasan penting dalam merancang solusi untuk mengatasi tantangan penelitian yang telah ditetapkan sebagai tujuan akhirnya.

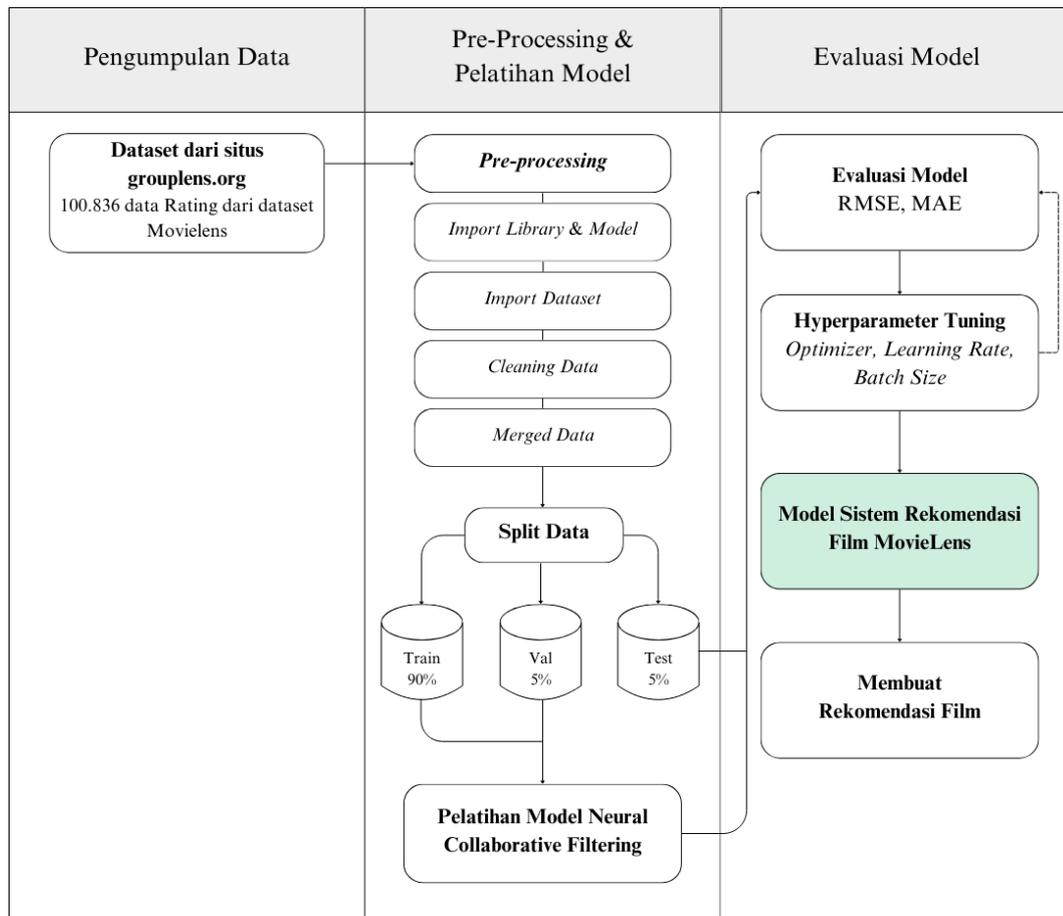
### Studi Prespektif

Pada tahap ini, dilakukan proses pengembangan model sistem rekomendasi film dengan pendekatan *Neural Collaborative Filtering* (NCF). Model ini dirancang berdasarkan referensi dari studi literatur yang telah dilakukan sebelumnya, tetapi dimodifikasi untuk menyesuaikan dengan kebutuhan penelitian ini. Penyesuaian tersebut dilakukan agar model dapat secara efektif menjawab rumusan masalah dan mencapai tujuan penelitian yang telah ditetapkan. Modifikasi mencakup berbagai aspek, seperti struktur model, parameter pelatihan, serta teknik optimalisasi yang relevan dengan konteks penelitian ini.

Setelah model selesai dikembangkan, tahap selanjutnya adalah melakukan evaluasi kinerja model. Evaluasi ini bertujuan untuk menilai sejauh mana model dapat memberikan rekomendasi yang akurat dan relevan. Untuk keperluan evaluasi, digunakan beberapa metrik utama yang umum digunakan dalam penelitian sistem rekomendasi, termasuk *Root Mean Square Error* (RMSE) yang mengukur rata-rata kesalahan prediksi dalam skala kuadrat, *Mean Absolute Error* (MAE) yang menunjukkan rata-rata besar kesalahan prediksi dalam skala absolut.

#### 3.1.3.1. Pengembangan Model

Dalam pengembangan model sistem rekomendasi film terdapat beberapa tahapan didalamnya. Model yang akan digunakan menggunakan *Neural Collaborative Filtering* (NCF), dimana model ini bertujuan untuk meningkatkan akurasi prediksi rating sehingga pengguna yang masih memiliki data yang sedikit atau sedikit memberikan *rating* film maka dapat diprediksi *rating* film yang belum diberikan *rating* lalu dengan mudah model memberikan rekomendasi sesuai dengan prediksi *rating*. Pengembangan model dibagi menjadi beberapa tahap yakni pengumpulan data, kemudian melalui tahap *preprocessing* dan pelatihan model, kemudian setelah mendapat model terbaik, maka model akan diimplementasikan untuk mendapat hasil dari penelitian. Alur dari pengembangan model ditunjukkan pada Gambar 3.2.



Gambar 3.2 Prosedur Pengembangan Model Sistem Rekomendasi Film

### 3.1.3.2. Pengumpulan Data

Tahapan pertama pada penelitian ini adalah pengumpulan data. Penelitian menggunakan data sekunder yang diambil dari *MovieLens dataset*. Dataset ini mencakup berbagai jenis data, seperti data *rating* yang berisi *user ID*, *movie ID*, *rating*, dan *timestamp*, data film yang mencakup *movie ID*, *title*, dan *genre*, data *tag* yang memuat *user ID*, *movie ID*, *tag*, dan *timestamp*, serta data link yang menghubungkan data film, *rating*, dan *tag*.

Semua data tersebut diperoleh dari situs GroupLens ([grouplens.org](http://grouplens.org)), yang menyediakan berbagai dataset untuk keperluan penelitian, termasuk dataset MovieLens. Pengumpulan data dilakukan dengan cara mengakses dan mengunduh dataset tersebut dari situs GroupLens. Dataset yang digunakan bersifat *open source* dengan ukuran sebanyak 100.836 data *rating*.

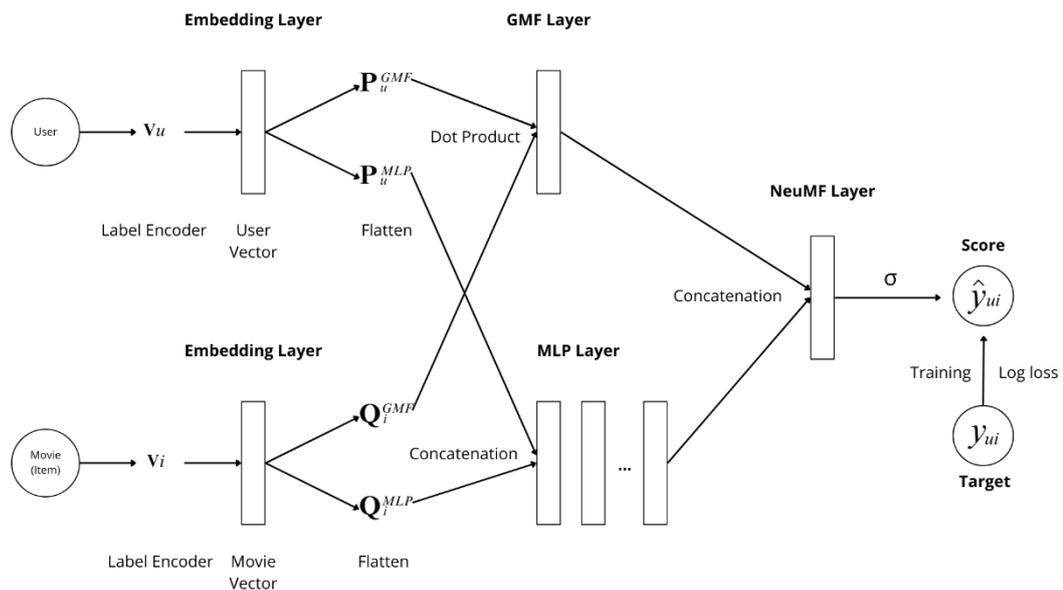
### 3.1.3.3. Preprocessing

Penelitian ini melakukan serangkaian tahapan pemrosesan data yang dirancang untuk memastikan kualitas dan konsistensi data yang digunakan dalam membangun model rekomendasi. Proses dimulai dengan pemuatan dataset yang terdiri dari ulasan film, metadata film, dan tag yang diberikan oleh pengguna. Dataset ini kemudian dibersihkan dengan menghapus nilai hilang dan duplikasi untuk menjaga kualitas data. Selanjutnya, dilakukan pemeriksaan konsistensi antar dataset, khususnya untuk memastikan bahwa setiap movieId dalam tabel ulasan memiliki pasangan yang valid di tabel metadata film. Setelah data dipastikan konsisten, ketiga dataset digabungkan menjadi satu tabel utama yang memuat informasi lengkap mengenai ulasan, metadata film, dan tag pengguna.

Data kemudian dibagi menjadi tiga subset: data latih (90%), data validasi (5%), dan data uji (5%), sebagaimana disarankan oleh penelitian (Putra dkk., 2024) dan (Ilhamsyah dkk., 2022). Kedua penelitian tersebut menyimpulkan bahwa pembagian data 90% untuk data latih, dengan 10% sisanya dibagi antara data validasi dan uji, memberikan hasil terbaik dari lima skenario pembagian data (50:50, 60:40, 70:30, 80:20, dan 90:10). Pendekatan ini digunakan untuk memastikan evaluasi yang objektif terhadap model yang dikembangkan. Langkah-langkah ini dirancang untuk menghasilkan data yang bersih, terstruktur, dan siap digunakan dalam pengembangan model rekomendasi berbasis NCF.

### 3.1.3.4. Arsitektur Neural Collaborative Filtering

Tahap selanjutnya dalam penelitian ini adalah pembangunan arsitektur model untuk prediksi rating film menggunakan metode *Neural Collaborative Filtering* (NCF). Model ini dibangun dengan memanfaatkan dua komponen utama, yaitu *Generalized Matrix Factorization* (GMF) dan *Multi-Layer Perceptron* (MLP), yang digabungkan dalam sebuah arsitektur NeuMF.



Gambar 3.3 Pengembangan Arsitektur *Neural Collaborative Filtering* (NCF)

Berdasarkan Gambar 3.3 yang menampilkan bagaimana pengembangan arsitektur NCF dalam penelitian ini, berikut ini penjelasan detail mengenai pengembangan arsitektur NCF:

### 1. *Input Layer*

- Model menerima input berupa ID pengguna dan ID film dalam bentuk *tensor* dengan ukuran (1,).
- Selanjutnya, ID pengguna dan ID film di *encoding* menggunakan label *encoder* menjadi format numerik agar kompatibel dengan model NCF.

### 2. *Embedding Layer* (Bobot sebagai Representasi vektor)

- Membuat representasi vektor untuk ID pengguna dan ID film.
- Setiap dimensi dalam embedding ini bisa dianggap sebagai sinapsis yang membawa informasi laten dari pengguna dan item.
- Matriks bobot embedding  $E_u$  untuk pengguna dan  $E_i$  untuk item bertindak sebagai bobot sinapsis.
- Matematis:

$$P_u = E_u \cdot V_u, Q_i = E_i \cdot V_i$$

Di mana:

$P_u$  : Representasi laten untuk pengguna.

$E_u$  : Matriks embedding dengan ukuran  $(N \times d)$  ( $N$  = jumlah pengguna,  $d$  = dimensi embedding).

$V_u$  : *label encoder* dari pengguna  $u$ , yang memilih baris  $u$  dari  $E_u$ .

$Q_i$  : Representasi laten untuk item.

$E_i$  : Matriks embedding untuk item dengan ukuran  $(M \times d)$  ( $M$  = jumlah item).

$V_i$  : *label encoder* dari item  $i$ .

### 3. Flatten

- Mengubah vektor hasil *embedding* dari *tensor* 2D menjadi vektor 1D.
- Digunakan untuk kompatibilitas dengan operasi *dot product* (GMF) dan *concatenation* (MLP).

### 4. GMF Layer

- Menggunakan hasil *embedding* pengguna dan film untuk menghitung *dot product* yaitu untuk menghitung kesesuaian antara vektor *user* dan *movie* dengan mengalikan elemen-elemen vektor dan menjumlahkan hasilnya.
- Menghasilkan nilai skalar yang merepresentasikan interaksi linier antara pengguna dan film.

### 5. MLP Layer (Bobot Sinapsis yang Belajar Interaksi Non-Linear)

- Menggabungkan vektor *embedding* pengguna dan film melalui *concatenation*.
- Memproses hasil *concatenation* melalui *fully connected network* dengan beberapa lapisan tersembunyi (*hidden layers*).
- Setiap lapisan memiliki bobot yang dipelajari  $W^{(l)}$  dan bias  $b^{(l)}$ , yang berperan sebagai sinapsis dalam NN klasik.

Matematis:

$$h^{(l)} = \text{ReLU}(W^{(l)} h^{(l-1)} + b^{(l)})$$

di mana:

$W^{(l)}$  : matriks bobot pada layer ke- $l$

$b^{(l)}$  : bias pada layer ke- $l$ ,

$h^{(l)}$  : hasil aktivasi setelah pembobotan.

- Menggunakan fungsi aktivasi ReLU serta menerapkan *dropout* dan regularisasi L2 untuk mencegah *overfitting*.

## 6. NeuMF Layer

1. *Output* dari GMF (skalar) dan MLP (vektor) digabungkan menggunakan *concatenation*.
2. Kombinasi informasi linier (GMF) dan non-linier (MLP) diteruskan untuk menghasilkan prediksi akhir.

## 7. Output Layer

- Menggunakan fungsi aktivasi linear untuk menghasilkan prediksi nilai kontinu seperti rating film.
- Prediksi ini mencerminkan preferensi pengguna terhadap film tertentu.

Model ini dirancang untuk melakukan prediksi *rating* film dengan menggunakan fungsi aktivasi linear pada lapisan *output*, yang sesuai dengan sifat regresi dalam memprediksi nilai kontinu seperti rating.

### 3.1.3.5 Pelatihan Model Sistem Rekomendasi Film

Pelatihan model dimulai dengan pembuatan model untuk setiap kombinasi parameter yang diuji, meliputi *optimizer*, *learning rate*, dan *batch size*. Pemilihan parameter pelatihan dalam penelitian ini dirancang dengan merujuk pada hasil penelitian sebelumnya untuk mengoptimalkan performa model rekomendasi. *Batch size* ditetapkan sebesar 512 karena penelitian sebelumnya (You dkk., 2020) menunjukkan bahwa ukuran ini mampu meningkatkan efisiensi pelatihan tanpa mengurangi akurasi. Ukuran *batch* tersebut memungkinkan model memproses lebih banyak sampel dalam satu iterasi, mempercepat pelatihan, dan menjaga stabilitas gradien, terutama pada dataset dengan ukuran 100k. *Optimizer* Adam dipilih karena hasil penelitian (Darban dkk., 2022) menunjukkan bahwa Adam unggul dibandingkan metode lain seperti RMSprop, Nadam, SGD, Adamax, Adadelta, dan Adagrad. *Optimizer* ini dapat menyesuaikan *learning rate* secara adaptif, yang memungkinkan konvergensi lebih cepat dengan stabilitas tinggi. *Learning rate* ditetapkan sebesar 0.00003 berdasarkan hasil perbandingan *learning rate* pada penelitian sebelumnya (Rashed dkk., 2019), yang membandingkan 10 nilai (0.001, 0.0005, 0.0003, 0.0002, 0.0001, 0.00005, 0.00003, 0.00002, 0.00001, 0.000005) menunjukkan bahwa nilai kecil ini memberikan

pembelajaran stabil untuk model dengan ukuran data 100K. Jenis *loss* yang digunakan adalah *mean squared error* (MSE), karena cocok untuk tugas prediksi *rating* dengan nilai kontinu (Nulhakim dkk., 2024). Pelatihan dilakukan selama 150 *epoch*, karena penelitian sebelumnya menunjukkan bahwa jumlah *epoch* ini menghasilkan performa terbaik dengan nilai *loss* yang rendah (Ramadhan & Hendrastuty, 2024). Jumlah *epoch* yang lebih panjang dibandingkan standar diperlukan untuk memastikan model dapat belajar sepenuhnya tanpa risiko *overfitting*, khususnya dalam skenario penggunaan *learning rate* kecil. *Early stopping* juga akan diterapkan pada penelitian ini, berfungsi untuk menghentikan pelatihan secara otomatis ketika akurasi validasi tidak lagi meningkat, sehingga menghindari *overfitting* dan mempercepat proses pelatihan tanpa mengorbankan kualitas model (Nurdiawan & Faqih, 2024).

Pada desain *layer* model, jumlah unit dalam representasi *embedding* ditetapkan sebesar 200. Arsitektur *hidden layer* menggunakan dua *layer* dengan jumlah unit sebanyak 500 dan *dropout* sebesar 0.5. Penggunaan *dropout* sebesar 0.5 didasarkan pada penelitian sebelumnya (Ayyiyah, 2019; Rashed, 2019) yang menggunakan *dataset* MovieLens dengan ukuran data yang sama, yaitu 100k. Penelitian tersebut menunjukkan bahwa nilai *dropout* yang besar, seperti 0.5, lebih cocok untuk data dengan ukuran besar, karena membantu mencegah *overfitting* sekaligus mempertahankan performa generalisasi model.

Setiap model dilatih menggunakan teknik pengoptimalan yang sesuai dengan kombinasi parameter yang telah ditentukan. Proses pelatihan ini bertujuan untuk menemukan model yang paling optimal dalam memprediksi *rating* film berdasarkan data yang ada. Selama pelatihan, model dipantau untuk memastikan bahwa ia dapat mempelajari pola-pola penting dari data *training* tanpa mengalami *overfitting*, dengan menggunakan data *validation* untuk memverifikasi performa model pada data yang belum dilihat sebelumnya. Secara keseluruhan, tahap pelatihan ini bertujuan untuk mengidentifikasi konfigurasi parameter terbaik yang dapat menghasilkan model dengan akurasi tertinggi dalam memprediksi *rating* film, yang menjadi dasar bagi pengembangan sistem rekomendasi berbasis *Neural Collaborative Filtering* (NCF).

### 3.1.3.6 Hyperparameter Tuning

Dalam proses pelatihan model, dilakukan penyesuaian *hyperparameter* untuk menentukan kombinasi parameter terbaik yang dapat menghasilkan performa optimal. *Hyperparameter* yang dioptimalkan meliputi jumlah *batch size*, *learning rate*, dan *optimizer*. Penelitian oleh Ayyiyah dkk., (2019) juga menunjukkan bahwa *learning rate* terbaik adalah 0,001, berdasarkan eksperimen yang membandingkan nilai 0,001 dan 0,01. *Batch size* terbaik ditemukan pada nilai 1024 dari lima opsi yang diuji, yaitu 158, 256, 512, dan 1024. Namun, hasil tersebut belum sepenuhnya memanfaatkan berbagai kemungkinan variasi nilai *hyperparameter*. Penelitian lainnya yang dilakukan oleh Mulyana & Rumaisa, (2024) menunjukkan bahwa *learning rate* 0,001 juga memberikan performa terbaik dari dua nilai yang diuji, yaitu 0,001 dan 0,01. *Batch size* terbaik dalam penelitian ini adalah 256, berdasarkan pengujian nilai 128, 256, dan 512. Hasil ini mengindikasikan pentingnya parameter *learning rate* dan *batch size* dalam menghasilkan evaluasi prediksi yang lebih baik. Namun, penelitian sebelumnya masih terbatas dalam eksplorasi nilai *hyperparameter* lainnya. Pada penelitian ini, akan diuji berbagai nilai *learning rate*, dimulai dari nilai kecil seperti 0.001, 0.0005, 0.0003, 0.0002, 0.0001, 0.00005, 0.00003, 0.00002, 0.00001, dan 0.000005. Pemilihan nilai *learning rate* kecil ini didasarkan pada ukuran data yang besar, merujuk pada penelitian Rashed dkk., (2019) yang juga menggunakan data sebesar 100k. Demikian pula, *batch size* belum mencakup nilai 64, yang potensial memberikan kinerja berbeda. Dalam eksperimen ini, akan diuji dengan *batch size* (64, 128, 256, 512, dan 1024) untuk mengeksplorasi dampaknya terhadap performa model.

Selain itu, penelitian oleh Anggara dkk., (2023) menyoroti pentingnya pemilihan *optimizer*. Dalam penelitian tersebut, *Optimizer Adam* menghasilkan nilai akurasi tertinggi sebesar 68,61%, diikuti oleh *Stochastic Gradient Descent* (SGD) dengan akurasi sebesar 57,68%, dan RMSprop dengan akurasi sebesar 54,83%. Hasil ini menunjukkan bahwa pemilihan *optimizer* memiliki pengaruh signifikan terhadap performa model. Namun, penelitian sebelumnya, khususnya pada sistem rekomendasi film menggunakan *Neural Collaborative Filtering* (NCF), belum banyak melakukan perbandingan antar-*optimizer*. Oleh karena itu, penelitian ini akan mengimplementasikan berbagai *optimizer*, termasuk Adam, RMSprop, Nadam, SGD,

Adamax, Adagrad, untuk mengevaluasi dampaknya terhadap kinerja model. Dengan memvariasikan nilai *learning rate*, *batch size*, dan jenis *optimizer*, penelitian ini diharapkan dapat memberikan wawasan mendalam tentang pengaruh kombinasi *hyperparameter* terhadap kinerja model dan menghasilkan konfigurasi yang optimal untuk prediksi.

#### 3.1.4. Studi Deskriptif 2

Setelah proses pengembangan model selesai, dilakukan evaluasi terhadap model yang dihasilkan. Evaluasi ini bertujuan untuk memperoleh pemahaman mendalam mengenai kinerja model yang dibangun. Pengujian dilakukan menggunakan data validasi untuk menentukan konfigurasi *hyperparameter* yang optimal tanpa melibatkan data uji. Setelah model dengan performa terbaik diperoleh, evaluasi selanjutnya dilakukan menggunakan data uji untuk memastikan hasil evaluasi bersifat objektif, mengingat data uji tidak digunakan selama proses pelatihan model. Penelitian terdahulu yang mengimplementasikan sistem rekomendasi menggunakan *Neural Collaborative Filtering* (NCF), seperti yang dilakukan oleh (Marzuki dkk., 2024) menggunakan evaluasi berbasis *Root Mean Square Error* (RMSE) dan *Mean Absolute Error* (MAE) dalam penelitian sistem rekomendasi dengan output prediksi *rating*. Evaluasi lain seperti *recall*, *precision*, dan *F1-Measure* tidak digunakan karena fokus pada perankingan dan klasifikasi (Rawat.dkk., 2019; Ariyanto dkk., 2021). Oleh karena itu, RMSE dan MAE relevan untuk menilai kinerja model yang dikembangkan.

### 3.2. Instrumen Penelitian

Instrumen yang digunakan dalam penelitian ini untuk pengumpulan data terdiri dari metrik evaluasi RMSE dan MAE, yang berfungsi sebagai alat untuk membantu proses pelatihan dan penilaian kinerja model. Berikut adalah rumus perhitungan RMSE dan MAE yang digunakan dalam penelitian ini.

#### 1. RMSE

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - x_i)^2}{n}} \quad (1)$$

Di mana:

$y_i$  : Nilai aktual (nilai yang benar atau yang diamati).

$x_i$  : Nilai prediksi (nilai yang dihasilkan oleh model).

$n$  : Jumlah total data (jumlah pasangan nilai aktual dan prediksi).

$y_i - x_i$  : Selisih antara nilai aktual dan nilai prediksi untuk data ke- $i$ .

$(y_i - x_i)^2$  : Kuadrat dari kesalahan prediksi untuk memastikan nilainya positif.

$\sum_{i=1}^n$  : Penjumlahan semua kuadrat kesalahan dari data pertama ( $i = 1$ ) hingga data terakhir ( $i = n$ ).

$\frac{1}{n}$  : Rata-rata dari kuadrat kesalahan.

$\sqrt{\quad}$  : Akar kuadrat dari rata-rata kuadrat kesalahan untuk mengembalikan skala kesalahan ke skala aslinya.

#### 2. MAE

$$MAE = \frac{\sum_{i=1}^n |y_i - x_i|}{n} \quad (2)$$

Di mana:

$y_i$  : Nilai aktual (nilai yang benar atau yang diamati).

$x_i$  : Nilai prediksi (nilai yang dihasilkan oleh model).

$n$  : Jumlah total data (jumlah pasangan nilai aktual dan prediksi).

$y_i - x_i$  : Selisih antara nilai aktual dan nilai prediksi untuk data ke- $i$ .

$(y_i - x_i)^2$  : Kuadrat dari kesalahan prediksi untuk memastikan nilainya positif.

$\sum_{i=1}^n$  : Penjumlahan untuk semua data dari indeks 1 hingga  $n$ .

$\frac{1}{n}$  : Rata-rata dari kuadrat kesalahan.

### 3.3. Alat dan Bahan Penelitian

Perangkat atau alat yang digunakan untuk keberlangsungan dalam penelitian dan pengembangan selama pengujian aplikasi web Vocasia adalah sebagai berikut:

Tabel 3.1  
Konfigurasi Perangkat Keras dan Perangkat Lunak Penelitian

Konfigurasi		Keterangan
<i>Hardware</i>	Processor	Intel i7 Gen 11
	RAM	8 GB
	SSD	512 GB
	GPU	Nvidia
<i>Software</i>	Sistem Operasi	Windows 11
	Python	3.10
	IDE	Visual Studio Code, Colab

Tabel 3.2  
Daftar *Library*

Library	Modul
Matplotlib	Visualisasi
Pandas	Pengolahan Data
Numpy	Pengolahan Data
TensorFlow	Pemodelan
Sklearn	<i>Machine Learning</i>

### 3.4. Analisis Data

Setelah memperoleh hasil dari setiap percobaan pelatihan model yang telah dievaluasi, dilakukan analisis untuk mengidentifikasi pola, tren, dan temuan signifikan yang dapat menjelaskan kinerja model secara rinci. Analisis data ini mencakup perbandingan kinerja model pada berbagai konfigurasi *hyperparameter* untuk menentukan konfigurasi yang memberikan hasil terbaik berdasarkan metrik evaluasi yang telah ditentukan. Metrik-metrik tersebut meliputi *Root Mean Square Error* (RMSE) dan *Mean Absolute Error* (MAE).

Untuk memberikan penilaian yang lebih komprehensif terhadap hasil akhir RMSE, beberapa indikator berikut akan digunakan pada penelitian ini untuk menilai apakah hasil RMSE yang diperoleh dapat dikategorikan baik atau buruk:

1. Menggunakan Standar dari Penelitian Sebelumnya

Dalam penelitian ini, evaluasi hasil akhir dilakukan dengan membandingkan nilai RMSE yang diperoleh dengan hasil yang dilaporkan dalam penelitian sebelumnya yang juga menggunakan dataset MovieLens 100K. Berdasarkan data yang tersedia di situs *Papers with Code* (<https://paperswithcode.com>), yang menyediakan rangkuman hasil RMSE terbaik dari berbagai penelitian yang diterapkan pada dataset MovieLens 100K, terdapat variasi nilai RMSE yang mencerminkan kinerja berbagai model dalam sistem rekomendasi. Penelitian oleh Darban & Valipour, (2022) dengan model GHRIS (*Graph-based Hybrid Recommendation System*) mencatatkan nilai RMSE terbaik sebesar 0,887, yang menempatkan penelitian ini pada urutan pertama dalam daftar hasil terbaik. Di sisi lain, penelitian oleh Kalofolias dkk., (2014) menggunakan model *Matrix Completion on Graphs*, yang menghasilkan nilai RMSE sebesar 0,996, menempatkan penelitian ini pada urutan terakhir dalam rangkaian penelitian yang diuji. Berdasarkan perbandingan dengan penelitian sebelumnya yang menggunakan dataset MovieLens 100K, nilai akhir RMSE pada penelitian ini dapat dievaluasi dalam tiga kategori, yaitu sangat baik, baik, dan kurang baik, dengan acuan sebagai berikut:

Tabel 3.3  
Indikator Penilaian RMSE berdasarkan Penelitian Sebelumnya

Penilaian	Rentang RMSE	Indikator
Sangat Baik	$RMSE < 0,887$	Kemampuan Prediksi Yang Optimal
Baik	$0,887 \leq RMSE \leq 0,996$	Masih Ada Ruang Untuk Perbaikan
Kurang Baik	$RMSE > 0,996$	Membutuhkan Perbaikan Signifikan

## 2. Membandingkan dengan Standar Deviasi Data

Indikator lainnya untuk memberikan penilaian yang lebih komprehensif terhadap hasil akhir RMSE yaitu membandingkan dengan standar deviasi data yang sudah dilakukan oleh penelitian sebelumnya (Köse dkk., 2017; Sulastri dkk., 2023). Dalam konteks rekomendasi film, standar deviasi dari data *rating* digunakan untuk memahami seberapa bervariasi atau konsisten penilaian yang diberikan oleh pengguna terhadap film (Rizqi dkk., 2024). Standar deviasi yang rendah menunjukkan bahwa pengguna cenderung memberikan *rating* yang serupa, sementara standar deviasi yang tinggi menunjukkan bahwa *rating* yang diberikan sangat bervariasi, mencerminkan berbagai preferensi individu yang lebih besar (Dary & Ilyas, 2017).

Pada dataset MovieLens 100K, hasil perhitungan standar deviasi adalah 1.0425. Dengan standar deviasi sebesar 1.0425, penyebaran *rating* dalam dataset relatif sedang. Artinya, *rating* yang diberikan oleh pengguna cukup bervariasi, tetapi tidak terlalu ekstrem. Jika RMSE terlalu tinggi, ini bisa menunjukkan bahwa model tidak berhasil menangani variasi tersebut dengan baik. Berdasarkan nilai tersebut, kemudian dapat diinterpretasikan kualitas model rekomendasi dan kinerja prediksi dengan beberapa indikator:

Tabel 3.4  
Indikator Penilaian RMSE berdasarkan Standar Deviasi Data

Penilaian	Rentang RMSE	Indikator
Sangat Baik	$RMSE < 1,0425$	Model mampu menangkap variasi dalam data daripada hanya mengandalkan estimasi rata-rata
Kurang Baik	$RMSE > 1,0425$	Model tidak mampu menangkap pola rating dengan baik

Untuk menilai efektivitas model secara lebih komprehensif, standar deviasi dari data aktual dapat digunakan sebagai tolok ukur. Jika model menghasilkan RMSE yang lebih rendah dari standar deviasi ini, maka model tersebut dianggap lebih efektif dalam mengurangi kesalahan prediksi dibandingkan menggunakan pendekatan rata-rata.