

**IMPLEMENTASI *MODEL-BASED COLLABORATIVE  
FILTERING* PADA SISTEM REKOMENDASI FILM  
MENGUNAKAN ALGORITMA SVD  
(*Singular Value Decomposition*)**

**SKRIPSI**

**OLEH:**

**MUHAMMAD RIZKY SITORUS**

**188160044**



**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2023**

**UNIVERSITAS MEDAN AREA**

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area

Document Accepted 23/1/24

Access From ([repository.uma.ac.id](https://repository.uma.ac.id))23/1/24

**IMPLEMENTASI *MODEL-BASED COLLABORATIVE  
FILTERING* PADA SISTEM REKOMENDASI FILM  
MENGUNAKAN ALGORITMA SVD  
(*Singular Value Decomposition*)**

**PROPOSAL SKRIPSI**

Diajukan sebagai Salah Satu Syarat untuk Memperoleh  
Gelar Sarjana di Fakultas Teknik  
Universitas Medan Area

**OLEH :**

**MUHAMMAD RIZKY SITORUS**

**188160044**

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS MEDAN AREA  
MEDAN  
2023**

UNIVERSITAS MEDAN AREA

© Hak Cipta Di Lindungi Undang-Undang

1. Dilarang Mengutip sebagian atau seluruh dokumen ini tanpa mencantumkan sumber
2. Pengutipan hanya untuk keperluan pendidikan, penelitian dan penulisan karya ilmiah
3. Dilarang memperbanyak sebagian atau seluruh karya ini dalam bentuk apapun tanpa izin Universitas Medan Area


Document Accepted 23/1/24

Access From (repository.uma.ac.id)23/1/24

## HALAMAN PENGESAHAN

Judul Skripsi : Implementasi *Model-Based Collaborative Filtering* pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (*Singular Value Decomposition*)  
Nama : Muhammad Rizky Sitorus  
NPM : 188160044  
Fakultas : Teknik

Disetujui Oleh  
Komisi Pembimbing

  
Rizki Muliono, S. Kom, M. Kom.  
Pembimbing



Dr. Edy Sulaino, S.T., M.T.  
Dekan Fakultas Teknik



Rizki Muliono, S. Kom, M. Kom.  
Ka. Prodi Teknik Informatika

Tanggal Lulus : 14 September 2023

## HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperoleh gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainnya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 14 September 2023  
Penulis,



Muhammad Rizky Sitorus  
NIM 188160044

## HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR/SKRIPSI/TESIS UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademik Universitas Medan Area, saya yang bertanda tangan di bawah ini:

Nama : Muhammad Rizky Sitorus

NPM : 188160044

Program Studi : Teknik Informatika

Fakultas : Teknik

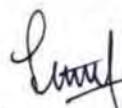
Jenis karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneksklusif (Non-exclusive Royalty-Free Right)** atas karya ilmiah saya yang berjudul :

**Implementasi *Model-Based Collaborative Filtering* pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (*Singular Value Decomposition*)**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Medan Area berhak menyimpan, mengalihmedia/format-kan, mengelola dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir/skripsi/tesis saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta. Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Medan  
Pada tanggal : 14 September 2023  
Yang menyatakan



(Muhammad Rizky Sitorus)

## ABSTRAK

Sistem rekomendasi film telah menjadi elemen penting dalam berbagai aplikasi dan situs *web* di era digital saat ini. Sistem ini membantu pengguna menemukan film-film yang sesuai dengan minat dan preferensi mereka. Terdapat berbagai jenis sistem rekomendasi seperti *Content Based Filtering*, *Collaborative Filtering* dan *Hybrid Filtering* (gabungan dari *Content Based Filtering* dan *Collaborative Filtering*). Dalam penelitian ini, dilakukan implementasi *Model-Based Collaborative Filtering* menggunakan algoritma *Singular Value Decomposition* (SVD) pada sistem rekomendasi film. Algoritma SVD digunakan untuk menganalisis interaksi pengguna dengan film-film tersebut dan menghasilkan representasi *latent* yang lebih ringkas tetapi tetap informatif. Dengan demikian, rekomendasi film dapat dibuat berdasarkan preferensi yang mendasari perilaku pengguna. Hasil evaluasi prediksi menunjukkan bahwa penggunaan algoritma SVD dengan nilai  $k = 3000$  dalam sistem rekomendasi film menghasilkan nilai evaluasi RMSE (*Root Mean Square Error*) sebesar 0,4002 yang menunjukkan tingkat akurasi model yang baik. Selain itu, nilai Mean Absolute Error (MAE) yang dihasilkan adalah 0,1186 yang menunjukkan tingkat kesalahan prediksi yang rendah.

**Kata Kunci :** *Model-Based, Collaborative Filtering, Singular Value Decomposition (SVD), Sistem Rekomendasi Film, Rating.*

## ABSTRACT

*Film recommendation systems have become a crucial element in various applications and websites in the current digital era. These systems assist users in discovering films that align with their interests and preferences. There are various types of recommendation systems such as Content-Based Filtering, Collaborative Filtering, and Hybrid Filtering (a combination of Content-Based Filtering and Collaborative Filtering). In this research, the implementation of Model-Based Collaborative Filtering using the Singular Value Decomposition (SVD) algorithm in the film recommendation system is carried out. The SVD algorithm is utilized to analyze user interactions with these films and generate more concise yet informative latent representations. Therefore, film recommendations can be made based on the underlying user preferences. Prediction evaluation results indicate that the use of the SVD algorithm with a value of  $k = 3000$  in the film recommendation system yields an RMSE (Root Mean Square Error) evaluation score of 0.4002, demonstrating a high level of model accuracy. Furthermore, the Mean Absolute Error (MAE) generated is 0.1186, indicating a low level of prediction error.*

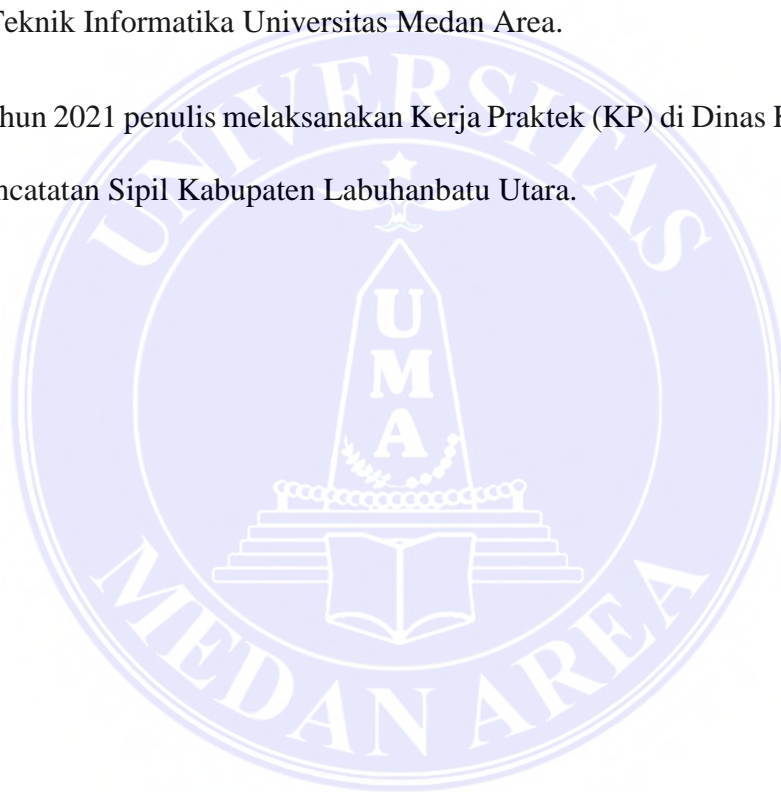
**Keywords:** *Model-Based, Collaborative Filtering, Singular Value Decomposition (SVD), Film Recommendation System, Rating.*

## RIWAYAT HIDUP

Penulis merupakan putra atau anak ke-4 (empat) dari ayah Zulhajar Sitorus dan ibu Jumiah yang dilahirkan di Simpang Siranggong, Desa Damuli Pekan, pada tanggal 09 Juli 1999.

Pada tahun 2017 Penulis lulus dari SMK Negeri Pertanian Pembangunan 1 Kualuh Selatan, lalu pada tahun 2018 terdaftar sebagai mahasiswa Fakultas Teknik Prodi Teknik Informatika Universitas Medan Area.

Pada tahun 2021 penulis melaksanakan Kerja Praktek (KP) di Dinas Kependudukan dan Pencatatan Sipil Kabupaten Labuhanbatu Utara.





## KATA PENGANTAR

Puji syukur penulis ucapkan kepada Allah SWT, karena atas rahmat dan hidayah-Nya penulis dapat menyelesaikan skripsi dengan judul **“Implementasi Model-Based Collaborative Filtering pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (Singular Value Decomposition)”** ini dengan baik. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan Program Strata-1 pada Fakultas Teknik Program Studi Teknik Informatika di Universitas Medan Area.

Dalam proses menyelesaikan skripsi ini, penulis menyadari bahwa tulisan ini masih jauh dari kata sempurna dan juga terdapat banyak kekurangan. Penulis juga mengharapkan kritik dan saran yang sifatnya membangun dari para pembaca. Kemudian penulis ingin mengucapkan terima kasih kepada:

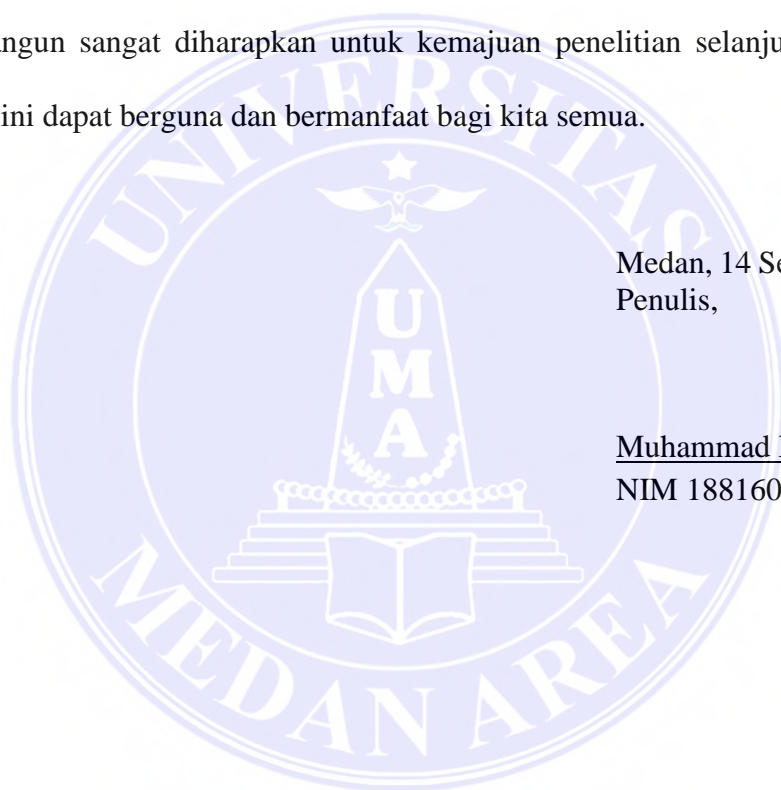
1. Bapak Prof. Dr. Dadan Ramdan, M.Eng., M.Sc., selaku Rektor Universitas Medan Area.
2. Bapak Dr. Eng. Supriatno, S.T, M.T. selaku Dekan Fakultas Teknik Universitas Medan Area.
3. Bapak Rizki Muliono, S.Kom., M.Kom., selaku Kepala Program Studi Teknik Informatika yang juga selaku dosen pembimbing yang telah membimbing, mengarahkan, dan memberikan motivasi kepada penulis dalam proses menyelesaikan skripsi ini.
4. Seluruh Dosen Teknik Informatika Universitas Medan Area yang selama ini telah membekali penulis dengan ilmu yang sangat bermanfaat.
5. Seluruh Pegawai Universitas Medan Area yang telah membantu dalam proses administrasi.

6. Kedua Orang tua dan keluarga penulis yang dengan penuh kasih sayang telah mendidik penulis serta memberikan doa terbaiknya kepada penulis.
7. Teman-teman Teknik Informatika Reguler 2018, terima kasih atas persahabatan dan persaudaraannya selama ini. Semoga Allah memudahkan untuk menyelesaikan study S-1 ini.

Penulis menyadari dalam penyusunan skripsi ini, masih terdapat banyak kekurangan dan jauh dari kesempurnaan, oleh karena itu kritik dan saran yang membangun sangat diharapkan untuk kemajuan penelitian selanjutnya. Semoga skripsi ini dapat berguna dan bermanfaat bagi kita semua.

Medan, 14 September 2023  
Penulis,

Muhammad Rizky Sitorus  
NIM 188160044



## DAFTAR ISI

	Halaman
<b>ABSTRAK</b> .....	vi
<b>ABSTRACT</b> .....	vii
<b>RIWAYAT HIDUP</b> .....	viii
<b>KATA PENGANTAR</b> .....	ix
<b>DAFTAR ISI</b> .....	xi
<b>DAFTAR TABEL</b> .....	xiii
<b>DAFTAR GAMBAR</b> .....	xiv
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian .....	4
1.4 Manfaat Penelitian .....	5
1.5 Batasan Masalah .....	5
<b>BAB II TINJAUAN PUSTAKA</b> .....	6
2.1 Sistem Rekomendasi .....	6
2.2 <i>Collaborative Filtering</i> .....	7
2.2.1 <i>Model-Based Collaborative Filtering</i> .....	8
2.2.2 <i>Singular Value Decomposition (SVD)</i> .....	9
2.3 <i>Mean Absolute Error (MAE)</i> .....	12
2.4 <i>Root Mean Square Error (RMSE)</i> .....	12
2.5 Film .....	13
2.6 Penelitian Terdahulu .....	13
<b>BAB III METODOLOGI PENELITIAN</b> .....	16
3.1 Tahapan Penelitian .....	16
3.2 Pengumpulan Data .....	16
3.3 Preprocessing Data .....	18
3.4 Pembuatan Model .....	18
3.5 Evaluasi Model .....	22
3.6 Analisis Hasil .....	24
<b>BAB IV HASIL DAN PEMBAHASAN</b> .....	25
4.1 Hasil .....	25
4.2 Pembahasan .....	25
4.2.1 Pengumpulan Data .....	25
4.2.2 Preprocessing Data .....	27
4.2.3 Algoritma SVD .....	29
4.2.4 Evaluasi Model .....	32
4.2.5 Implementasi SDV Pada Sistem Rekomendasi Film .....	35

<b>BAB V KESIMPULAN DAN SARAN.....</b>	<b>39</b>
5.1 Kesimpulan.....	39
5.2 Saran.....	39
<b>DAFTAR PUSTAKA.....</b>	<b>40</b>
<b>LAMPIRAN.....</b>	<b>43</b>



## DAFTAR TABEL

Tabel 2. 1 Penelitian Terdahulu .....	14
Tabel 3. 1 Data Anime.csv .....	17
Tabel 3. 2 Data Rating.csv .....	18
Tabel 3. 3 Data Sample <i>User-Rating</i> .....	19
Tabel 3. 4 Contoh Prediksi <i>Rating</i> .....	21
Tabel 3. 5 Evaluasi <i>Rating</i> prediksi .....	23
Tabel 4. 1 Hasil Pengujian Pengaruh Nilai K .....	34



## DAFTAR GAMBAR

Gambar 2. 1 Klasifikasi sistem rekomendasi .....	7
Gambar 2. 2 Singular value decomposition matrix.....	10
Gambar 3. 1 Tahapan Penelitian .....	16
Gambar 4. 1 Koneksi ke <i>Google Drive</i> .....	26
Gambar 4. 2 Data Film.....	26
Gambar 4. 3 Data <i>Rating</i> .....	27
Gambar 4. 4 Tahapan <i>Preprocessing</i> Data Film.....	27
Gambar 4. 5 Tahapan <i>Preprocessing</i> Data <i>Rating</i> .....	28
Gambar 4. 6 Hasil <i>Preprocessing</i> Data.....	29
Gambar 4. 7 Tahapan Pembentukan Matriks .....	29
Gambar 4. 8 Matriks U.....	30
Gambar 4. 9 Matriks S .....	30
Gambar 4. 10 Matriks V.....	30
Gambar 4. 11 Mencari Nilai $\mu$ .....	31
Gambar 4. 12 Fungsi Prediksi <i>Rating</i> .....	31
Gambar 4. 13 Fungsi Rekomendasi Film.....	32
Gambar 4. 14 Hasil Rekomendasi Film .....	32
Gambar 4. 15 Tahapan mencari nilai RMSE dan MAE.....	33
Gambar 4. 16 Hasil RMSE dan MAE.....	33
Gambar 4. 17 Halaman Utama Sistem Rekomendasi Film.....	35
Gambar 4. 18 Halaman Detail Film .....	36
Gambar 4. 19 Halaman <i>Sign Up</i> .....	36
Gambar 4. 20 Halaman <i>Sign in/Log in</i> .....	37
Gambar 4. 21 Halaman Detail Film Setelah <i>Login</i> .....	37
Gambar 4. 22 Halam <i>Profile User</i> .....	38
Gambar 4. 23 Halaman Hasil Rekomendasi Film.....	38

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Di era digital saat ini, sistem rekomendasi telah menjadi bagian penting dari banyak aplikasi dan situs *web*, seperti *e-commerce*, media *streaming*, dan lain-lain. Sistem rekomendasi membantu pengguna menemukan *item* yang sesuai dengan minat dan preferensi mereka. Dalam bidang hiburan, sistem rekomendasi film sangat berguna bagi pengguna untuk menemukan film yang sesuai dengan minat mereka dan membuat pilihan yang lebih mudah. Dengan menggunakan data historis tentang interaksi pengguna dengan *item* atau konten sebelumnya, sistem rekomendasi dapat mengidentifikasi preferensi pengguna dan memberikan rekomendasi yang sesuai dengan minat mereka.

Sistem rekomendasi telah menjadi topik yang semakin penting dalam dunia teknologi informasi dan bisnis, karena mampu memberikan rekomendasi yang personal dan akurat kepada pengguna. Dalam era di mana data semakin melimpah, sistem rekomendasi menjadi semakin efektif dan efisien untuk memproses data dan menghasilkan rekomendasi yang relevan. Terdapat berbagai jenis sistem rekomendasi seperti *Content Based Filtering*, *Collaborative Filtering* dan *Hybrid Filtering* (gabungan dari *Content Based Filtering* dan *Collaborative Filtering*) (Zhang dkk., 2019).

Penelitian yang dilakukan oleh Larasati & Februariyanti (2021) pada Sistem Rekomendasi *Product Emina Cosmetics* Dengan Menggunakan Metode *Content-Based Filtering* menyebutkan bahwa sistem rekomendasi dengan metode *Content Based Filtering* umumnya digunakan untuk merekomendasikan artikel, berita, atau

situs *web*. Metode ini akan mengevaluasi data yang terkait dengan suatu *item* lalu membandingkannya dengan data *item* yang pernah diakses atau dipilih oleh pengguna. Penelitian menggunakan teknik *Content Based Collaborative Filtering* ini menghasilkan 10 produk yang direkomendasikan dengan nilai *similaritas* tertinggi sebesar 0,7195 berdasarkan perhitungan *cosine similarity*.

Penelitian mengenai *collaborative filtering* pada *Recommender System* *Movie Netflix using Collaborative Filtering with Weighted Slope One Algorithm in Twitter* yang dilakukan oleh Rifaldy & Setiawan, (2022) memaparkan bahwa *Collaborative Filtering* terbukti mampu menghasilkan nilai MAE yang cukup rendah sebesar 0,924 dengan  $N = 5$  saat melakukan eksperimen menggunakan *dataset* yang terdiri dari 6183 *records* dengan 35 pengguna dan 791 film. *Collaborative Filtering* juga terbukti mampu meminimalkan jumlah kekosongan data, namun tidak semua data kosong dapat diatasi. *Weighted Slope One* juga terbukti mampu memprediksi nilai *rating* dengan cukup baik dan dapat menghasilkan nilai MAE yang cukup rendah sebesar 0,568 dengan  $k = 10$ . Dapat disimpulkan bahwa dengan nilai MAE yang rendah yang dihasilkan, sistem dapat bekerja dengan baik menggunakan *dataset* yang diperoleh dari Twitter.

Dari beberapa metode sistem rekomendasi yang telah disebutkan sebelumnya, metode *Collaborative Filtering* (CF) merupakan yang paling sering digunakan. *Collaborative Filtering* adalah metode yang berdasarkan pada data interaksi pengguna dengan *item*. *Collaborative Filtering* adalah model rekomendasi klasik, yang banyak digunakan di layanan *Web* dan *IoT*. *Collaborative Filtering* Tradisional menggunakan *item* yang direkomendasikan oleh pengguna yang serupa (misalnya, preferensi serupa), yang dapat diklasifikasikan ke dalam algoritma



berbasis ketetanggaan (*neighborhood based*) dan berbasis model (*model based*). Pendekatan berbasis ketetanggaan (*neighborhood-based approach*) juga dikenal sebagai pendekatan berbasis memori (*memory based*), menggunakan teknik statistik untuk membangun hubungan ketetanggaan antara pengguna (*user based CF*) atau antar *item* (*item based CF*) (Cui dkk., 2020).

Ada dua jenis *CF* yaitu *Memory-Based CF* dan *Model-Based CF*. *Memory-Based CF* menggunakan interaksi pengguna dengan *item* untuk menghitung kesamaan antar pengguna atau *item* sehingga dapat merekomendasikan *item* yang sesuai dengan preferensi yang didapat, sementara *Model-Based CF* menggunakan model matematis atau algoritma *machine learning* untuk membuat rekomendasi (Prayoga & Kusnawi, 2022). *Model-Based CF* memiliki beberapa keunggulan dibandingkan dengan *Memory-Based CF*, seperti memiliki *scalability* yang lebih baik dan dapat menangani masalah *sparsity* data dengan lebih baik. Dalam *Model-Based CF*, algoritma *Singular Value Decomposition* (SVD) sering digunakan untuk menganalisis interaksi pengguna dengan *item* dan membuat rekomendasi (W. Zhang dkk., 2019).

Sistem rekomendasi film sering menghadapi masalah *sparsitas* pada matriks *rating* pengguna-film, di mana sebagian besar entri matriks kosong atau tidak ada *rating* yang diberikan untuk banyak film. Hal ini menyulitkan untuk menghasilkan rekomendasi yang akurat dan personal. Namun, terdapat teknik yang dapat digunakan untuk mengatasi masalah ini yaitu algoritma SVD. Dengan menerapkan SVD, sistem dapat mengurangi dimensi matriks dan menghasilkan representasi latent yang lebih lebih ringkas tetapi tetap informatif. Dengan

demikian, rekomendasi dapat dibuat berdasarkan fitur tersembunyi atau preferensi yang mendasari perilaku pengguna (Wibisono dkk., 2021).

Oleh karena itu, penelitian ini bertujuan untuk mengevaluasi efektivitas *Model-Based Collaborative Filtering* dengan menggunakan algoritma *Singular Value Decomposition* (SVD) pada sistem rekomendasi film. Penelitian ini diharapkan dapat memberikan hasil rekomendasi film dan menyediakan solusi alternatif yang lebih akurat dan personal bagi pengguna.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, rumusan masalah yang dapat diambil dalam penelitian ini adalah:

1. Bagaimana mengimplementasikan algoritma *Singular Value Decomposition* (SVD) dalam sistem rekomendasi film menggunakan teknik *Model-Based Collaborative Filtering*.
2. Bagaimana nilai akurasi menggunakan *Mean Absolute Error* pada algoritma *Singular Value Decomposition* (SVD) dalam memberikan rekomendasi film.

## 1.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah:

- 1 Mengimplementasikan algoritma *Singular Value Decomposition* (SVD) menggunakan teknik *Model-Based Collaborative Filtering* dalam sistem rekomendasi film.

- 2 Mengetahui nilai akurasi menggunakan *Mean Absolute Error* pada algoritma *Singular Value Decomposition* (SVD) dalam memberikan rekomendasi film.

#### 1.4 Manfaat Penelitian

Manfaat yang diperoleh dari penelitian ini adalah:

1. Memberikan rekomendasi pada pengguna mengenai film yang akan ditonton selanjutnya.
2. Memberikan pengetahuan mengenai implementasi algoritma *Singular Value Decomposition* (SVD) dalam sistem rekomendasi film menggunakan teknik *Model-Based Collaborative Filtering*.
3. Meningkatkan pengalaman pengguna dalam menemukan film yang sesuai dengan preferensi mereka.

#### 1.5 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah:

1. Data yang digunakan dalam penelitian ini adalah data sekunder yang diambil dari sumber terbuka pada situs *web Kaggle*.
2. Teknik yang digunakan adalah *model-based collaborative filtering*.
3. Untuk Evaluasi model menggunakan teknik RMSE dan MAE.

## BAB II

### TINJAUAN PUSTAKA

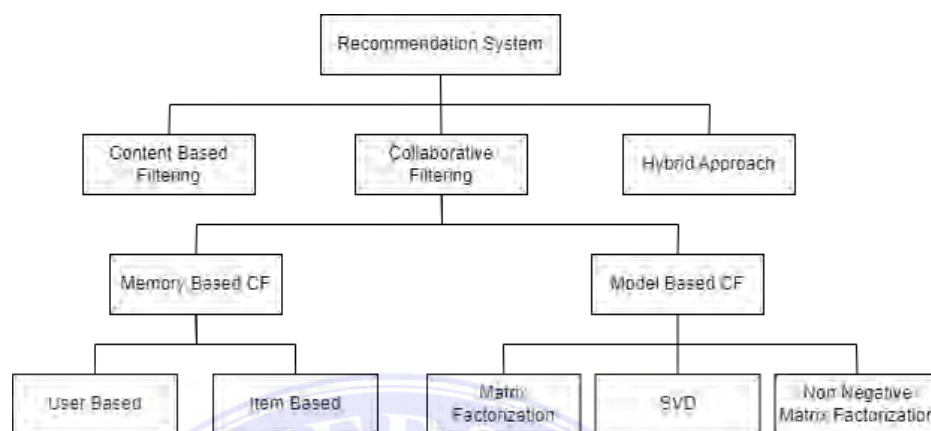
#### 2.1 Sistem Rekomendasi

Sistem rekomendasi adalah sistem atau aplikasi yang dibuat untuk dapat menyediakan dan memberikan rekomendasi dari suatu *item* untuk membuat suatu keputusan yang diinginkan oleh pengguna sistem (Februariyanti dkk., 2021). Sistem rekomendasi membantu pengguna untuk menemukan dan memilih *item* (misalnya buku, film, barang, toko, restoran dan sebagainya) dari sejumlah besar data yang tersedia di *web* atau sumber informasi elektronik lain. Dengan menggunakan *dataset* yang besar dan deskripsi kebutuhan pengguna, sistem merekomendasikan beberapa *item* kepada pengguna yang sesuai dengan deskripsi tersebut.

Menurut Putri dkk (2022) pada Literature Review: Sistem Rekomendasi untuk Buku dan Film, dalam situasi ketika para pengguna merasa bingung dalam memilih *item* yang sesuai dengan preferensi mereka, sistem rekomendasi dapat memberikan solusi. Dengan menggunakan data pengguna yang telah diolah, sistem rekomendasi dapat memberikan rekomendasi yang spesifik dan personal yang lebih cocok dengan preferensi dan kesukaan pengguna. Oleh karena itu, sistem rekomendasi sangat membantu pengguna dalam memilih *item* yang tepat dan sesuai dengan keinginan mereka. Dengan demikian, pengguna dapat menghemat waktu dan usaha dalam mencari judul yang cocok dengan preferensi mereka.

Sistem rekomendasi bertujuan untuk memperkirakan preferensi pengguna terhadap suatu *item* dan memberikan rekomendasi secara proaktif terhadap barang yang mungkin disukai oleh pengguna. Model rekomendasi umumnya

dikelompokkan menjadi tiga kategori yaitu *collaborative filtering*, *content based filtering* dan *hybrid filtering* (Zhang dkk., 2019).



Gambar 2. 1 Klasifikasi sistem rekomendasi

## 2.2 Collaborative Filtering

Menurut (Hadi dkk., 2020) Metode yang umum digunakan dalam pembuatan sistem rekomendasi adalah *Collaborative Filtering*. *Collaborative Filtering* memberikan rekomendasi untuk suatu *item* dengan cara mencari kemiripan antara preferensi *user* dan preferensi *user* lain terhadap *item* tersebut, atau mencari kesamaan antara *item* tersebut dengan *item* lain. Metode *Collaborative filtering* adalah suatu proses dalam mengevaluasi *item* atau penyaringan *item* dengan menggunakan persepsi atau opini orang lain. *Collaborative filtering* melakukan proses penyaringan terhadap semua pengguna untuk mendapatkan informasi pengguna dalam memberikan suatu rekomendasi. Dalam proses penyaringan data *Collaborative filtering* bekerja berdasarkan kemiripan karakteristik pengguna yang nantinya akan mampu memberikan suatu informasi yang baru kepada pengguna. Hal ini dikarenakan sistem akan memberikan informasi didasarkan pada pola dalam satu kelompok pengguna yang hampir mirip atau sama. Dibeberapa anggota kelompok dengan peminatan yang berbeda

kemungkinan akan memberikan informasi yang baru yang kemungkinan dapat bermanfaat bagi anggota kelompok lainnya (Februariyanti dkk., 2021).

### 2.2.1 *Model-Based Collaborative Filtering*

*Model-Based Collaborative Filtering* adalah metode rekomendasi yang menggunakan model matematis untuk memprediksi preferensi pengguna terhadap *item* tertentu. Ini berbeda dari metode lain seperti *Memory-Based Collaborative Filtering*, yang memanfaatkan interaksi pengguna dan *rating* sebelumnya untuk menentukan rekomendasi. *Model-Based Collaborative Filtering* diperkenalkan untuk mengatasi tantangan yang dihadapi oleh *Memory-Based Collaborative Filtering*. Teknik ini menemukan pola penilaian dari data historis dan memberikan rekomendasi yang sangat akurat dan efektif dari beberapa data sampel. Model pengelompokan dan faktorisasi matriks adalah teknik yang biasa digunakan dalam pendekatan berbasis model. SVD, SVD++, dan ALS adalah teknik Faktorisasi Matriks yang populer dan telah digunakan dalam sistem rekomendasi *Netflix Price Challenge* (Bhalse & Thakur, 2021).

Pada penelitian *Singular Value Decomposition Based Recommendation Using Imputed Data* yang dilakukan oleh Yuan dkk., (2019) menyebutkan *Model-based Collaborative Filtering* menghasilkan *rating* prediksi dengan model yang dilatih dari matriks *user-item*. *Model-based Collaborative Filtering* meliputi *Latent Semantic Model* (LFM), pLSA, LDA, *Latent Class Model*, *Latent Topic Model* dan *Matrix Factorization*. Metode-metode ini pada dasarnya sama, di antaranya *Matrix Factorization* merupakan metode yang paling umum digunakan dalam sistem rekomendasi. Metode *Model-Based Collaborative Filtering* menggunakan teknik faktorisasi matriks (*matrix factorization*) untuk mempelajari preferensi latent (tidak

terlihat) dari pengguna dan atribut latent dari *item*, berdasarkan *rating* yang dikenal. Setelah mempelajari preferensi dan atribut tersebut, metode ini bisa memprediksi *rating* yang tidak dikenal dengan mengalikan preferensi pengguna dan atribut *item*.

Keuntungan utama dari *Model-Based Collaborative Filtering* adalah skalabilitas dan efisiensi. Ini bisa menangani jumlah *item* dan pengguna yang besar dengan lebih baik daripada *Memory-based Collaborative Filtering*. Selain itu, metode ini juga bisa mengatasi *sparsity* (kekurangan data) dengan lebih baik, sehingga bisa membuat rekomendasi yang lebih akurat.

### 2.2.2 *Singular Value Decomposition (SVD)*

*Singular Value Decomposition (SVD)* adalah teknik faktorisasi matriks yang digunakan dalam banyak aplikasi termasuk kompresi data, pencarian informasi, dan sistem rekomendasi. Menurut Bhalse & Thakur (2021), SVD adalah salah satu teknik faktorisasi matriks yang digunakan dalam sistem rekomendasi *Model-Based Collaborative Filtering*. Teknik SVD mengurangi fitur *dataset* dan dimensi ruang dari  $N$  dimensi menjadi  $K$  dimensi (dimana  $K < N$ ). Hal ini dapat membantu mengatasi masalah kepadatan data dan mempercepat proses pemrosesan data dalam sistem rekomendasi. SVD dapat mengurai matriks menjadi tiga matriks, yaitu matriks pengguna-fitur, matriks nilai singular, dan matriks *item*-fitur. Nilai singular mewakili pentingnya atau kekuatan hubungan antara pengguna dan *item*. SVD biasanya digunakan dalam sistem rekomendasi untuk mengidentifikasi faktor laten yang berkontribusi pada preferensi pengguna untuk *item*.

Menurut Yuan dkk. (2019) SVD sering digunakan dalam *Matrix Factorization* pada sistem rekomendasi. Inti dari SVD adalah bahwa hubungan antara pengguna dan *item* dapat digabungkan secara *linear* oleh beberapa faktor.

Dengan kata lain, SVD mendekomposisi matriks *rating* menjadi dua matriks *low-rank* dan hasil kali keduanya adalah matriks *rating* yang mendekati nilai sebenarnya.

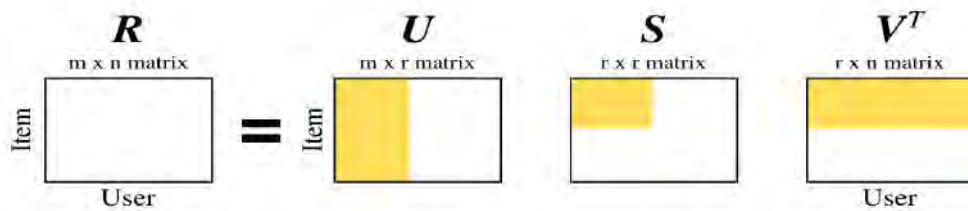
SVD adalah sebuah algoritma yang memecah sebuah matriks *A* menjadi sebuah aproksimasi dari matriks *A* asli. Secara matematis, SVD mendekomposisi matriks *A* menjadi sebuah matriks diagonal dan dua matriks unitaris. SVD dapat didefinisikan sebagai berikut:

$$R = U S V^T \tag{2.1}$$

keterangan:

- R** = matriks *rating* berukuran  $m \times n$ .
- U** = matriks singular kiri ortogonal berukuran  $m \times r$  yang mewakili matriks fitur pengguna.
- S** = matriks diagonal singular value berukuran  $r \times r$ .
- $V^T$**  = matriks singular kanan ortogonal berukuran  $r \times n$  yang menunjukkan kemiripan antara *item* dan faktor laten.

**U** mewakili berapa banyak pengguna yang "menyukai" setiap fitur, dan  **$V^T$**  mewakili seberapa relevan setiap fitur dengan setiap film. SVD mengurangi dimensi matriks *rating* **A** dengan mengekstrak faktor-faktor laten dan memetakan setiap pengguna ke setiap *item* dalam ruang laten berdimensi *r* (Bhalse & Thakur, 2021).



Gambar 2. 2 Singular value decomposition matrix



Matriks  $U$  dan  $V$  dapat diperoleh dengan melakukan *eigen*-dekomposisi (*eigenvector* dan *eigenvalue*) dari matriks  $RR^T$  atau  $R^TR$ , serta matriks singular  $S$  dapat diperoleh dari akar kuadrat dari *eigenvalues* dari matriks  $RR^T$  atau  $R^TR$  (Lidkk., 2019).

Untuk mencari matriks  $U$  dilakukan normalisasi terhadap *eigenvektor* yang telah didapat. Persamaan yang digunakan untuk melakukan normalisasi pada *eigenvektor* adalah:

$$v_{norm} = \frac{v}{\|v\|} \tag{2.2}$$

Dimana:

$v_{norm}$  = *eigenvector* yang dinormalisasi

$v$  = *eigenvector* asli

$\|v\|$  = modulus *eigenvector*  $v$ , yang didapat dengan persamaan:

$$\|v\| = \sqrt{v_1^2 + v_2^2 + \dots + v_n^2} \tag{2.3}$$

Dimana  $v_1^2 + v_2^2 + \dots + v_n^2$  adalah elemen dari *eigenvector*  $v$ .

Setelah mendapatkan matriks  $S$  dan matriks  $U$ , selanjutnya adalah mencari matriks  $V^T$ , untuk mencari matriks  $V^T$  dapat digunakan persamaan berikut:

$$V^T = S^{-1} \cdot U^T \cdot R \tag{2.4}$$

Dimana:

$S^{-1}$  = invers dari matriks  $S$

$U^T$  = transpose matriks  $U$

$R$  = matriks *rating* awal

Dalam menghitung nilai prediksi *rating* dapat digunakan persamaan berikut:

$$\hat{r}_{ui} = \mu + \sum_{j \in R(u)} S_j \cdot (U_{uj} \cdot V_{ij}) / \sum_{j \in R(u)} S_j \quad (2.5)$$

Keterangan:

$\hat{r}_{ui}$  = prediksi *rating* untuk *item*  $i$  oleh pengguna  $u$

$\mu$  = nilai rata-rata dari seluruh *rating* dalam matriks  $R$

$R(u)$  = himpunan *item* yang telah diberi *rating* oleh pengguna  $u$

$S_j$  = singular value ke- $j$  dalam matriks  $S$

$U_{uj}$  = elemen pada baris ke- $u$  dan kolom ke- $j$  dalam matriks  $U$

$V_{ij}$  = elemen pada baris ke- $i$  dan kolom ke- $j$  dalam matriks  $V^T$

### 2.3 Mean Absolute Error (MAE)

MAE (*Mean Absolute Error*) adalah salah satu metrik evaluasi yang umum digunakan dalam *machine learning* untuk mengukur seberapa dekat prediksi model dengan nilai aktual. MAE didefinisikan sebagai rata-rata selisih absolut antara nilai *rating* sebenarnya dan nilai *rating* yang diprediksi. Semakin rendah nilai MAE menunjukkan prediksi yang lebih baik (Kumar dkk., 2019). MAE didefinisikan sebagai berikut:

$$MAE = \frac{\sum_{i=0}^n |r_i - p_i|}{n} \quad (2.6)$$

Keterangan:

$r_i$  = *rating* yang diprediksi

$p_i$  = *rating* yang sebenarnya

$n$  = jumlah data

### 2.4 Root Mean Square Error (RMSE)

*Root Mean Square Error* sangat populer untuk menilai algoritma mesin pembelajaran, termasuk algoritma yang jauh lebih canggih dari regresi linier.

Nilai RMSE digunakan untuk membedakan kinerja model dalam periode kalibrasi dengan periode validasi serta untuk membandingkan kinerja model individual dengan model prediksi lainnya (Suwarno, 2022).

$$RMSE = \sqrt{\sum_{j=i}^n (y' - y)^2} \quad (2.7)$$

Keterangan:

$n$  = Jumlah data.

$e$  = Error.

$y'$  = Nilai output (prediksi)

$y$  = Nilai aktual.

## 2.5 Film

Film adalah bentuk hiburan yang menggabungkan audio, visual, dan narasi untuk menceritakan kisah atau mempresentasikan ide. Film merupakan audio visual yang memiliki banyak sekali genre, dari genre komedi, drama, horor, action dan masih banyak lagi lainnya (Sari dkk., 2020). Film dapat ditonton pada media elektronik seperti TV ataupun bioskop. Menurut Arfisko & Wibowo (2022) budaya menonton acara TV dan film sekarang ini dipermudah dengan adanya internet. Platform *streaming* seperti Netflix, HBO Max dan Disney+ memberikan lebih banyak fleksibilitas kepada pengguna untuk menonton acara TV dan film favorit mereka, kapan saja dan di perangkat apapun.

## 2.6 Penelitian Terdahulu

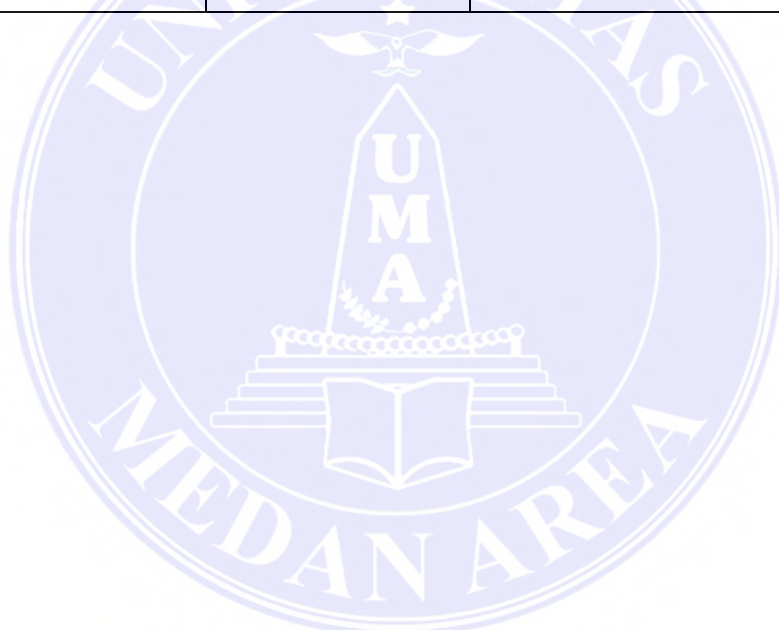
Penelitian terdahulu merujuk pada studi atau riset yang sudah dilakukan oleh peneliti sebelumnya dan terkait dengan topik yang akan diteliti oleh penulis. Penelitian terdahulu dapat memberikan kontribusi penting dalam penelitian yang

sedang dilakukan, baik sebagai sumber informasi maupun referensi. Oleh karena itu, penting bagi penulis untuk melakukan tinjauan literatur yang komprehensif dan mempertimbangkan penelitian terdahulu dalam merancang dan menjalankan penelitian yang akan dilakukan. Berikut adalah beberapa penelitian terdahulu tentang sistem rekomendasi dengan menggunakan teknik *collaborative filtering* yang menjadi acuan dan referensi penulis dalam melakukan penelitian ini:

Tabel 2. 1 Penelitian Terdahulu

No	Penulis	Judul	Hasil
1	Sageri Fikri Ramadhan, ZK Abdurahman Baizal, Rita Rismala (2020)	<i>Lodging Recommendations Using the SparkML Engine ALS and Surprise SVD</i>	Hasil nilai RMSE dari metode <i>Alternating Least Square</i> yang dijalankan oleh <i>SparkML Engine</i> mendapatkan skor 1,0125 dan metode <i>Singular Value Decomposition</i> yang dijalankan oleh <i>Surprise Engine</i> mendapatkan skor 0,3658.
2	Wervyan Shalannanda, Rafi Falih Mulia, Arief Insanu Muttaqien, Naufal Rafi Hibatullah, Annisabelia Firdaus ( 2022)	<i>Singular Value Decomposition Model Application for e- commerce Recommendation System</i>	SVD menghasilkan akurasi yang lebih baik dibandingkan dengan algoritma berbasis faktorisasi matriks lain. Nilai RMSE terendah yang diperoleh adalah 1,055586. Kemudian, peneliti melakukan optimasi dengan mengubah <i>hyperparameters</i> algoritma sehingga diperoleh model dengan nilai RMSE lebih baik yaitu 1,041784.
3	Yoan Amri Akbar, Z. K. A. Baizal, Agung Toto Wibowo (2021)	<i>Tourism Recommender System using Weighted Parallel Hybrid Method with Singular Value Decomposition</i>	Metode hybrid dengan teknik pembobotan memberikan akurasi prediksi yang lebih tinggi daripada saat menjalani metode sistem rekomendasi secara terpisah dengan nilai rata-rata 0,4140 (MAE), 0,2548 (MSE), 0,5006 (RMSE).

4	Ankur A. Ranjan, Amod Rai, Saiful Haque, Bhanu P. Lohani dan Pradeep K. Kushwaha (2019)	<i>An Approach for Netflix Recommendation System using Singular Value Decomposition</i>	SVD menghasilkan nilai <i>rating</i> terbaik di antara algoritma lain dengan RMSE sebesar 0,8190 pada data set 1M dan peningkatan performa sebesar 0,09936% dari data set 100K ke 1M.
5	Nisha Bhalse & Ramesh Thakur (2021)	<i>Algorithm for Movie Recommendation System using Collaborative Filtering</i>	SVD dapat menangani <i>dataset</i> yang besar dan kekosongan pada matriks <i>rating</i> . Algoritma yang digunakan dapat merekomendasikan top-n film sesuai dengan minat pengguna aktif.

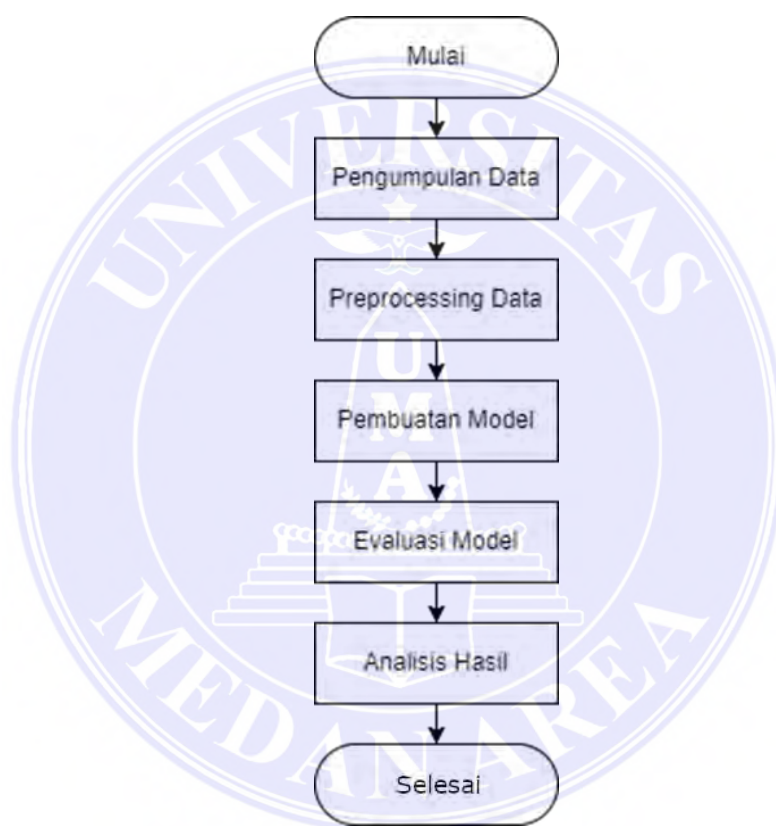


## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Tahapan Penelitian

Tahapan penelitian berisi langkah-langkah dilakukannya penelitian. Mulai dari pengumpulan data sampai melakukan analisa pada hasil penelitian. Berikut adalah gambar tahapan penelitian yang akan digunakan.



Gambar 3. 1 Tahapan Penelitian

#### 3.2 Pengumpulan Data

Data yang dipakai pada penelitian ini merupakan data sekunder yang didapat dari *website* penyedia data, yaitu *Kaggle*. Data yang digunakan pada penelitian ini, yaitu *Anime Recommendations Database* (<https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database>). *Dataset* ini berisi file *anime.csv* yang memuat data sebanyak 12.295 baris

7 kolom dan file *rating.csv* yang didalamnya terdapat 7,8 juta baris dengan 3 kolom. *Dataset* tersebut dipilih berdasarkan dengan kebutuhan data yang akan digunakan dalam penelitian. Setelah pemilihan *dataset*, dilakukan eksplorasi terhadap data untuk mempelajari atau memahami variabel-variabel pada data serta korelasi antar variabel. Data yang digunakan dalam penelitian ini dapat dilihat seperti tersaji pada Tabel 3.1 dan Tabel 3.2.

Tabel 3. 1 Data Anime.csv

No	Anime ID	Name	Genre	Type	Episodes	Rating	Members
1	5258	Hajime no Ippo: New Challenger	Comedy, Drama, Shounen, Sports	TV	26	8,75	88995
2	918	Gintama	Action, Comedy, Historical, Parody, Samurai, Sci-Fi, Shounen	TV	201	9,04	336376
3	2904	Code Geass: Hangyaku no Lelouch R2	Action, Drama, Mecha, Military, Sci-Fi, Super Power	TV	25	8,98	572888
4	28891	Haikyuu!! Second Season	Comedy, Drama, School, Shounen, Sports	TV	25	8,93	179342
5	199	Sen to Chihiro no Kamikakushi	Adventure, Drama, Supernatural	Movie	1	8,93	466254
6	23273	Shigatsu wa Kimi no Uso	Drama, Music, Romance, School, Shounen	TV	22	8,92	416397
7	24701	Hunter x Hunter (2011)	Action, Adventure, Shounen, Super Power	TV	148	9,13	425855
8	1	Cowboy Bebop	Action, Adventure, Comedy, Drama, Sci-Fi, Space	TV	26	8,82	486824
9	30276	One Punch Man	Action, Comedy, Parody, Sci-Fi, Seinen, Super Power, Supernatural	TV	12	8,82	552458
10	164	Mononoke Hime	Action, Adventure, Fantasy	Movie	1	8,81	339556

Pada tabel data *Anime.csv* dapat dilihat bahwa tabel ini memiliki 8 atribut atau kolom yaitu *Anime ID*, *Name*, *Genre*, *Type*, *Episodes*, *Rating* dan *Members*.

Tabel 3. 2 Data Rating.csv

No	User ID	Anime ID	Rating
1	3690	521	5
2	813	11111	7
3	4446	5525	6
4	5831	8861	8
5	4948	1218	6
6	3405	4224	10
7	2469	20021	7
8	5354	7110	9
9	6347	135	10
10	7988	8408	9

Pada tabel data *Rating.csv* dapat dilihat bahwa tabel ini memiliki 3 atribut atau kolom yaitu *User ID*, *Anime ID* dan *Rating*.

### 3.3 Preprocessing Data

Tahap ini dilakukan dengan melakukan pembersihan, pengolahan dan transformasi data yang diperlukan untuk membuat data menjadi lebih mudah diolah oleh model. Preprocessing data meliputi penghapusan data yang tidak diperlukan, penggantian nilai yang hilang atau tidak valid, dan integrasi data.

### 3.4 Pembuatan Model

Pada tahap ini data yang sudah dinormalisasi diubah menjadi matriks dengan kolom merepresentasikan setiap *item* dan baris merepresentasikan setiap *user*. Selanjutnya matriks tersebut dapat dipecah menjadi tiga matriks, yaitu matriks



$U$ , matriks  $S$ , dan matriks  $V^T$ . Berikut adalah sampel data yang akan digunakan dalam pembuatan model SVD:

Tabel 3. 3 Data Sample *User-Rating*

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
U1	8	5	6	3		5	6	7	9	
U2		9	10	8	5	9			4	8
U3	8	7	8	7	9	8	7	7		
U4	7	6		8			8	9	8	10
U5	9		8	6	8	9	7		5	7

Keterangan:

U1-U5 mempresentasikan *user* atau pengguna.

F1-F10 mempresentasikan film.

Nilai dalam tabel mempresentasikan *rating* yang diberikan *user* terhadap film.

Tahap pertama adalah mengubah data pada tabel 3.3 ke dalam bentuk matriks:

$$R = \begin{bmatrix} 8 & 5 & 6 & 3 & 0 & 5 & 6 & 7 & 9 & 0 \\ 0 & 9 & 10 & 8 & 5 & 9 & 0 & 0 & 4 & 8 \\ 8 & 7 & 8 & 7 & 9 & 8 & 7 & 7 & 0 & 0 \\ 7 & 6 & 0 & 8 & 0 & 0 & 8 & 9 & 8 & 10 \\ 9 & 0 & 8 & 6 & 8 & 9 & 7 & 0 & 5 & 7 \end{bmatrix}$$

Selanjutnya matriks  $R$  didekomposisikan menjadi 3 matriks, yaitu matriks  $U$ , matriks  $S$ , dan matriks  $V^T$ . Untuk mencari matriks  $U$ ,  $S$  dan  $V^T$  terlebih dahulu harus mencari nilai *eigenvalue* dan *eigenvector* dari matriks  $RR^T$ .

$$RR^T = \begin{bmatrix} 325 & 210 & 299 & 293 & 270 \\ 210 & 431 & 316 & 230 & 325 \\ 299 & 316 & 469 & 273 & 371 \\ 293 & 230 & 273 & 458 & 277 \\ 270 & 325 & 371 & 277 & 449 \end{bmatrix}$$

Nilai *eigenvalue* dan *eigenvector* dari matriks  $RR^T$  adalah:

$$v_1 \approx \begin{bmatrix} 7.587 \\ 1.447 \\ -5.191 \\ -3.549 \\ 1 \end{bmatrix}, \quad \lambda_1 \approx 59,037 \quad v_2 \approx \begin{bmatrix} -0.336 \\ -0.376 \\ -0.470 \\ 0.098 \\ 1 \end{bmatrix}, \quad \lambda_2 \approx 88,988$$

$$v_3 \approx \begin{bmatrix} 2.116 \\ -4.738 \\ 3.833 \\ -2.733 \\ 1 \end{bmatrix}, \quad \lambda_3 \approx 145,620 \quad v_5 \approx \begin{bmatrix} 0.813 \\ 0.888 \\ 1.022 \\ 0.888 \\ 1 \end{bmatrix}, \quad \lambda_5 \approx 1582,272$$

$$v_4 \approx \begin{bmatrix} -1.297 \\ 2.001 \\ 0.849 \\ -2.916 \\ 1 \end{bmatrix}, \quad \lambda_4 \approx 256,084$$

Keterangan:

$v$  : *eigenvector*

$\lambda$  : *eigenvalue*

Setelah mendapatkan nilai *eigenvalue*, maka didapat matriks  $S$  dengan nilai akar eigenvalue menjadi nilai matriks diagonal  $S$  berurutan dari yang terbesar ke yang terkecil.

$$S = \begin{bmatrix} \sqrt{1582,272} & 0 & 0 & 0 & 0 \\ 0 & \sqrt{256,084} & 0 & 0 & 0 \\ 0 & 0 & \sqrt{145,620} & 0 & 0 \\ 0 & 0 & 0 & \sqrt{88,988} & 0 \\ 0 & 0 & 0 & 0 & \sqrt{59,037} \end{bmatrix}$$

$$S = \begin{bmatrix} 39,777 & 0 & 0 & 0 & 0 \\ 0 & 16,281 & 0 & 0 & 0 \\ 0 & 0 & 12,067 & 0 & 0 \\ 0 & 0 & 0 & 9,433 & 0 \\ 0 & 0 & 0 & 0 & 7,685 \end{bmatrix}$$

Dengan menggunakan persamaan (2.2) dan (2.3) maka didapat nilai matriks  $U$  sebagai berikut:

$$U = \begin{bmatrix} -0,39288197 & 0,32511332 & -0,29895006 & 0,2757168 & 0,75799353 \\ -0,42930271 & -0,50161772 & 0,66943656 & 0,30829976 & 0,14451572 \\ -0,49364383 & -0,21272003 & -0,54165463 & 0,38571564 & -0,51855536 \\ -0,42913976 & 0,73114984 & 0,38617265 & -0,08030589 & -0,35451464 \\ -0,48321608 & -0,25070119 & -0,14129581 & -0,8207959 & 0,09990354 \end{bmatrix}$$

Menghitung matriks  $V^T$  menggunakan persamaan (2.4):

$$S^{-1} = \begin{bmatrix} 0,025 & 0 & 0 & 0 & 0 \\ 0 & 0,061 & 0 & 0 & 0 \\ 0 & 0 & 0,083 & 0 & 0 \\ 0 & 0 & 0 & 0,106 & 0 \\ 0 & 0 & 0 & 0 & 0,130 \end{bmatrix}$$

$$U^T = \begin{bmatrix} -0,392 & -0,429 & -0,493 & -0,429 & -0,483 \\ 0,325 & -0,501 & -0,212 & 0,731 & -0,250 \\ -0,298 & 0,669 & -0,541 & 0,386 & -0,141 \\ 0,275 & 0,308 & 0,385 & -0,080 & -0,820 \\ 0,757 & 0,144 & -0,518 & -0,354 & 0,099 \end{bmatrix}$$

$$R = \begin{bmatrix} 8 & 5 & 6 & 3 & 0 & 5 & 6 & 7 & 9 & 0 \\ 0 & 9 & 10 & 8 & 5 & 9 & 0 & 0 & 4 & 8 \\ 8 & 7 & 8 & 7 & 9 & 8 & 7 & 7 & 0 & 0 \\ 7 & 6 & 0 & 8 & 0 & 0 & 8 & 9 & 8 & 10 \\ 9 & 0 & 8 & 6 & 8 & 9 & 7 & 0 & 5 & 7 \end{bmatrix}$$

Dari hasil perkalian matriks  $S^{-1} \cdot U^T \cdot A$  didapat matriks  $V^T$  sebagai berikut:

$$V^T = \begin{bmatrix} -0,363 & -0,298 & -0,364 & -0,362 & -0,263 & -0,355 & -0,317 & -0,253 & -0,279 & -0,279 \\ 0,231 & 0,001 & -0,416 & -0,011 & -0,395 & -0,421 & 0,280 & 0,452 & 0,339 & 0,095 \\ -0,439 & 0,253 & -0,047 & 0,241 & -0,220 & -0,089 & -0,289 & -0,200 & 0,196 & 0,682 \\ -0,282 & 0,675 & 0,133 & 0,045 & -0,165 & -0,016 & -0,216 & 0,414 & -0,109 & -0,433 \\ 0,043 & -0,087 & 0,344 & -0,317 & -0,409 & 0,240 & -0,159 & -0,197 & 0,659 & -0,220 \end{bmatrix}$$

Setelah mendapatkan semua matriks yang dibutuhkan, selanjutnya adalah melakukan pengujian prediksi *rating*.

Tabel 3. 4 Contoh Prediksi *Rating*

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
U4	7	6	?	8	?	?	8	9	8	10

Pada tabel 3.4 terlihat *user-4* belum memberikan *rating* untuk semua film. *User-4* belum memberikan *rating* terhadap film-3, 5 & 6. Data *rating user-4* terhadap film-3 akan dijadikan data uji dalam prediksi *rating* menggunakan SVD.

Untuk menghitung nilai prediksi *rating* maka dapat menggunakan persamaan 2.5, Berikut ini proses menghitung nilai prediksi *rating* untuk *user-4* terhadap film-3:

Mencari nilai rata-rata seluruh *rating*:

$$\mu = 278 / 38 = 7,31$$

$$\begin{aligned} & \text{Hitung nilai } \sum(j \in R(u)) S_j \cdot (U_{uj} \cdot V_{ij}) \\ & = 39,777 \cdot (-0,429) \cdot (-0,364) + 16,281 \cdot 0,731 \cdot (-0,423) + 12,067 \\ & \quad \cdot 0,386 \cdot (-0,047) + 9,433 \cdot (-0,080) \cdot 0,133 + 7,685 \\ & \quad \cdot (-0,354) \cdot 0,344 \\ & = -0,078014835 \end{aligned}$$

$$\begin{aligned} & \text{Hitung nilai } \sum(j \in R(u)) S_j \\ & = 39,777 + 16,281 + 12,067 + 9,433 + 7,685 \\ & = 85,243 \end{aligned}$$

Hitung prediksi *rating*.

$$\begin{aligned} \hat{r}_{ui} & = \mu + \sum(j \in R(u)) S_j \cdot (U_{uj} \cdot V_{ij}) / \sum(j \in R(u)) S_j \\ & = 7,31 + (-0,078014835/85,243) \\ & = 7,3 \end{aligned}$$

Prediksi *rating* yang diberikan *user-4* terhadap film-3 adalah 7,3

### 3.5 Evaluasi Model

Setelah membuat model dan melakukan prediksi, tahap selanjutnya adalah mengevaluasi keakuratan model. Salah satu teknik evaluasi yang umum digunakan untuk mengevaluasi performa model dalam memprediksi nilai adalah *Mean Absolute Error* (MAE). MAE didefinisikan sebagai rata-rata selisih absolut antara nilai *rating* sebenarnya dan nilai *rating* yang diprediksi. Pada prediksi sebelumnya telah didapat prediksi pada film yang belum pernah diberi *rating*, sehingga nilai prediksi tersebut tidak dapat dibandingkan dengan nilai *rating* sesungguhnya. Jadi pada evaluasi model ini akan diasumsikan nilai *rating user-4* terhadap film-1 belum

pernah *dirating*. Pada Tabel 3.5, kolom berwarna merah dianggap masih kosong guna untuk mengevaluasi model yang telah dibuat.

Tabel 3.5 Evaluasi *Rating* prediksi

	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10
U4	7	6	?	8	?	?	8	9	8	10

Menghitung nilai prediksi *rating* untuk *user-4* terhadap film-1:

Mencari nilai rata-rata seluruh *rating*:

$$\mu = 271 / 37 = 7,32$$

Hitung nilai  $\sum(j \in R(u)) S_j \cdot (U_{uj} \cdot V_{ij})$

$$\begin{aligned}
 &= 39,777 \cdot (-0,429) \cdot (-0,363) + 16,281 \cdot 0,731 \cdot 0,235 + 12,067 \\
 &\quad \cdot 0,386 \cdot (-0,439) + 9,433 \cdot (-0,080) \cdot (-0,282) + 7,685 \\
 &\quad \cdot (-0,354) \cdot 0,043 \\
 &= 7,04
 \end{aligned}$$

Hitung nilai  $\sum(j \in R(u)) S_j$

$$\begin{aligned}
 &= 39,777 + 16,281 + 12,067 + 9,433 + 7,685 \\
 &= 85,243
 \end{aligned}$$

Hitung prediksi *rating*.

$$\begin{aligned}
 \hat{r}_{ui} &= \mu + \sum(j \in R(u)) S_j \cdot (U_{uj} \cdot V_{ij}) / \sum(j \in R(u)) S_j \\
 &= 7,32 + (7,04/85,243) \\
 &= 7,40
 \end{aligned}$$

Prediksi *rating* yang diberikan *user-4* terhadap film-1 adalah 7,40

Hasil prediksi *rating* yang akan diberikan *user-4* terhadap film-1 adalah 7,40 sedangkan *rating* aslinya adalah 7. Selisih inilah yang akan dihitung untuk mendapatkan seberapa besar nilai Mean Absolute Error.

$$MAE = \frac{|7.40 - 7|}{1} = 0,40$$

Berdasarkan perhitungan di atas, dapat dilihat bahwa nilai MAE dari prediksi *rating* *user-4* terhadap film-1 adalah sebesar 0,40.

### 3.6 Analisis Hasil

Setelah menghitung nilai *Mean Absolute Error* (MAE) dari model SVD yang telah dibuat, diperoleh nilai sebesar 0,40. Nilai MAE ini menunjukkan bahwa model memiliki kesalahan sebesar 0,40 dalam memprediksi *rating*. Meskipun nilai MAE ini tergolong rendah, namun tetap perlu dianalisis apakah nilai ini sudah cukup akurat atau masih perlu ditingkatkan.

Penggunaan *dataset* yang berkualitas akan meningkatkan kualitas hasil prediksi. *Dataset* yang berkualitas akan memberikan informasi yang lebih akurat dan representatif tentang data yang akan diprediksi oleh model. *Dataset* yang berkualitas biasanya memiliki jumlah data yang cukup banyak, sehingga model dapat menemukan pola yang lebih akurat, serta menghasilkan prediksi yang lebih akurat pula.

## BAB V

### KESIMPULAN DAN SARAN

#### 5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan dengan implementasi *Model-Based Collaborative Filtering* pada sistem rekomendasi film dengan menggunakan algoritma SVD berdasarkan *dataset anime.csv* dan *rating.csv* menghasilkan rekomendasi judul film berdasarkan *rating* yang telah diberikan *user* pada film sebelumnya. Berdasarkan pengujian yang telah dilakukan penggunaan algoritma SVD dengan parameter  $k = 3000$  dalam sistem rekomendasi film menghasilkan nilai evaluasi prediksi menggunakan RMSE senilai 0,4002 menunjukkan bahwa tingkat akurasi model yang terbentuk baik, sedangkan nilai MAE yang peroleh 0,1186 menunjukkan tingkat kesalahan dalam prediksi rendah.

#### 5.2 Saran

Untuk pengembangan penelitian yang telah dilakukan, maka penulis menyarankan untuk penelitian selanjutnya yang dapat dipertimbangkan, sebagai berikut :

1. Menerapkan metode regularisasi pada model SVD, seperti *Ridge Regression* atau *Lasso Regression*. Metode ini dapat membantu mengurangi *overfitting* dan meningkatkan generalisasi model, terutama ketika *dataset* memiliki banyak fitur atau variabel.
2. Penerapan *Hybrid Filtering*: Menggabungkan pendekatan *Collaborative Filtering* dengan *Content-Based Filtering* dapat meningkatkan kualitas rekomendasi. Pendekatan ini dapat memanfaatkan dua sumber informasi yaitu data peringkat pengguna dan fitur konten film.

## DAFTAR PUSTAKA

- Akbar, Y. A., Wibowo, A. T., & others. (2021). Tourism Recommender System using Weighted Parallel Hybrid Method with Singular Value Decomposition. *Indonesia Journal on Computing (Indo-JC)*, 6(2), 53–64.
- Arfisko, H. H., & Wibowo, A. T. (2022). Sistem Rekomendasi Film Menggunakan Metode Hybrid Collaborative Filtering Dan Content-Based Filtering. *EProceedings of Engineering*, 9(3).
- Bhalse, N., & Thakur, R. (2021). Algorithm For Movie Recommendation System Using Collaborative Filtering. *Materials Today: Proceedings*, xxxx, 1–6. <https://doi.org/10.1016/j.matpr.2021.01.235>
- Cui, Z., Xu, X., Fei, X. U. E., Cai, X., Cao, Y., Zhang, W., & Chen, J. (2020). Personalized Recommendation System Based on Collaborative Filtering for IoT Scenarios. *IEEE Transactions on Services Computing*, 13(4), 685–695.
- Februariyanti, H., Laksono, A. D., Wibowo, J. S., & Utomo, M. S. (2021). Implementasi Metode Collaborative Filtering Untuk Sistem Rekomendasi Penjualan Pada Toko Mebel. *Jurnal Khatulistiwa Informatika*, 9(1).
- Hadi, I., Santoso, L. W., & Tjondrowiguno, A. N. (2020). Sistem Rekomendasi Film menggunakan User-based Collaborative Filtering dan K-modes Clustering. *Jurnal Infra*, 3(1), 18–21. <http://publication.petra.ac.id/index.php/teknik-informatika/article/view/9800>
- Kumar, P., Kumar, V., & Thakur, R. S. (2019). A New Approach For Rating Prediction System Using Collaborative Filtering. *Iran Journal of Computer Science*, 2, 81–87.
- Larasati, F. B. A., & Februariyanti, H. (2021). Sistem Rekomendasi Product Emina Cosmetics Dengan Menggunakan Metode Content-Based Filtering. *Jurnal Manajemen Informatika Dan Sistem Informasi*, 4(1), 45–54.



- Li, X., Wang, S., & Cai, Y. (2019). *Tutorial: Complexity analysis of Singular Value Decomposition and its variants*. 1–12. <http://arxiv.org/abs/1906.12085>
- Prayoga, F. A., & Kusnawi, K. (2022). Smartphone Recommendation System Using Model-Based Collaborative Filtering Method. *Jurnal Teknik Informatika (Jutif)*, 3(6), 1613–1622.
- Putri, I. H., Nurakhmadyavi, S. M. K. H., & Wahyudi, E. E. (2022). Literature Review: Sistem Rekomendasi untuk Buku dan Film. *Prosiding Seminar Nasional Teknik Elektro, Sistem Informasi, Dan Teknik Informatika (SNESTIK)*, 1(1), 75–81.
- Ramadhan, S. F., Baizal, Z. K. A., & Rismala, R. (2020). *Lodging Recommendations Using the SparkML Engine ALS and Surprise SVD*. 4, 889–897. <https://doi.org/10.30865/mib.v>
- Ranjan, A. A., Rai, A., Haque, S., Lohani, B. P., & Kushwaha, P. K. (2019). An Approach For Netflix Recommendation System Using Singular Value Decomposition. *Journal of Computer and Mathematical Sciences*, 10(4), 774–779.
- Rifaldy, R., & Setiawan, E. B. (2022). Recommender System Movie Netflix using Collaborative Filtering with Weighted Slope One Algorithm in Twitter. *Building of Informatics, Technology and Science (BITS)*, 4(2), 500–506.
- Sari, K. R., Suharso, W., & Azhar, Y. (2020). Pembuatan Sistem Rekomendasi Film dengan Menggunakan Metode Item Based Collaborative Filtering pada Apache Mahout. *Jurnal Repositor*, 2(6), 767–774.
- Shalannanda, W., Mulia, R. F., Muttaqien, A. I., Hibatullah, N. R., & Firdaus, A. (2022). Singular value decomposition model application for e-commerce recommendation system: Aplikasi model dekomposisi nilai tunggal untuk sistem rekomendasi e-commerce. *JITEL (Jurnal Ilmiah Telekomunikasi, Elektronika, Dan Listrik Tenaga)*, 2(2), 103–110.

- Suwarno. (2022). *Pengembangan Pembangkit Listrik Hibrid pada Lampu Penerangan Jalan Melalui Pemanfaatan Radiasi Matahari dan Kecepatan Angin*. UMSU Press.
- Wibisono, C., Haryadi, L. S., Widyaya, J. E., & Liliawati, S. L. (2021). Sistem Rekomendasi Suku Cadang Berdasarkan Item Based Filtering. *Jurnal Teknik Informatika Dan Sistem Informasi*, 7(1), 10–19. <https://doi.org/10.28932/jutisi.v7i1.3036>
- Yuan, X., Han, L., Qian, S., Xu, G., & Yan, H. (2019). Singular Value Decomposition Based Recommendation Using Imputed Data. *Knowledge-Based Systems*, 163, 485–494. <https://doi.org/10.1016/j.knosys.2018.09.011>
- Zhang, S., Yao, L., Sun, A., & Tay, Y. (2019). Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.*, 52(1). <https://doi.org/10.1145/3285029>
- Zhang, W., Zhang, X., Wang, H., & Chen, D. (2019). A deep Variational Matrix Factorization Method for Recommendation on Large Scale Sparse Dataset. *Neurocomputing*, 334, 206–218. <https://doi.org/10.1016/j.neucom.2019.01.028>

## LAMPIRAN

### 1. Source code

#### 1. Pengujian Data Pada Google Colaboratory

```

#Import library
import pandas as pd
import numpy as np
import random
import time
from scipy.sparse.linalg import svds
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from math import sqrt

#Koneksi ke Google Drive
from google.colab import drive
drive.mount('/content/drive')

#Import dataset
film = pd.read_csv('/content/drive/MyDrive/Datasets/anime.csv')
ratings =
pd.read_csv('/content/drive/MyDrive/Datasets/rating.csv')

#Filter data
#mengisi nilai genre yang null dengan string kosong
film['genre'] = film['genre'].fillna('')

#filter genre yang eksplisit
film = film[~film['genre'].str.contains('Hentai', case=False,
regex=False)]
film = film[~film['genre'].str.contains('Yuri', case=False,
regex=False)]
film = film[~film['genre'].str.contains('Yaoi', case=False,
regex=False)]
film = film[~film['genre'].str.contains('Shounen Ai', case=False,
regex=False)]
film = film[~film['genre'].str.contains('Shoujo Ai', case=False,
regex=False)]

#hapus data dengan rating kosong
film = film.dropna(subset=['rating'])

#mengambil data dengan rating lebih besar dari 5
film = film[film['rating'] > 5]

#mengambil data dengan members lebih dari 1000
film = film[film['members'] > 1000]

#urutkan film dari jumlah members nya
film = film.sort_values('members')

```

```

#Hapus kolom yang tidak diperlukan
film_rating =
film.drop(['name', 'genre', 'type', 'episodes', 'members'], axis=1)

#Filter data rating
#merubah tipe data rating pada data menjadi numeric
ratings['rating'] = pd.to_numeric(ratings['rating'],
errors='coerce')

#filter data rating yang nilainya -1 dan lebih kecil dari 5
ratings = ratings[(ratings['rating'] != -1) & (ratings['rating'] >
5)]

#filter data user, hanya mengambil data user yang telah merating
lebih dari 100 film
ratings = ratings.groupby('user_id').filter(lambda x: len(x) >
100)

#mencari unique value dari kolom user_id (untuk menghitung jumlah
user)
unique_ids = ratings['user_id'].unique()
jumlah_user = len(unique_ids)

print('Jumlah User : ' + str(jumlah_user))

#Membuat data sampel acak dari data user sebanyak 10000 baris
num_samples = 10000
random_samples = random.sample(list(unique_ids), num_samples)
ratings = ratings[ratings['user_id'].isin(random_samples)]

#Membuat tabel baru rating film yang diberikan user
tabel_user_rating = pd.merge(ratings, film_rating, on='anime_id')
pivot_table = pd.pivot_table(tabel_user_rating, values='rating_x',
index='user_id', columns='anime_id').fillna(0)

#Membuat matrix SVD dari pivot tabel
# Mengubah pivot table menjadi matriks
matrix_R = pivot_table.values

# Melakukan SVD pada matriks
k=2500
start = time.time()
U, S, V = svds(matrix_R, k) # K adalah jumlah faktor latennya
end = time.time()

exc_time = end - start
print(f"Execution Time = {exc_time:.2f}s")

#Mencari nilai  $\mu$  (mu)
mu = np.mean(matrix_R[np.nonzero(matrix_R)])
print('Nilai rata-rata Rating : ' + str(mu))

#Membuat fungsi prediksi rating
def calculate_rating(matrix_R, user_id, item_id, S, U, V):

```

```

    Ru = matrix_R[user_id, :] # Nilai rating user u pada semua
    item
    R_u_indices = Ru.nonzero()[0] # Indeks item dengan nilai
    rating tidak nol
    R_u_indices = R_u_indices[R_u_indices < S.shape[0]] #
    Memastikan indeks tidak melebihi ukuran matriks S
    denominator = np.sum(S[R_u_indices]) # Penyebut pada rumus

    rating = 0
    for j in R_u_indices:
        if j < V.shape[1]: # Periksa apakah j tidak melebihi
            ukuran matriks V
                rating += S[j] * (U[user_id, j] * V[j, item_id]) /
            denominator

    return rating

#Membuat fungsi rekomendasi film untuk user tertentu berdasarkan
prediksi rating
# Contoh penggunaan untuk mencari rekomendasi untuk user_id = 29
(dari tabel vipot_table)
user_id = 73507
index_position = pivot_table.index.get_loc(user_id)
recommendations = []

for item_id in range(matrix_R.shape[1]):
    if matrix_R[index_position, item_id] == 0:
        rating = calculate_rating(matrix_R, index_position,
            item_id, S, U, V)
        rating += mu
        recommendations.append((item_id, rating))

# Mengurutkan rekomendasi berdasarkan rating secara menurun
recommendations.sort(key=lambda x: x[1], reverse=True)

# Batasi hasil rekomendasi hanya 10 item
recommendations = recommendations[:10]

# Konversi hasil rekomendasi menjadi DataFrame
recommendations_df = pd.DataFrame(recommendations,
    columns=['Item_ID', 'Predicted_Rating'])

# Mengambil informasi film dari tabel Film
film_info = film[['anime_id', 'name', 'rating']]

# Gabungkan tabel rekomendasi dengan informasi film berdasarkan
'anime_id'
recommendations_df = recommendations_df.merge(film_info,
    left_on='Item_ID', right_index=True)

# Menampilkan tabel hasil rekomendasi dengan judul kolom yang
sesuai
recommendations_df =
recommendations_df.rename(columns={'anime_id': 'ID Film', 'name':
    'Name', 'rating': 'Rating'})

```

```

# Simpan tabel hasil rekomendasi ke dalam variabel "Data"
Data = recommendations_df
Data[['ID Film', 'Name', 'Rating', 'Predicted_Rating']]

# Membuat fungsi mencari nilai RMSE
def rmse(prediction, ground_truth):
    prediction = prediction[ground_truth.nonzero()].flatten()
    ground_truth = ground_truth[ground_truth.nonzero()].flatten()
    return sqrt(mean_squared_error(prediction, ground_truth))

# Membuat fungsi mencari nilai MAE
def mae(prediction, ground_truth):
    prediction = prediction[ground_truth.nonzero()].flatten()
    ground_truth = ground_truth[ground_truth.nonzero()].flatten()
    return mean_absolute_error(prediction, ground_truth)

# Mengisi matriks diagonal S
s_diag_matrix=np.diag(S)

# Menghitung nilai RMSE dari hasil prediksi
X_pred = np.dot(np.dot(U, s_diag_matrix), V)
print('RMSE: ' + str(rmse(X_pred, matrix_R)))

# Menghitung nilai MAE dari hasil prediksi
X_pred_mae = np.dot(np.dot(U, s_diag_matrix), V)
print('MAE: ' + str(mae(X_pred_mae, matrix_R)))

```

## 2. Membuat API SVD dengan Flask

```

# Import library
from flask import Flask, jsonify
import pandas as pd
import numpy as np
from scipy.sparse.linalg import svds
from scipy import linalg
from flask_sqlalchemy import SQLAlchemy
from sqlalchemy import text
from flask_cors import CORS
import pickle

app = Flask(__name__)
CORS(app)
# Konfigurasi database
app.config['SQLALCHEMY_DATABASE_URI'] =
'mysql://root:@localhost/sistem_rekomendasi'
app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
db = SQLAlchemy(app)

# Membuat fungsi prediksi rating
def calculate_rating(matrix_R, user_id, item_id, S, U, V):
    Ru = matrix_R[user_id, :]

```

```

R_u_indices = Ru.nonzero()[0]
R_u_indices = R_u_indices[R_u_indices < S.shape[0]]
denominator = np.sum(S[R_u_indices])

rating = 0
for j in R_u_indices:
    if j < V.shape[1]:
        rating += S[j] * (U[user_id, j] * V[j, item_id]) /
denominator

return rating

# Membuat route generate model
@app.route('/generate-model', methods=['GET'])
def generate_model():
    try:
        query = db.session.execute(text('SELECT * FROM ratings'))
        rows = query.fetchall()
        df_user_rating = pd.DataFrame(rows, columns=query.keys())
        pivot_table = pd.pivot_table(df_user_rating,
values='rating', index='user_id', columns='film_id').fillna(0)
        matrix_R = pivot_table.values
        U, S, V = linalg.svd(matrix_R, full_matrices=False)

        # Menyimpan matrix_R, pivot_table, U, S, V ke dalam file
        .pkl

        with open('data.pkl', 'wb') as f:
            pickle.dump((matrix_R, pivot_table, U, S, V), f)

        return jsonify({'success': 'Matrix R and SVD model
generated and saved successfully'})
    except Exception as e:
        print(f"Error executing SQL query: {e}")
        return jsonify({'error': 'An error occurred while
generating the matrix and SVD model'})

# Membuat route rekomendasi untuk user
@app.route('/recommendations/<int:user_id>', methods=['GET'])
def get_recommendations(user_id):
    try:
        # Load matrix_R, pivot_table, U, S, V from .pkl file
        with open('data.pkl', 'rb') as f:
            matrix_R, pivot_table, U, S, V = pickle.load(f)

        index_position = pivot_table.index.get_loc(user_id)
        recommendations = []

        for item_index in range(matrix_R.shape[1]):
            if matrix_R[index_position, item_index] == 0:
                rating = calculate_rating(matrix_R,
index_position, item_index, S, U, V)
                item_id = pivot_table.columns[item_index]
                recommendations.append((item_id, rating))

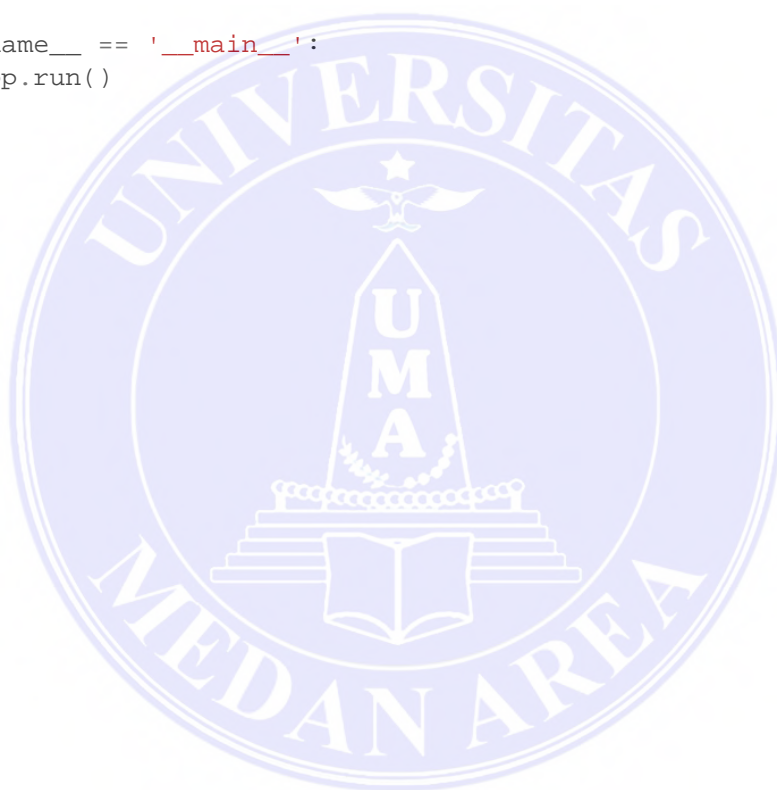
```

```
recommendations.sort(key=lambda x: x[1], reverse=True)
recommendations = recommendations[:10]

# Mengkonversi recommendations menjadi dictionary dengan
item_id sebagai string
recommendations_dict = [{'item_id': str(item_id),
'rating': rating} for item_id, rating in recommendations]

return jsonify({'recommendations': recommendations_dict})
except Exception as e:
print(f"Error executing SQL query: {e}")
return jsonify({'error': 'An error occurred while fetching
recommendations'}), 500

if __name__ == '__main__':
app.run()
```





## 2. Sample Data

ID Anime	Title	Title English	Genre	Score
51535	Shingeki no Kyojin: The Final Season - Kanketsu-hen	Attack on Titan: Final Season - The Final Chapters	Action, Drama	9.22
5114	Fullmetal Alchemist: Brotherhood	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy	9.11
41467	Bleach: Sennen Kessen-hen	Bleach: Thousand-Year Blood War	Action, Adventure, Fantasy	9.1
9253	Steins;Gate	Steins;Gate	Drama, Sci-Fi, Suspense	9.08
28977	Gintama°	Gintama Season 4	Action, Comedy, Sci-Fi	9.07
43608	Kaguya-sama wa Kokurasetai: Ultra Romantic	Kaguya-sama: Love is War - Ultra Romantic	Comedy, Romance	9.07
39486	Gintama: The Final	Gintama: The Very Final	Action, Comedy, Drama, Sci-Fi	9.05
38524	Shingeki no Kyojin Season 3 Part 2	Attack on Titan Season 3 Part 2	Action, Drama	9.06
9969	Gintama'	Gintama Season 2	Action, Comedy, Sci-Fi	9.04
11061	Hunter x Hunter (2011)	Hunter x Hunter	Action, Adventure, Fantasy	9.04
820	Ginga Eiyuu Densetsu	Legend of the Galactic Heroes	Drama, Sci-Fi	9.03
15417	Gintama': Enchousen	Gintama: Enchousen	Action, Comedy, Sci-Fi	9.03
42938	Fruits Basket: The Final	Fruits Basket: The Final Season	Drama, Romance, Supernatural	9.01
34096	Gintama.	Gintama Season 5	Action, Comedy, Sci-Fi	8.99
52198	Kaguya-sama wa Kokurasetai: First Kiss wa Owaranai	Kaguya-sama: Love is War - The First Kiss That Never Ends	Comedy, Drama, Romance	8.97
35180	3-gatsu no Lion 2nd Season	March Comes In Like A Lion 2nd Season	Drama, Slice of Life	8.94
4181	Clannad: After Story	Clannad: After Story	Drama, Romance, Supernatural	8.94
918	Gintama	Gintama	Action, Comedy, Sci-Fi	8.94

28851	Koe no Katachi	A Silent Voice	Award Winning, Drama	8.94
47917	Bocchi the Rock!	Bocchi the Rock!	Comedy	8.92
15335	Gintama Movie 2: Kanketsu-hen - Yorozuya yo Eien Nare		Action, Comedy, Sci-Fi	8.92
2904	Code Geass: Hangyaku no Lelouch R2	Code Geass: Lelouch of the Rebellion R2	Action, Award Winning, Drama, Sci-Fi	8.91
37987	Violet Evergarden Movie	Violet Evergarden the Movie	Award Winning, Drama, Fantasy	8.91
35247	Owarimonogatari 2nd Season	Owarimonogatari Second Season	Comedy, Mystery, Supernatural	8.89
19	Monster	Monster	Drama, Mystery, Suspense	8.86
37491	Gintama.: Shirogane no Tamashii-hen - Kouhan-sen	Gintama.: Silver Soul Arc - Second Half War	Action, Comedy, Sci-Fi	8.88
32281	Kimi no Na wa.	Your Name.	Award Winning, Drama, Supernatural	8.85
47778	Kimetsu no Yaiba: Yuukaku-hen	Demon Slayer: Kimetsu no Yaiba Entertainment District Arc	Action, Fantasy	8.82
36838	Gintama.: Shirogane no Tamashii-hen	Gintama.: Silver Soul Arc	Action, Comedy, Sci-Fi	8.81
40682	Kingdom 3rd Season	Kingdom: Season 3	Action	8.81
40028	Shingeki no Kyojin: The Final Season	Attack on Titan: Final Season	Action, Drama	8.81
37510	Mob Psycho 100 II	Mob Psycho 100 II	Action, Comedy, Supernatural	8.8
31758	Kizumonogatari III: Reiketsu-hen	Kizumonogatari Part 3: Cold-Blooded	Action, Mystery, Supernatural	8.79
199	Sen to Chihiro no Kamikakushi	Spirited Away	Adventure, Award Winning, Supernatural	8.78
17074	Monogatari Series: Second Season	Monogatari Series: Second Season	Comedy, Mystery, Romance, Supernatural	8.77

32935	Haikyuu!! Karasuno Koukou vs. Shiratorizawa Gakuen Koukou	Haikyuu!! 3rd Season	Sports	8.78
48583	Shingeki no Kyojin: The Final Season Part 2	Attack on Titan: The Final Season Part 2	Action, Drama	8.77
1	Cowboy Bebop	Cowboy Bebop	Action, Award Winning, Sci-Fi	8.75
44074	Shiguang Dailiren	Link Click	Drama, Supernatural	8.75
263	Hajime no Ippo	Fighting Spirit	Sports	8.75
50160	Kingdom 4th Season	Kingdom: Season 4	Action	8.75
50172	Mob Psycho 100 III	Mob Psycho 100 III	Action, Comedy, Supernatural	8.74
24701	Mushishi Zoku Shou 2nd Season	Mushi-shi: Next Passage Part 2	Adventure, Fantasy, Mystery, Slice of Life, Supernatural	8.73
33095	Showa Genroku Rakugo Shinjuu: Sukeroku Futatabi-hen	Descending Stories: Showa Genroku Rakugo Shinju	Drama	8.73
37521	Vinland Saga		Action, Adventure, Drama	8.73
49387	Vinland Saga Season 2	Vinland Saga Season 2	Action, Adventure, Drama	8.73
45576	Mushoku Tensei: Isekai Ittara Honki Dasu Part 2	Mushoku Tensei: Jobless Reincarnation Part 2	Drama, Fantasy, Ecchi	8.71
44	Rurouni Kenshin: Meiji Kenkaku Romantan - Tsuioku-hen	Samurai X: Trust and Betrayal	Action, Drama, Romance	8.71
48569	86 Part 2	86 Eighty-Six Part 2	Action, Drama, Sci-Fi	8.71

### 3. Plagiasi

**turnitin** Similarity Report ID: old:29477:46437630

PAPER NAME	AUTHOR
Muhammad_Rizky_188160044.docx	Muhammad Rizky Sitorus

---

WORD COUNT	CHARACTER COUNT
5549 Words	35201 Characters

PAGE COUNT	FILE SIZE
40 Pages	2.1MB

SUBMISSION DATE	REPORT DATE
Nov 10, 2023 12:15 PM GMT+7	Nov 10, 2023 12:16 PM GMT+7

---

- **20% Overall Similarity**  
The combined total of all matches, including overlapping sources, for each database.
  - 17% Internet database
  - 3% Publications database
  - Crossref database
  - Crossref Posted Content database
  - 12% Submitted Works database
- **Excluded from Similarity Report**
  - Small Matches (Less than 10 words)

Summary

#### 4. Surat Keterangan Pembimbing



## UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 ☎(061) 7366876, 7360166, 7364348, 7366781, Fax.(061) 7366996 Medan 20223  
Kampus II : Jalan Selayubi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225602, Fax. (061) 8226331 Medan 20122  
Website: www.teknik.uma.ac.id E-mail: univ\_medanarea@uma.ac.id

Nomor : 125/FT.6/01.10/II/2023  
Lamp : -  
Hal : **Pembimbing Tugas Akhir**

16 Februari 2023

Yth. Pembimbing Tugas Akhir  
**Rizki Muliono, S.Kom, M.Kom**  
di  
Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas :

N a m a : Muhammad Rizky Sitorus  
N P M : 188160044  
Jurusan : Teknik Informatika

Maka dengan hormat kami mengharapkan kesediaan saudara :

**Rizki Muliono, S.Kom, M.Kom** (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

**"Implementasi Model-Based Collaborative Filtering pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (Singular Value Decomposition)".**

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,

Dr. Rahmad Syah, S. Kom, M.Kom

## 5. Surat Ijin Melakukan Riset dari Fakultas Teknik



# UNIVERSITAS MEDAN AREA FAKULTAS TEKNIK

Kampus I : Jalan Kolam Nomor 1 Medan Estate/Jalan PBSI Nomor 1 ☎ (061) 7366878, 7360168, 7364348, 7366781, Fax (061) 7366998 Medan 20225  
Kampus II : Jalan Setabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A, ☎ (061) 8225802, Fax, (061) 8226331 Medan 20122  
Website: [www.teknik.uma.ac.id](http://www.teknik.uma.ac.id) E-mail: [univ\\_medanarea@uma.ac.id](mailto:univ_medanarea@uma.ac.id)

Nomor : 426 /FT.6/01.10/VI/2023 6 Juni 2023  
Lamp : -  
Hal : Penelitian Dan Pengambilan Data Tugas Akhir

Yth. Wakil Rektor Bid. Pengembangan SDM & Adm. Keuangan  
Jln. Kolam No.1  
Di  
Medan

Dengan hormat, kami mohon kesediaan ibu kiranya berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	N A M A	N P M	PRODI
1	Muhammad Rizky Sitorus	188160044	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir di **Laboratorium Komputer Program Studi Teknik Informatika Fakultas Teknik Universitas Medan Area.**

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan Ilmiah dan Skripsi, yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul :

**Implementasi Model-Based Collaborative Filtering pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (Singular Value Decomposition).**

Mohon kiranya tanggal Surat Izin Pengambilan Data Tugas Akhir agar disesuaikan dengan tanggal Terbitnya SK ini.

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.



Dr. Rahnawati S. Kom, M. Kom

Tembusan :  
1. Ka. BAMA  
2. Mahasiswa  
3. File

## 6. Surat Izin Riset dari BATRI



# UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360188, 7366878, 7364348 📠 (061) 7368012 Medan 20222  
Kampus II : Jalan Seiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 8225602 📠 (061) 8226331 Medan 20122  
Website: [www.uma.ac.id](http://www.uma.ac.id) E-Mail: [univ\\_medanarea@uma.ac.id](mailto:univ_medanarea@uma.ac.id)

Nomor : 999/UMA/B/01.7/1/2023  
Lamp. : 1 (satu) Berkas  
Hal : Izin Penelitian Dan Pengambilan Data Tugas Akhir.

09 Juni 2023.

Kepada Yth.  
Dekan Fakultas Teknik  
Universitas Medan Area  
di - M e d a n

Dengan hormat, sesuai dengan surat Dekan Fakultas Teknik Universitas Medan Area nomor 426/FT.6/01.10.VI/2023 tertanggal 06 Juni 2023 perihal Permohonan Izin Penelitian dan Pengambilan Data Tugas Akhir di Laboratorium Komputer Fakultas Teknik Universitas Medan Area oleh mahasiswa sebagai berikut :

Nama : Muhammad Rizky Sitorus  
No. Pokok Mahasiswa : 188160044  
Program Studi : Teknik Informatika  
Fakultas : Teknik

Pada prinsipnya disetujui yang bersangkutan melaksanakan pengambilan data di Laboratorium Komputer Fakultas Teknik Universitas Medan Area, untuk penunjang tugas akhir dengan judul Penelitian "**Implementasi Model-Based Collaborative Filtering Pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (Singular Value Decomposition)**".

Demikian surat ini diterbitkan untuk dipergunakan sebagaimana mestinya.

Wakil Rektor Bidang Pengembangan  
Sistem dan Administrasi Keuangan,



**Tembusan :**  
1. Mahasiswa Ybs  
2. File

## 7. Surat Selesai Riset



# UNIVERSITAS MEDAN AREA

Kampus I : Jalan Kolem Nomor 1 Medan Estate ☎ (061) 7360168, 7366878, 7364348 📠 (061) 7368012 Medan 20223  
Kampus II : Jalan Seiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 8225602 📠 (061) 8226331 Medan 20122  
Website: [www.uma.ac.id](http://www.uma.ac.id) E-Mail: [univ\\_medanarea@uma.ac.id](mailto:univ_medanarea@uma.ac.id)

### SURAT KETERANGAN

Nomor : 1298/UMA/B/01.7/VII/2023

Rektor Universitas Medan Area dengan ini menerangkan bahwa :

Nama : Muhammad Rizky Sitorus  
No. Pokok Mahasiswa : 188160044  
Program Studi : Teknik Informatika  
Fakultas : Teknik

Benar telah selesai Pengambilan Data di Laboratorium Komputer Prodi Teknik Informatika Fakultas Teknik Universitas Medan Area dengan Judul "*Implementasi Model-Based Collaborative Filtering Pada Sistem Rekomendasi Film Menggunakan Algoritma SVD (Singular Value Decomposition).*"

Dan kami harapkan Data tersebut kiranya dapat membantu yang bersangkutan dalam penyusunan skripsi dan dapat bermanfaat bagi mahasiswa khususnya Fakultas Teknik.

Demikian surat ini diterbitkan untuk dapat digunakan seperlunya.

Medan, 18 Juni 2023.

An Rektor,  
Wakil Rektor Bidang Pengembangan SDM dan  
Administrasi Keuangan,

  
Suswati, MP

Tembusan :

1. Mahasiswa
2. File