



**OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN
METODE *HYBRID FILTERING***

SKRIPSI

Oleh:

SEPTA HENDRAWAN

20211000021

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS BUDDHI DHARMA
TANGERANG**

2025



**OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN
METODE *HYBRID FILTERING***

SKRIPSI

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana pada
Program Studi Teknik Informatika Fakultas Sains dan Teknologi**

**Universitas Buddhi Dharma
Jenjang Pendidikan Strata 1**

Oleh:

SEPTA HENDRAWAN

20211000021

FAKULTAS SAINS DAN TEKNOLOGI

**UNIVERSITAS BUDDHI DHARMA
TANGERANG**

2025

LEMBAR PERNYATAAN KEASLIAN SKRIPSI

Yang bertanda tangan di bawah ini.

NIM : 20211000021
Nama : Septa Hendrawan
Jenjang Studi : Strata 1
Program Studi : Teknik Informatika
Peminatan : Data Science

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapat gelar akademik Sarjana atau kelengkapan studi, baik di Universitas Buddhi Dharma maupun di Perguruan Tinggi lainnya.
2. Skripsi ini saya buat sendiri tanpa bantuan dari pihak lain, kecuali arahan dosen pembimbing.
3. Dalam Skripsi ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dan dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan daftar pustaka.
4. Dalam Skripsi ini tidak terdapat pemalsuan (kebohongan), seperti buku, artikel, jurnal, data sekunder, pengolahan data, dan pemalsuan tanda tangan dosen atau Ketua Program Studi Universitas Buddhi Dharma yang dibuktikan dengan keasliannya.
5. Lembar pernyataan ini saya buat dengan sesungguhnya, tanpa paksaan dan apabila dikemudian hari atau pada waktu lainnya terdapat penyimpangan dan ketidak benaran dalam pernyataan ini, saya bersedia menerima sanksi akademik berupa pencabutan gelar akademik yang telah saya peroleh karena Skripsi ini serta sanksi lainnya sesuai dengan peraturan dan norma yang berlaku.

Tangerang, 06-08-2025

Yang membuat pernyataan,



Septa Hendrawan

20211000021

LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH

Yang bertanda tangan di bawah ini,

NIM	: 20211000021
Nama	: Septa Hendrawan
Jenjang Studi	: Strata 1
Program Studi	: Teknik Informatika
Peminatan	: Data Science

Dengan ini menyetujui untuk memberikan ijin kepada pihak Universitas Buddhi Dharma, Hak Bebas Royalti Non – Eksklusif (*Non-exclusive Royalty-Free Right*) atas karya ilmiah kami yang berjudul: **“OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN METODE *HYBRID FILTERING*”**, beserta alat yang diperlukan (apabila ada).

Dengan Hak Bebas Royalti Non – Eksklusif ini pihak Universitas Buddhi Dharma berhak menyimpan, mengalih-media atau format-kan, mengelolanya dalam pangkalan data (database), mendistribusikannya, dan menampilkan atau mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis pertama atau pencipta karya ilmiah tersebut. Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Universitas Buddhi Dharma, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini saya buat dengan sebenarnya.

Tangerang, 06-08-2025
Yang menandatangani pernyataan,



LEMBAR PENGESAHAN PEMBIMBING
OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN
METODE *HYBRID FILTERING*

Dibuat Oleh:

NIM : 20211000021

Nama : Septa Hendrawan

Telah disetujui untuk dipertahankan di hadapan Tim Penguji Ujian
Komprehensif

Program Studi Teknik Informatika

Peminatan Data Science

Tahun Akademik 2024/2025

Disahkan oleh,

Tangerang, 06-08-2025

Pembimbing,



Dr. Yakub, M.Kom, M.M.

NUPTK. 1836747648130172

LEMBAR PENGESAHAN SKRIPSI
OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN
METODE *HYBRID FILTERING*

Dibuat Oleh:

NIM : 20211000021

Nama : Septa Hendrawan

Telah disetujui untuk dipertahankan di hadapan Tim Penguji Ujian

Komprehensif

Program Studi Teknik Informatika

Peminatan Data Science

Tahun Akademik 2024/2025

Disahkan oleh,

Tangerang, 06-08-2025

Dekan,

Ketua Program Studi


Dr. Yakub, M.Kom., M.M.

NUPTK. 1836747648130172


Hartana Wijaya, M.Kom

NUPTK. 3844759660131192

LEMBAR PENGESAHAN TIM PENGUJI

Nama : Septa Hendrawan
NIM : 20211000021
Fakultas : Sains dan Teknologi
Judul Skripsi : OPTIMASI SISTEM REKOMENDASI FILM
MENGGUNAKAN METODE *HYBRID FILTERING*

Dinyatakan LULUS setelah mempertahankan di depan Tim Penguji pada hari Rabu, 06-08-2025

Nama penguji: Tanda Tangan:

Ketua Sidang : **Dram Renaldi, M.Kom**
NUPTK. 0443768669130322

Penguji I : **Hartana Wijaya, M.Kom**
NUPTK. 3844759660131192

Penguji II : **Dr. Yakub, M.Kom, M.M.**
NUPTK. 1836747648130172

Mengetahui,
Dekan Fakultas Sains dan Teknologi


Dr. Yakub, M.Kom., M.M.
NUPTK. 1836747648130172

KATA PENGANTAR

Dengan mengucapkan Puji Syukur kepada Tuhan Yang Maha Esa, yang telah memberikan Rahmat dan karunia-Nya kepada penulis sehingga dapat menyusun dan menyelesaikan Skripsi ini dengan judul **“OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN METODE *HYBRID FILTERING*”**. Tujuan utama dari pembuatan Skripsi ini adalah sebagai salah satu syarat kelengkapan dalam menyelesaikan program pendidikan Strata 1 Program Studi Teknik Informatika di Universitas Buddhi Dharma. Dalam penyusunan Skripsi ini penulis banyak menerima bantuan dan dorongan baik moril maupun materiil dari berbagai pihak, maka pada kesempatan ini penulis menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Ibu Dr. Limajatini, S.E., M.M., B.K.P, sebagai Rektor Universitas Buddhi Dharma
2. Bapak Dr. Yakub, S.Kom., M.Kom., M.M, sebagai Dekan Fakultas Sains dan Teknologi, dan juga sebagai pembimbing yang telah membantu dan memberikan dukungan serta harapan untuk menyelesaikan penulisan Skripsi ini
3. Bapak Hartana Wijaya, S.Kom., M.Kom, sebagai Ketua Program Studi Teknik Informatika
4. Orang tua dan keluarga yang selalu memberikan dukungan baik moril dan materiil.
5. Teman-teman yang selalu membantu dan memberikan semangat

Serta semua pihak yang terlalu banyak untuk disebutkan satu-persatu sehingga terwujudnya penulisan ini. Penulis menyadari bahwa penulisan Skripsi ini masih belum sempurna, untuk itu penulis mohon kritik dan saran yang bersifat membangun demi kesempurnaan penulisan di masa yang akan datang.

Akhir kata semoga Skripsi ini dapat berguna bagi penulis khususnya dan bagi para pembaca yang berminat pada umumnya.

Tangerang, 06-08-2025

Penulis

ABSTRAK

Transformasi digital dalam industri hiburan mendorong kebutuhan akan sistem rekomendasi yang cerdas dan adaptif, khususnya untuk platform *streaming* berskala mikro. Penelitian ini bertujuan mengembangkan sistem rekomendasi film anime dengan pendekatan *Hybrid Filtering* melalui studi kasus simulatif pada platform *streaming* mikro *MJ CELL*. Metode yang digunakan menggabungkan *Collaborative Filtering* (CF) berbasis algoritma *Singular Value Decomposition* (SVD) dan *Content-Based Filtering* (CBF) berbasis *TF-IDF* dan *Cosine Similarity*. *Dataset* sekunder diambil dari *MyAnimeList* melalui *Kaggle*, mencakup lebih dari 73.000 pengguna dan 12.000 judul anime. Pengembangan sistem mengikuti tahapan *CRISP-DM*, mulai dari *preprocessing* hingga evaluasi kinerja menggunakan metrik *RMSE*, *MAE*, *Precision@10*, *Recall@10*, dan *F1-Score@10*. Hasil evaluasi menunjukkan bahwa model CF memberikan akurasi prediktif terbaik (*RMSE* = 1,1488; *Precision@10* = 81,22%), sementara model *hybrid* menghasilkan keseimbangan optimal antara akurasi dan cakupan rekomendasi (*RMSE* = 1,1490; *F1@10* = 0,6059). Sistem diimplementasikan dalam bentuk prototipe web berbasis *Flask*, lengkap dengan fitur autentikasi pengguna, pemberian *rating*, dan tampilan rekomendasi personal. Penelitian ini menyimpulkan bahwa pendekatan *Hybrid Filtering* efektif untuk meningkatkan kualitas rekomendasi dan relevan diterapkan pada platform *streaming* simulatif berskala kecil yang memiliki keterbatasan data pengguna.

Kata kunci: *Collaborative Filtering*, *Content-Based Filtering*, *Hybrid Filtering*, Rekomendasi Anime, Sistem Cerdas

ABSTRACT

The digital transformation in the entertainment industry has increased the demand for intelligent and adaptive recommendation systems, especially for micro-scale streaming platforms. This study aims to develop an anime recommendation system using a Hybrid Filtering approach through a simulated case study on a micro streaming platform, MJ CELL. The method combines Collaborative Filtering (CF) based on the Singular Value Decomposition (SVD) algorithm and Content-Based Filtering (CBF) utilizing TF-IDF and Cosine Similarity. The system uses a secondary dataset from MyAnimeList available on Kaggle, consisting of more than 73,000 users and 12,000 anime titles. The development follows the CRISP-DM methodology, including preprocessing, model training, and performance evaluation using RMSE, MAE, Precision@10, Recall@10, and F1-Score@10. Results show that the CF model yields the best predictive accuracy (RMSE = 1.1488; Precision@10 = 81.22%), while the hybrid model achieves optimal balance between accuracy and recommendation coverage (RMSE = 1.1490; F1@10 = 0.6059). The system was implemented as a web-based prototype using Flask, equipped with user authentication, rating functionality, and personalized recommendations. This study concludes that the Hybrid Filtering approach is effective in enhancing recommendation quality and is suitable for simulated small-scale streaming platforms with limited user data.

Keywords: *Anime Recommendation, Collaborative Filtering, Content-Based Filtering, Hybrid Filtering, Intelligent System*

DAFTAR ISI

LEMBAR PERNYATAAN KEASLIAN SKRIPSI	i
LEMBAR PERSETUJUAN PUBLIKASI KARYA ILMIAH	ii
LEMBAR PENGESAHAN PEMBIMBING	iii
LEMBAR PENGESAHAN SKRIPSI.....	iv
LEMBAR PENGESAHAN TIM PENGUJI	v
KATA PENGANTAR.....	vi
ABSTRAK	vii
<i>ABSTRACT</i>	viii
DAFTAR ISI	ix
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
DAFTAR LAMPIRAN	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	3
1.3 Ruang Lingkup	3
1.4 Tujuan dan Manfaat	4
1.4.1 Tujuan.....	4
1.4.2 Manfaat	4
1.5 Sistematika Penulisan	5
BAB II LANDASAN TEORI	6
2.1 Konsep Dasar Sistem Rekomendasi	6
2.1.1 Sistem Rekomendasi.....	6
2.1.2 Optimasi.....	7
2.1.3 Film.....	7
2.1.4 <i>Anime</i>	8
2.2 Metode Algoritma.....	8
2.2.1 <i>Collaborative Filtering (CF)</i>	8
2.2.2 <i>Content-Based Filtering (CBF)</i>	9
2.2.3 <i>Popularity-Based Recommender</i>	11
2.2.4 <i>Hybrid Filtering</i>	11

2.3	Metodologi <i>CRIPS-DM</i>	13
2.4	<i>Tools</i> dan <i>Framework</i>	14
2.4.1	Bahasa Pemrogram <i>Python</i>	14
2.4.2	<i>Flask</i>	15
2.4.3	<i>SQLite</i>	15
2.4.4	<i>Surprise</i>	16
2.4.5	<i>Scikit-learn</i>	16
2.4.6	<i>NumPy</i>	16
2.4.7	<i>Pandas</i>	17
2.5	Evaluasi Sistem Rekomendasi	17
2.5.1	Evaluasi Prediksi <i>Rating</i>	17
2.5.2	Evaluasi Top-N Rekomendasi	19
2.5.3	Evaluasi Fungsional (<i>Black Box Testing</i>)	21
2.6	Tinjauan Studi	21
2.7	Kesimpulan Tinjauan Studi	30
BAB III METODOLOGI PENELITIAN		32
3.1	Analisa Kebutuhan	32
3.1.1	Kebutuhan Fungsional	33
3.1.2	Kebutuhan Non Fungsional	33
3.2	Kerangka Pemikiran	34
3.3	<i>Data Understanding</i>	35
3.3.1	<i>Anime.csv</i>	35
3.3.2	<i>Rating.csv</i>	36
3.3.3	<i>Merge Dataset</i>	37
3.4	<i>Pre-processing Data</i>	39
3.5	<i>Data Splitting</i>	42
3.6	<i>Model Training</i>	43
3.6.1	Alur Proses Pelatihan Model	43
3.6.2	<i>Hyperparameter Tuning</i>	43
3.6.3	<i>Collaborative Filtering</i>	45
3.6.4	<i>Content-Based Filtering</i>	47
3.6.5	<i>Hybrid Approach</i>	48
3.7	Metode Evaluasi	49
3.7.1	Evaluasi Akurasi Prediksi	49

3.7.2	Evaluasi Rekomendasi Top-N	50
3.8	Ruang Lingkup Pengembangan	50
3.8.1	Spesifikasi Perangkat Keras	50
3.8.2	Spesifikasi Perangkat Lunak	51
3.9	Perancangan Layar, Menu	52
3.9.1	Perancangan Layar, Menu	52
3.9.2	Perancangan <i>Database</i>	56
3.9.3	<i>Use Case Diagram</i>	58
BAB IV HASIL DAN PEMBAHASAN		59
4.1	Evaluasi Model	59
4.1.1	Evaluasi Model <i>Collaborative Filtering</i>	59
4.1.2	Evaluasi Model <i>Content-Based Filtering</i>	62
4.1.3	Evaluasi Model <i>Hybrid Filtering</i>	65
4.2	Komparasi Hasil Evaluasi Model	67
4.3	Implementasi <i>Website</i>	68
4.3.1	Pelatihan dan Penyimpanan Model <i>Hybrid</i>	69
4.3.2	Memuat Model & Logika <i>Hybrid</i>	69
4.3.3	<i>Endpoint</i> Rekomendasi	70
4.3.4	Validasi, Pra-proses, dan <i>Fallback CBF</i>	70
4.3.5	Integrasi <i>Front-end</i>	72
4.4	Tampilan Program	73
4.4.1	Tampilan Halaman Awal	73
4.4.2	Tampilan Halaman <i>Login</i>	74
4.4.3	Tampilan Halaman Registrasi	75
4.4.4	Tampilan Halaman Riwayat <i>Rating User</i>	75
4.4.5	Tampilan Halaman <i>Dashboard</i>	76
4.4.6	Tampilan Halaman <i>Detail Anime</i>	76
4.4.7	Tampilan Halaman <i>Profile</i> Pengguna	77
4.4.8	Tampilan Halaman <i>Dashboard Admin</i>	77
4.5	<i>Blackbox Testing</i>	78
4.6	Pembahasan	80
BAB V SIMPULAN DAN SARAN		83
5.1	Simpulan	83
5.2	Saran	84

DAFTAR PUSTAKA.....	85
DAFTAR RIWAYAT HIDUP.....	88
LAMPIRAN	89



DAFTAR GAMBAR

Gambar 2.1 Gambar <i>CRIPS-DM</i>	13
Gambar 3.1 Kerangka Pemikiran	34
Gambar 3.2 Alur Proses Pelatihan Model	43
Gambar 3.3 <i>Visual Studio Code</i>	51
Gambar 3.4 <i>Wireframe Landing Page</i>	52
Gambar 3.5 <i>Wireframe Login page</i>	52
Gambar 3.6 <i>Wireframe Sign up Page</i>	53
Gambar 3.7 <i>Wireframe User Rating History Page</i>	53
Gambar 3.8 <i>Wireframe Dashbord Page</i>	54
Gambar 3.9 <i>Wireframe Anime Detail Page</i>	55
Gambar 3.10 <i>Wireframe Profile Page</i>	55
Gambar 3.11 <i>Wireframe Admin Dashboard Page</i>	56
Gambar 3.12 Rancangan ERD Database.....	56
Gambar 3.13 <i>Use Case Diagram</i>	58
Gambar 4.1 Histogram Perbandingan Performa Model	67
Gambar 4.2 Tampilan Halaman Awal.....	74
Gambar 4.3 Tampilan Halaman <i>Login</i>	74
Gambar 4.4 Tampilan Halaman Registrasi	75
Gambar 4.5 Tampilan Halaman Riwayat <i>Rating User</i>	75
Gambar 4.6 Tampilan Halaman <i>Dashboard</i>	76
Gambar 4.7 Tampilan Halaman <i>Detail Anime</i>	76
Gambar 4.8 Tampilan Halaman Profile Pengguna.....	77
Gambar 4.9 Tampilan Halaman <i>Dashboard Anime</i>	77

DAFTAR TABEL

Tabel 3.1 Tabel Analisa Kebutuhan	32
Tabel 3.2 Informasi <i>Dataset anime.csv</i>	35
Tabel 3.3 Sampel <i>Dataset anime.csv</i>	36
Tabel 3.4 informasi <i>Dataset rating.csv</i>	37
Tabel 3.5 Sampel <i>Dataset rating.csv</i>	37
Tabel 3.6 Informasi <i>Dataset merge.csv</i>	38
Tabel 3.7 Sampel <i>Dataset merge.csv</i>	38
Tabel 3.8 Sampel <i>Dataset merge_clean.csv</i>	41
Tabel 3.9 Distribusi <i>Training dan Testing</i>	42
Tabel 3.10 Konfigurasi Parameter <i>RandomizedSearchCV</i> Model <i>SVD</i>	44
Tabel 3.11 Parameter <i>Grid</i> Model <i>SVD</i>	45
Tabel 3.12 Parameter <i>Grid</i> untuk <i>Content-Based Filtering</i> Model <i>TF-IDF</i>	45
Tabel 3.13 Hasil Penyaringan untuk model <i>CF</i>	46
Tabel 3.14 Hasil penyaringan untuk Model <i>CBF</i>	47
Tabel 3.15 Spesifikasi Perangkat Keras	50
Tabel 3.16 Tabel <i>User</i>	57
Tabel 3.17 Tabel <i>Anime</i>	57
Tabel 3.18 Tabel <i>Rating</i>	57
Tabel 4.1 Perbandingan Performa Algoritma <i>Collaborative Filtering</i>	60
Tabel 4.2 Hasil Top-10 <i>Tuning</i> Parameter Model <i>SVD</i>	60
Tabel 4.3 Hasil Evaluasi Final Model <i>Collaborative Filtering (SVD)</i>	61
Tabel 4.4 Hasil <i>Grid Search TF-IDF</i> (Diagnostik, sampel n=1.000)	62
Tabel 4.5 Hasil Evaluasi Final Model <i>CBF (TF-IDF + Cosine)</i>	64
Tabel 4.6 Eksperimen Optimasi Bobot <i>Hybrid</i>	65
Tabel 4.7 Hasil Evaluasi Final Model <i>Hybrid Filtering</i>	66
Tabel 4.8 Perbandingan Performa Model Rekomendasi.....	67
Tabel 4.9 Hasil <i>Black Box Testing</i>	78
Tabel 4.10 Perbandingan Evaluasi Model.....	81

DAFTAR LAMPIRAN

Lampiran 1 Listing Program – inti <i>Hybrid Filtering</i>	89
Lampiran 2 Kartu Bimbingan Skripsi	92



BAB I

PENDAHULUAN

1.1 Latar Belakang

Di era digital saat ini, kemajuan teknologi informasi telah membawa dampak signifikan terhadap berbagai aspek kehidupan, termasuk industri hiburan. Salah satu bentuk transformasi terbesar terjadi dalam distribusi dan konsumsi film serta animasi Jepang (*anime*), yang kini semakin terdigitalisasi melalui kehadiran *internet* dan platform *streaming* berbasis *web* (Velamentosa & Zuliarso, 2025).

Jika sebelumnya akses terhadap film terbatas pada media konvensional seperti bioskop, televisi terjadwal, atau penyewaan *DVD*, kini masyarakat dapat menikmati konten secara fleksibel melalui layanan digital *on-demand*. Perubahan ini memang memberikan kemudahan luar biasa, tetapi juga memunculkan tantangan baru, yaitu fenomena *paradox of choice*, di mana pengguna kerap merasa kewalahan dalam memilih tontonan akibat terlalu banyak pilihan yang tersedia (Wiputra & Shandi, 2021). Untuk menjawab tantangan tersebut, sistem rekomendasi dikembangkan sebagai solusi berbasis data untuk membantu pengguna menemukan konten yang relevan dengan preferensi mereka.

Namun demikian, adopsi sistem rekomendasi cerdas masih didominasi oleh platform berskala besar. Belum banyak pelaku usaha kecil, seperti warung digital, toko *retail*, atau konter pulsa, yang mengadopsi teknologi ini untuk meningkatkan layanan mereka. Padahal, kebutuhan akan pengalaman pengguna yang lebih personal juga relevan dalam konteks usaha mikro. Oleh karena itu, penelitian ini mengangkat studi kasus berbasis simulasi pada *MJ CELL*, sebuah konter pulsa keluarga yang dalam konteks penelitian digunakan sebagai media uji coba sistem rekomendasi layanan *streaming anime*.

MJ CELL dipilih karena mewakili karakteristik umum usaha mikro, seperti tidak memiliki sistem *login* pengguna, tidak mencatat histori tontonan, serta tidak menyimpan

preferensi pelanggan secara eksplisit. Kondisi ini menimbulkan dua tantangan utama dalam pengembangan sistem rekomendasi, yaitu *cold start* (minimnya data pengguna baru) dan *data sparsity* (kurangnya data interaksi). Oleh sebab itu, dibutuhkan pendekatan rekomendasi yang tetap dapat memberikan hasil yang relevan meskipun data yang tersedia sangat terbatas.

Sebagai solusi atas permasalahan tersebut, pendekatan *Hybrid Filtering* digunakan dengan menggabungkan kekuatan dua metode utama, yaitu *Collaborative Filtering* (CF) yang menganalisis pola kesamaan antar pengguna, serta *Content-Based Filtering* (CBF) yang mengevaluasi kesamaan antar konten berdasarkan atribut seperti *genre*, sinopsis, dan peringkat.

Berbagai studi sebelumnya telah membuktikan efektivitas pendekatan *Hybrid Filtering*. (Salmani & Kulkarni, 2021) mencatat bahwa penerapan *hybrid* dinamis dapat menurunkan nilai *Root Mean Square Error* (RMSE) sebesar 20% dibandingkan dengan pendekatan model tunggal. Sementara itu, (Arfisko & Wibowo, 2022) berhasil mengimplementasikan metode *Cascade Hybrid Filtering* untuk sistem rekomendasi film dengan nilai *Mean Average Precision* sebesar 0,3159. Dalam studi lain, (Mukti & Mardhiyah, 2022) mengembangkan sistem rekomendasi lisensi film berbasis *Hybrid Filtering* yang terbukti efektif dalam konteks distribusi konten skala kecil.

Namun demikian, sebagian besar penelitian tersebut berfokus pada skenario platform digital berskala besar atau *e-commerce*. Masih terbatas kajian yang mengangkat konteks sektor informal seperti usaha mikro dengan keterbatasan data. Oleh karena itu, penelitian ini bertujuan untuk mengisi celah tersebut dengan mengembangkan sistem rekomendasi film *anime* berbasis *Hybrid Filtering* dalam skenario simulatif pada konter pulsa *MJ CELL*. Sistem ini diharapkan efisien, adaptif, dan relevan bagi kebutuhan pelaku usaha kecil dalam mengadopsi teknologi berbasis data. Dengan latar belakang masalah yang dituliskan diatas,

penulis mengangkat judul “**OPTIMASI SISTEM REKOMENDASI FILM MENGGUNAKAN METODE *HYBRID FILTERING***”.

1.2 Identifikasi Masalah

Berdasarkan latar belakang masalah di atas, dapat diidentifikasi masalah sebagai berikut:

- a. Bagaimana merancang sistem rekomendasi film *anime* berbasis *Hybrid Filtering* yang efektif?
- b. Bagaimana mengatasi permasalahan *cold start* dan *data sparsity* dalam simulasi platform *streaming* berbasis usaha mikro?
- c. Bagaimana mengimplementasikan sistem rekomendasi tersebut dalam bentuk prototipe *web* yang efisien dan sederhana?

1.3 Ruang Lingkup

Ruang lingkup penelitian ini adalah sebagai berikut:

- a. Objek penelitian terbatas pada simulasi platform *streaming* mini yang merepresentasikan usaha mikro, dengan menggunakan *MJ CELL* sebagai media uji coba sistem rekomendasi. Penelitian ini tidak dilakukan secara langsung terhadap pengguna *MJ CELL* secara riil, melainkan sebagai kerangka simulatif.
- b. *Dataset* yang digunakan adalah *Dataset* sekunder yang bersumber dari *MyAnimeList*, diperoleh melalui platform *Kaggle* (Union, n.d.). *Dataset* ini mencakup data *rating* dari 73.516 pengguna terhadap 12.294 judul *anime*.
- c. Metode sistem rekomendasi yang dikembangkan mengadopsi pendekatan *Weighted Hybrid Filtering*, yaitu penggabungan antara *Collaborative Filtering* (*SVD*) dan *Content-Based Filtering* (*TF-IDF* + *Cosine Similarity*) dengan bobot yang dapat disesuaikan.

- d. Implementasi sistem difokuskan pada pengembangan prototipe *web* berbasis *Python* dan *Flask* sebagai backend, dengan pustaka pendukung seperti *Pandas*, *Scikit-learn*, dan *Surprise*. Penelitian tidak mencakup *deployment* ke server produksi atau pengujian di lingkungan bisnis secara langsung.
- e. Jenis konten yang direkomendasikan dibatasi hanya pada *anime*. Film non-*anime* atau *genre* lain di luar *anime* tidak termasuk dalam cakupan penelitian ini.
- f. Sistem ini hanya bersifat simulasi dan tidak digunakan untuk distribusi konten secara penuh. Fitur *video playback* yang tersedia pada prototipe hanya menampilkan *trailer* yang diambil dari platform *YouTube*, sebagai bentuk simulasi pemutaran konten.

1.4 Tujuan dan Manfaat

1.4.1 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

- a. Merancang dan mengembangkan sistem rekomendasi film *anime* berbasis *web* dengan pendekatan *Weighted Hybrid Filtering*, yang menggabungkan metode *Collaborative Filtering (SVD)* dan *Content-Based Filtering (TF-IDF + Cosine Similarity)*.
- b. Mengatasi permasalahan *cold start* dan *data sparsity* pada sistem rekomendasi di konteks usaha mikro yang tidak memiliki histori interaksi pengguna secara eksplisit, melalui simulasi studi kasus pada *MJ CELL*.
- c. Mengimplementasikan sistem rekomendasi tersebut dalam bentuk prototipe *website* yang dapat berfungsi sebagai model awal penerapan sistem cerdas untuk layanan *streaming* berskala kecil.

1.4.2 Manfaat

Manfaat yang dapat diperoleh dalam penelitian ini adalah sebagai berikut:

- a. Membantu pengguna baru yang belum memiliki riwayat tontonan agar tetap mendapatkan rekomendasi film *anime* yang relevan.
- b. Memungkinkan pemberian rekomendasi personal secara otomatis meskipun data interaksi pengguna terbatas.
- c. Memberikan dukungan bagi usaha mikro dalam menyediakan layanan tambahan berupa platform *streaming* yang cerdas, adaptif, dan efisien.

1.5 Sistematika Penulisan

Penulisan sistematis dimaksudkan untuk mempermudah pemahaman dan penelaahan penelitian. Dalam laporan penelitian ini, bahan yang ditulis dikelompokkan menjadi beberapa sub bab, yang secara garis besar dapat dijelaskan sebagai berikut:

BAB I PENDAHULUAN

Bab ini memuat penjelasan mengenai latar belakang masalah, identifikasi masalah, ruang lingkup penelitian, tujuan dan manfaat penelitian, serta sistematika penulisan.

BAB II LANDASAN TEORI

Bab ini memuat paparan mengenai teori-teori yang berhubungan dengan penelitian ini, serta mencakup tinjauan studi.

BAB III METODOLOGI PENELITIAN

Bab ini memuat paparan terkait metode yang digunakan dalam penelitian.

BAB IV HASIL DAN PEMBAHASAN

Bab ini memuat hasil evaluasi terhadap model yang diangkat, tampilan program, dan pengujian aplikasi.

BAB V SIMPULAN DAN SARAN

Bab ini memuat penjelasan mengenai simpulan penelitian berdasarkan hasil evaluasi dan penerapan yang telah dilakukan, serta rekomendasi yang dapat dikembangkan untuk penelitian di masa mendatang.

BAB II

LANDASAN TEORI

2.1 Konsep Dasar Sistem Rekomendasi

2.1.1 Sistem Rekomendasi

Sistem rekomendasi merupakan perangkat yang dirancang untuk membantu pengguna menemukan konten yang sesuai dengan preferensi, minat atau interaksi mereka sebelumnya (Sarhan et al., 2024). Sistem ini menjadi solusi penting dalam mengatasi fenomena *paradox of choice*, yakni kondisi di mana pengguna kesulitan menentukan pilihan karena banyaknya konten yang tersedia (Wiputra & Shandi, 2021). *Collaborative Filtering (CF)* adalah salah satu metode utama yang bekerja dengan memanfaatkan kemiripan perilaku antar pengguna, namun bergantung pada ketersediaan data historis yang cukup sehingga kurang efektif untuk pengguna atau item baru (Arfisko & Wibowo, 2022; Mukti & Mardhiyah, 2022).

Di sisi lain, *Content-Based Filtering (CBF)* mengandalkan perbandingan atribut konten yang pernah disukai pengguna dengan memanfaatkan teknik NLP seperti *TF-IDF* dan *Cosine Similarity* (Velamentosa & Zuliarso, 2025). Metode ini memberikan rekomendasi yang personal tetapi cenderung menghasilkan konten yang terlalu mirip, sehingga kurang beragam. Untuk mengatasi kekurangan dari masing-masing pendekatan, *Hybrid Filtering* menggabungkan *CF* dan *CBF* sehingga mampu memberikan rekomendasi yang lebih relevan dan variatif (Arfisko & Wibowo, 2022; Mukti & Mardhiyah, 2022).

Berdasarkan beragam tinjauan literatur, dapat diambil kesimpulan bahwa *Hybrid Filtering* muncul sebagai pendekatan paling adaptif dalam mengatasi kekurangan metode tunggal. Kolaborasi antara keunggulan *CF* dalam membaca perilaku pengguna serta ketepatan *CBF* dalam menilai kesamaan konten, memungkinkan sistem rekomendasi menghadirkan hasil yang lebih variatif sekaligus relevan. Pendekatan ini pun terbukti fleksibel untuk diadopsi di berbagai skala layanan, mulai dari platform berskala kecil hingga

aplikasi digital mikro, lantaran mampu menyesuaikan diri terhadap keterbatasan data maupun kebutuhan personalisasi yang tinggi.

2.1.2 Optimasi

Optimasi pada sistem rekomendasi berfokus pada peningkatan kinerja dan akurasi dalam memberikan rekomendasi yang relevan bagi pengguna. Sistem rekomendasi digunakan secara luas dalam berbagai domain, termasuk *e-commerce*, *e-learning*, dan platform *streaming*, untuk memfasilitasi pengambilan keputusan pengguna dengan filter informasi yang lebih baik (Fayyaz et al., 2020; Şeref et al., 2021). Berdasarkan pendapat beberapa peneliti diatas dapat disimpulkan bahwa peningkatan kinerja dan akurasi ini berarti bahwa sistem mampu menyajikan rekomendasi yang tidak hanya sesuai dengan preferensi pengguna, tetapi juga meminimalkan kesalahan prediksi. Dengan demikian, pengguna dapat menemukan produk, materi pembelajaran, atau konten hiburan yang sesuai secara lebih cepat dan efisien.

2.1.3 Film

Film merupakan media visual yang menampilkan rangkaian gambar bergerak yang memiliki alur cerita dan disertai elemen audio, dan secara umum dikonsumsi sebagai hiburan di berbagai platform seperti bioskop, televisi, dan layanan *streaming* (Girsang et al., 2020). Dalam konteks sistem rekomendasi, film direpresentasikan sebagai objek data yang memiliki atribut seperti judul, *genre*, durasi, serta *rating* yang diberikan oleh pengguna berdasarkan pengalaman menonton (Jena et al., 2022). Setiap interaksi antara pengguna dan film akan direkam dan diubah menjadi nilai numerik dalam bentuk matriks *user-item*, yang kemudian dianalisis untuk memprediksi preferensi pengguna terhadap film lain yang belum ditonton (Sarhan et al., 2024).

2.1.4 *Anime*

Anime merupakan salah satu bentuk animasi asal Jepang yang memiliki ciri khas pada gaya visual, kedalaman cerita, dan *genre* yang beragam, serta memiliki basis penggemar yang kuat di seluruh dunia (Putri & Faisal, 2023). Popularitas *anime* yang semakin meningkat telah mendorong banyak layanan *streaming* menghadirkan katalog khusus *anime* dalam platform mereka, sehingga jumlah konten yang tersedia menjadi sangat besar dan beragam (Girsang et al., 2020). Dalam konteks sistem rekomendasi, *anime* dipandang sebagai objek data yang ideal karena memiliki atribut yang kompleks seperti multi-*genre*, format penayangan (*TV series*, *OVA*, *movie*), dan skor penilaian komunitas yang kaya untuk dianalisis lebih lanjut (Jena et al., 2022). Kompleksitas tersebut memungkinkan sistem rekomendasi untuk mengidentifikasi kesamaan antar judul dan preferensi pengguna secara lebih presisi, sehingga pengalaman menonton dapat menjadi lebih personal dan relevan (Sarhan et al., 2024).

2.2 Metode Algoritma

2.2.1 *Collaborative Filtering (CF)*

Collaborative Filtering (CF) metode populer dalam sistem rekomendasi yang menggunakan data interaksi pengguna, seperti *Rating* atau ulasan, untuk memberikan rekomendasi yang dipersonalisasi (Jena et al., 2022; Sarhan et al., 2024). *CF* bekerja dengan menemukan pola kesamaan preferensi antar pengguna atau item (Girsang et al., 2020). Ada dua jenis utama: *user-based*, yang merekomendasikan item berdasarkan kesamaan antar pengguna, dan *item-based*, yang merekomendasikan item serupa dengan yang pernah disukai pengguna (Jena et al., 2022; Putri & Faisal, 2023). Keunggulan *CF* adalah kemampuannya memberikan rekomendasi relevan hanya dengan data interaksi pengguna, tanpa memerlukan informasi detail tentang fitur item (Sarhan et al., 2024).

a. Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) merupakan teknik reduksi dimensi yang umum digunakan dalam sistem rekomendasi untuk menyederhanakan struktur data pada matriks besar, khususnya matriks *Rating* antara pengguna dan item (Sarhan et al., 2024). Dalam konteks sistem rekomendasi, *SVD* digunakan untuk memprediksi nilai-nilai yang hilang dalam matriks interaksi, seperti *Rating* yang belum diberikan oleh pengguna terhadap item tertentu (Putri & Faisal, 2023). Nilai prediktif tersebut kemudian digunakan untuk menghasilkan rekomendasi yang dipersonalisasi berdasarkan pola laten dari pengguna dan item (Jena et al., 2022). Secara matematis, proses dekomposisi matriks dalam metode *SVD* direpresentasikan sebagai berikut:

$$\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T \quad (1)$$

di mana $\mathbf{R}$ merupakan matriks interaksi antara pengguna dan item, $\mathbf{U}$ adalah matriks ortogonal pengguna terhadap fitur laten, $\mathbf{\Sigma}$ adalah matriks diagonal yang berisi nilai singular, dan $\mathbf{V}^T$ adalah transpose dari matriks item terhadap fitur laten.

2.2.2 Content-Based Filtering (CBF)

Content-Based Filtering (CBF) adalah metode dalam sistem rekomendasi yang memberikan saran kepada pengguna berdasarkan atribut atau fitur dari item yang telah disukai sebelumnya, seperti *genre*, sinopsis, atau deskripsi lainnya (Pramesti & Santiyasa, 2022). Metode ini bekerja dengan membandingkan kesamaan antar item berdasarkan karakteristik kontennya, sehingga sistem dapat menemukan item yang memiliki kemiripan dengan item yang pernah diinteraksikan oleh pengguna (Sarhan et al., 2024). Pendekatan ini tidak memerlukan data dari pengguna lain, melainkan membangun profil pengguna berdasarkan item yang pernah mereka sukai (Jena et al., 2022).

a. *TF-IDF*

Salah satu teknik representasi teks yang sering digunakan dalam *Content-Based Filtering* adalah *Term Frequency–Inverse Document Frequency (TF-IDF)*. *TF-IDF* adalah metode pembobotan teks yang menghitung seberapa penting suatu kata dalam sebuah dokumen relatif terhadap seluruh dokumen dalam korpus (Pramesti & Santiyasa, 2022). *Term Frequency (TF)* mengukur seberapa sering kata muncul dalam suatu dokumen, sedangkan *Inverse Document Frequency (IDF)* menilai apakah kata tersebut umum atau jarang di seluruh dokumen. Kombinasi TF dan IDF ini digunakan untuk membentuk vektor dari setiap item atau dokumen, yang kemudian digunakan untuk menghitung kemiripan antar item (Jena et al., 2022).

Secara matematis, bobot dari setiap kata t dalam dokumen d dihitung dengan mengalikan nilai TF dan IDF, yang dirumuskan sebagai berikut:

$$tf-idf(t, d) = tf(t, d) \times idf(t) \quad (2)$$

Di mana: $tf(t, d)$ adalah jumlah kemunculan kata t pada dokumen d , dan $idf(t)$ adalah ukuran yang menunjukkan seberapa jarang kata t muncul pada seluruh koleksi dokumen

b. *Cosine Simillarity*

Cosine Simillarity merupakan metode yang digunakan untuk mengukur tingkat kemiripan antara dua vektor dalam ruang multidimensi dengan cara menghitung nilai sudut kosinus di antara keduanya (Juni Permana & Wibowo, 2023). Dalam konteks sistem rekomendasi berbasis konten, *Cosine Similarity* banyak digunakan untuk menghitung kesamaan antar item seperti film, berdasarkan representasi numerik hasil ekstraksi fitur menggunakan metode *TF-IDF* (Pramesti & Santiyasa, 2022). Secara matematis, perhitungan *Cosine Similarity* dapat dituliskan sebagai berikut:

$$\text{Cosine Simillarity } (x, y) = \frac{x \cdot y}{\|x\| \times \|y\|} \quad (3)$$

di mana $x \cdot y$ merupakan hasil perkalian titik (dot product) dari dua vektor, dan $\|x\|$ serta $\|y\|$ masing-masing adalah panjang (norma Euclidean) dari vektor x dan y .

2.2.3 *Popularity-Based Recommender*

Popularity-Based Recommender merupakan metode rekomendasi sederhana yang menyarankan item berdasarkan tingkat popularitas, diukur dari metrik seperti jumlah pembelian, unduhan, penayangan, atau *rating* tertinggi dalam periode tertentu. Popularitas dihitung secara agregat tanpa memperhatikan preferensi individu, sehingga semua pengguna menerima daftar rekomendasi yang sama (Chaudhary & Anupama, 2020).

Metode ini unggul karena mudah diimplementasikan, tidak memerlukan data personal yang kompleks, serta memiliki kecepatan tinggi dengan biaya komputasi rendah. Umumnya digunakan sebagai *baseline* penelitian atau *fallback* pada situasi *cold-start* (Herce-Zelaya et al., 2022). Namun, kelemahannya terletak pada *popularity bias*, yaitu kecenderungan merekomendasikan item populer secara berulang sehingga mengurangi eksposur item baru atau niche. Meski demikian, metode ini tetap relevan sebagai pembanding, solusi awal, atau komponen dalam model *hybrid* untuk menangkap tren global secara efisien (Yalcin & Bilge, 2021).

2.2.4 *Hybrid Filtering*

Hybrid Filtering merupakan pendekatan dalam sistem rekomendasi yang menggabungkan dua atau lebih metode dasar, biasanya *Collaborative Filtering* (CF) dan *Content-Based Filtering* (CBF), untuk mengatasi kelemahan dari masing-masing metode (Arfisko & Wibowo, 2022). CF unggul dalam menemukan pola kolektif dari interaksi pengguna, namun lemah dalam kondisi *cold start* dan *data sparsity*. Sebaliknya, CBF dapat

tetap berfungsi meskipun data interaksi minim, tetapi cenderung menghasilkan rekomendasi yang sempit atau *over-specialized* (Putri & Faisal, 2023).

Dengan menggabungkan kedua pendekatan tersebut, *Hybrid Filtering* dapat menghasilkan sistem yang lebih stabil, akurat, dan adaptif terhadap berbagai situasi data pengguna (Mukti & Mardhiyah, 2022). Sistem *hybrid* mampu menggunakan kekuatan prediktif dari *CF* dan memanfaatkan representasi atribut konten dari *CBF* untuk memberikan rekomendasi yang lebih relevan. Kombinasi ini juga meningkatkan kemampuan sistem dalam menghasilkan rekomendasi yang lebih beragam dan mengurangi risiko bias dari satu metode tunggal (Sarhan et al., 2024). Terdapat beberapa teknik untuk mengimplementasikan *Hybrid Filtering*, antara lain: *weighted hybrid*, *switching hybrid*, *cascade hybrid*, *feature combination*, *Dynamic Hybrid* di mana masing-masing memiliki cara berbeda dalam menggabungkan informasi dari berbagai sumber untuk menghasilkan rekomendasi yang lebih relevan dan personal (Arfisko & Wibowo, 2022).

a. Weighted Hybrid

Dalam penelitian ini digunakan pendekatan *Weighted Hybrid Filtering*, yaitu metode yang menggabungkan skor prediksi dari *Collaborative Filtering* (*CF*) dan *Content-Based Filtering* (*CBF*) dengan bobot tertentu. Kombinasi skor akhir rekomendasi dapat dirumuskan secara matematis sebagai berikut:

$$Score_{Hybrid} = \alpha \cdot Score_{CF} + (1 - \alpha) \cdot Score_{CBF} \quad (4)$$

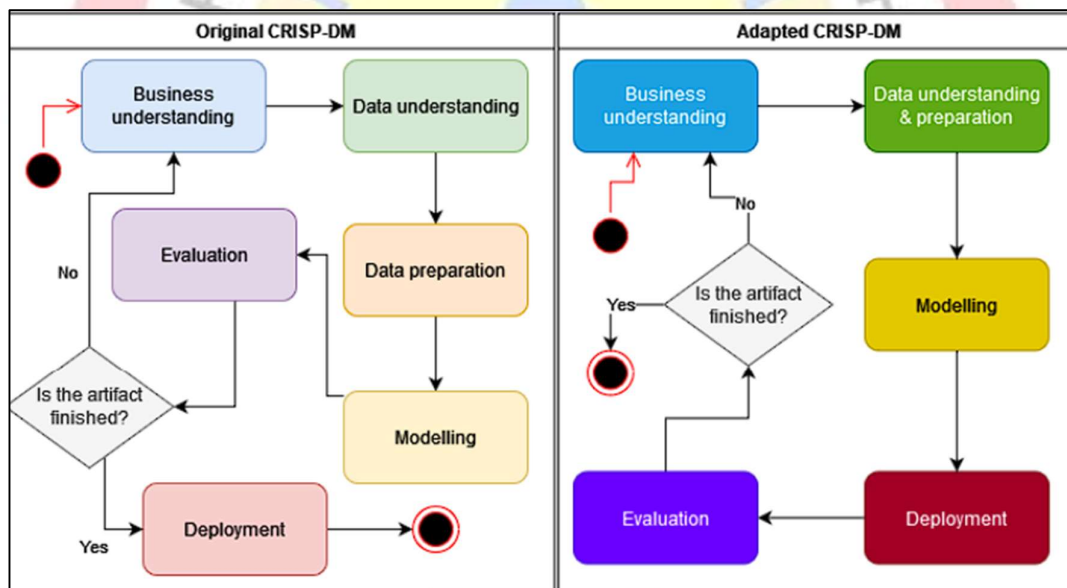
di mana :

- a. α adalah bobot untuk model *CF* (nilai antara 0 dan 1)
- b. $1 - \alpha$ adalah bobot untuk model *CBF*
- c. $Score_{CF}$ adalah skor prediksi dari model Collaborative Filtering
- d. $Score_{CBF}$ adalah skor prediksi dari model Collaborative Filtering

Nilai α ditentukan berdasarkan hasil pengujian performa masing-masing model, sehingga kontribusi CF dan CBF dapat disesuaikan untuk menghasilkan rekomendasi dengan akurasi optimal.

2.3 Metodologi *CRIPS-DM*

CRISP-DM (*Cross Industry Standard Process for Data Mining*) adalah metodologi siklik standar industri untuk proyek penambangan data, mencakup enam fase utama: *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, dan *Deployment* (Schröder et al., 2021). Meskipun dikembangkan pada akhir tahun 1990-an, literatur terbaru menunjukkan bahwa *CRISP-DM* masih menjadi model utama dalam proyek *machine learning* dan *data science* karena kapabilitasnya untuk mendukung iterasi dan adaptasi berbasis kebutuhan proyek yang berubah (Grotentraast et al., 2024).



Gambar 2.1 Gambar *CRIPS-DM*

(Grotentraast et al., 2024)

Dalam konteks sistem rekomendasi, seperti yang ditunjukkan oleh (Grotentraast et al., 2024), *CRISP-DM* dapat diadaptasi secara efektif untuk mendukung seluruh tahapan pengembangan, mulai dari pemahaman kebutuhan pengguna hingga *deployment* sistem rekomendasi berbasis data. (Grotentraast et al., 2024) menunjukkan bahwa adaptasi proses

ini memungkinkan integrasi kebutuhan pengguna (*business understanding*), pengolahan data interaksi pengguna (*data preparation*), penerapan model rekomendasi berbasis filtering (*modeling*), hingga penyajian sistem melalui antarmuka rekomendasi (*deployment*). berdasarkan jarak yang paling dekat dengannya (Isman et al., 2021). Dengan fleksibilitas tersebut, *CRISP-DM* memfasilitasi pengujian model alternatif, evaluasi menggunakan metrik kuantitatif, serta penyempurnaan berulang—fitur penting untuk sistem rekomendasi yang harus terus disesuaikan dengan perilaku pengguna. Oleh karena itu, dalam penelitian ini digunakan kerangka *CRISP-DM* yang telah disesuaikan (*adapted CRISP-DM*) untuk mendukung proses pengembangan sistem rekomendasi film *anime* berbasis *Weighted Hybrid Filtering*.

2.4 Tools dan Framework

2.4.1 Bahasa Pemrograman Python

Python merupakan salah satu bahasa pemrograman yang banyak digunakan dalam pengembangan aplikasi modern, termasuk pembuatan sistem berbasis *web*, analisis data, dan implementasi algoritma pada bidang ilmu komputer lanjutan seperti *artificial intelligence* dan *data processing* skala besar. Salah satu keunggulan utama *Python* terletak pada sintaksisnya yang sederhana dan mudah dipahami, sehingga memungkinkan proses penulisan kode menjadi lebih efisien dibandingkan dengan bahasa pemrograman lain seperti Java atau C++ (Ling et al., 2021). Dengan penggunaan modul siap pakai dari ekosistem *Python*, pengembang dapat dengan cepat membangun dan menguji prototipe, yang mempercepat proses inovasi. *Python* juga menyediakan beragam modul siap pakai, seperti Matplotlib, Numpy, *Pandas*, *Scikit-learn*, dan banyak pustaka lainnya yang mendukung berbagai kebutuhan pengembangan (Silaparasetty, 2020).

Berdasarkan pendapat sebelumnya, dapat disimpulkan bahwa *Python* merupakan bahasa pemrograman yang sangat mendukung untuk pengembangan sistem berbasis data,

khhususnya sistem rekomendasi. Pada penelitian ini, pemilihan *Python* didasarkan pada kemampuannya dalam mengelola data serta mengimplementasikan algoritma *Hybrid Filtering* secara efektif melalui beragam pustaka siap pakai yang tersedia.

2.4.2 Flask

Flask merupakan framework aplikasi web berbasis *Python* yang ringan dan fleksibel, dirancang untuk memudahkan pengembangan aplikasi web secara cepat sekaligus mampu diskalakan untuk kebutuhan yang lebih kompleks (Silaparasetty, 2020). *Flask* adalah sebuah *microframework web Python* yang ringan, mengikuti prinsip minimalis (*micro-framework*) dengan menyediakan komponen dasar tanpa membatasi struktur aplikasi secara ketat. *Flask* digunakan sebagai antarmuka yang menghubungkan antara model rekomendasi dengan pengguna akhir melalui tampilan berbasis *web*. *Framework* ini mendukung integrasi yang mulus dengan pustaka-pustaka analisis data seperti *Pandas*, *NumPy*, dan *Scikit-learn*, serta memudahkan implementasi *routing* dan penyajian *output* rekomendasi kepada pengguna (Mukti & Mardhiyah, 2022).

2.4.3 SQLite

SQLite merupakan sistem manajemen basis data relasional (*Relational Database Management System / RDBMS*) yang ringan dan bersifat *self-contained*, dirancang untuk menyimpan, mengelola, dan memproses data tanpa memerlukan proses server terpisah. *SQLite* menyimpan seluruh data dalam satu file tunggal yang portabel, sehingga mudah dipindahkan antar perangkat atau sistem operasi tanpa konfigurasi rumit (Pawlaszczyk, 2022).

SQLite digunakan sebagai media penyimpanan utama dalam penelitian ini untuk menyimpan dataset anime, data pengguna, dan riwayat *rating* secara terstruktur. Sistem ini mendukung bahasa kueri *SQL* secara penuh, memudahkan pengelolaan data dengan perintah seperti *SELECT*, *INSERT*, *UPDATE*, dan *DELETE* (Jain et al., 2023). *SQLite* juga mendukung

integrasi langsung dengan bahasa pemrograman *Python* melalui pustaka seperti *sqlite3* atau *ORM (Object-Relational Mapping)* seperti *SQLAlchemy*, sehingga mempermudah proses pengambilan data, penyimpanan hasil prediksi, dan pengelolaan informasi pengguna dalam sistem rekomendasi (Hug, 2020).

2.4.4 Surprise

Surprise (Simple Python Recommendation System Engine) merupakan library *Python* yang dirancang khusus untuk membangun dan mengevaluasi sistem rekomendasi berbasis *Collaborative Filtering*, baik dengan pendekatan *user-based* maupun *item-based*, termasuk metode *matrix factorization* seperti *Singular Value Decomposition (SVD)* (Jena et al., 2022). Library ini mempermudah pengembang dalam melakukan eksperimen sistem rekomendasi melalui antarmuka yang sederhana, beragam algoritma siap pakai, serta dukungan fitur evaluasi performa seperti *RMSE*, *MAE*, dan *cross-validation* (Putri & Faisal, 2023).

2.4.5 Scikit-learn

Scikit-learn adalah pustaka *Python* yang umum digunakan dalam pengembangan sistem berbasis *machine learning*, termasuk dalam konteks sistem rekomendasi berbasis konten (*Content-Based Filtering*). Pustaka ini menyediakan berbagai algoritma klasifikasi, regresi, klustering, serta fungsi evaluasi model seperti *Precision*, *Recall*, dan *F1-Score* (Putri & Faisal, 2023). Dalam penelitian ini, *Scikit-learn* dimanfaatkan untuk melakukan vektorisasi fitur menggunakan *TF-IDF*, menghitung kemiripan antar item dengan *Cosine Similarity*, serta mendukung proses validasi model berbasis metrik kuantitatif (Jena et al., 2022).

2.4.6 NumPy

NumPy (Numerical Python) adalah pustaka fundamental dalam ekosistem *Python* yang menyediakan dukungan untuk array multidimensi dan operasi numerik berskala besar. Dalam pengembangan sistem rekomendasi, *NumPy* berperan penting dalam proses

komputasi matriks, terutama saat membangun matriks interaksi *user-item*, perhitungan nilai *similarity*, dan manipulasi struktur data numerik lainnya (Putri & Faisal, 2023).

2.4.7 *Pandas*

Pandas adalah pustaka *Python* yang digunakan untuk manipulasi dan analisis data dalam format tabular seperti *CSV*. Dalam sistem rekomendasi, *Pandas* sangat berguna untuk memuat *Dataset*, membersihkan data, serta mengelola atribut seperti *user_id*, *anime_id*, *rating*, dan *genre*. Keunggulan utama *Pandas* terletak pada struktur *DataFrame* yang fleksibel dan mendukung proses data secara efisien (Mukti & Mardhiyah, 2022; Putri & Faisal, 2023).

2.5 Evaluasi Sistem Rekomendasi

2.5.1 Evaluasi Prediksi *Rating*

Evaluasi prediksi digunakan untuk mengukur seberapa akurat sistem dalam memprediksi *rating* yang akan diberikan pengguna terhadap item yang belum pernah diinteraksikan. Dua metrik umum yang sering digunakan sebagai berikut :

a. *Root Mean Square Error (RMSE)*

Root Mean Square Error (RMSE) merupakan salah satu metrik evaluasi yang paling umum digunakan untuk mengukur tingkat perbedaan antara nilai prediksi yang dihasilkan oleh sistem dengan nilai aktual (*ground truth*). Dalam konteks sistem rekomendasi berbasis prediksi *rating*, *RMSE* sangat relevan karena mampu memberikan ukuran kesalahan dalam satuan yang sama dengan skala *rating* yang digunakan.

Secara formal, *RMSE* didefinisikan sebagai akar kuadrat dari rata-rata kuadrat selisih antara nilai prediksi ($\hat{r}_{ui}$) dan nilai aktual (r_{ui}) untuk setiap pasangan pengguna-item (u, i). dalam himpunan evaluasi. Semakin rendah nilai *RMSE*, semakin baik kinerja model dalam menghasilkan prediksi yang mendekati nilai sebenarnya (Valentino & Setiawan, 2024). Rumusnya dituliskan sebagai berikut:

$$RMSE = \sqrt{\frac{1}{N} \sum_{(u,i) \in T} (r_{ui} - \hat{r}_{ui})^2} \quad (5)$$

Di mana :

- a. N adalah jumlah total pasangan *user-item* dalam data uji T ,
- b. r_{ui} adalah *rating* aktual dari pengguna u terhadap item i ,
- c. $\hat{r}_{ui}$ adalah *rating* prediksi untuk pengguna u terhadap item i .
- d. T adalah himpunan pasangan pengguna–item yang digunakan untuk evaluasi.

Semakin kecil nilai $RMSE$, semakin tinggi tingkat akurasi prediksi yang dihasilkan model. Karena perhitungannya melibatkan kuadrat dari selisih prediksi dan nilai aktual, $RMSE$ memberikan penalti yang lebih besar terhadap kesalahan prediksi yang besar, sehingga menjadikannya lebih peka terhadap *outlier* dibandingkan metrik seperti MAE (*Mean Absolute Error*) (Hodson, 2022).

b. Mean Absolute Error (MAE)

Mean Absolute Error (MAE) merupakan salah satu metrik evaluasi yang digunakan untuk mengukur rata-rata besar kesalahan absolut antara nilai prediksi yang dihasilkan model dengan nilai aktual (*ground truth*). Dalam konteks sistem rekomendasi berbasis prediksi *rating*, MAE menunjukkan seberapa besar, secara rata-rata, perbedaan antara *rating* yang diprediksi $\hat{r}_{ui}$ dan *rating* sebenarnya r_{ui} untuk setiap pasangan pengguna–item (u, i) .

Tidak seperti $RMSE$ yang mengkuadratkan selisih sebelum dihitung rata-ratanya, MAE hanya menggunakan nilai absolut dari selisih tersebut, sehingga metrik ini tidak terlalu sensitif terhadap *outlier*. Semakin kecil nilai MAE , semakin baik kemampuan model dalam menghasilkan prediksi yang mendekati nilai aktual. Secara matematis, MAE dirumuskan sebagai berikut:

$$MAE = \frac{1}{N} \sum_{(u,i) \in T} |r_{ui} - \hat{r}_{ui}| \quad (6)$$

Di mana :

- a. N adalah jumlah total pasangan *user-item* dalam data uji T ,
- b. r_{ui} adalah *rating* aktual dari pengguna u terhadap item i ,
- c. $\hat{r}_{ui}$ adalah *rating* prediksi untuk pengguna u terhadap item i .
- d. T adalah himpunan pasangan pengguna–item yang digunakan untuk evaluasi.

MAE merepresentasikan rata-rata besar kesalahan prediksi dalam skala *rating*, serta memiliki toleransi yang lebih tinggi terhadap *outlier* dibandingkan $RMSE$ karena tidak mengkuadratkan nilai error (Robeson & Willmott, 2023). Metrik ini dianggap intuitif dan mudah dipahami, sehingga sering digunakan pada tahap awal evaluasi sistem rekomendasi, khususnya ketika distribusi *rating* relatif seragam atau tidak mengandung banyak nilai ekstrem (Li et al., 2021).

2.5.2 Evaluasi Top-N Rekomendasi

Evaluasi ini bertujuan untuk mengukur sejauh mana sistem mampu merekomendasikan daftar item terbaik kepada pengguna. Metode ini digunakan ketika fokus evaluasi bukan pada prediksi nilai *rating* secara numerik, melainkan pada kualitas daftar rekomendasi yang ditampilkan. Beberapa metrik yang umum digunakan dalam evaluasi berbasis *top-N recommendation* antara lain:

a. *Precision@K*

Precision@K merupakan metrik evaluasi yang digunakan untuk mengukur proporsi item relevan di antara K item teratas yang direkomendasikan kepada pengguna. Dalam konteks sistem rekomendasi, nilai *Precision@K* menunjukkan seberapa tepat rekomendasi yang dihasilkan sistem dalam menghadirkan item yang benar-benar sesuai

dengan preferensi pengguna. Semakin tinggi nilai $Precision@K$, semakin besar persentase item relevan yang ditampilkan pada daftar rekomendasi teratas (Thouvenin et al., 2022). Secara matematis, $Precision@K$ dirumuskan sebagai berikut:

$$Precision@K = \frac{\text{Jumlah item relevan dalam top } K}{K} \quad (7)$$

b. Recall@k

$Recall@K$ adalah metrik yang digunakan untuk mengukur proporsi item relevan yang berhasil direkomendasikan oleh sistem dibandingkan dengan total item relevan yang tersedia untuk pengguna. Metrik ini menilai sejauh mana cakupan sistem dalam menemukan semua item relevan yang mungkin diminati pengguna. Nilai $Recall@K$ yang tinggi menunjukkan bahwa sistem berhasil menampilkan sebagian besar item relevan dalam daftar rekomendasi (Hosseinzadeh Kassani et al., 2021). Secara matematis, $Recall@K$ dirumuskan sebagai berikut:

$$Recall@k = \frac{\text{Jumlah item relevan yang diambil di dalam top } k}{k} \quad (8)$$

c. F1-Score@k

$F1-Score@K$ merupakan rata-rata harmonis dari $Precision@K$ dan $Recall@K$, yang digunakan untuk menilai keseimbangan antara ketepatan (*precision*) dan cakupan (*recall*) sistem rekomendasi. Metrik ini berguna ketika diperlukan penilaian yang mempertimbangkan kedua aspek secara seimbang. Nilai $F1-Score@K$ yang tinggi mengindikasikan bahwa sistem mampu menghasilkan rekomendasi yang relevan sekaligus mencakup sebagian besar item relevan yang tersedia (Yacoubby & Axman, 2020). Secara matematis, $F1-Score@K$ dirumuskan sebagai berikut:

$$F1 - Score@K = 2 \times \frac{Recall \times Precision}{Recall + Precision} \quad (9)$$

2.5.3 Evaluasi Fungsional (*Black Box Testing*)

Selain metrik kuantitatif, sistem juga diuji melalui pendekatan *Black Box Testing*, yaitu metode pengujian yang berfokus pada fungsi dan *output* sistem tanpa memeriksa struktur internal kode. *Black Box Testing* adalah metode pengujian perangkat lunak yang fokus pada evaluasi fungsi sistem tanpa mempertimbangkan struktur internal atau kode sumbernya. Dalam pengujian ini, *tester* hanya mendalami *input* dan *output* sistem untuk menentukan apakah fungsinya sesuai dengan spesifikasi yang telah ditetapkan (Ibitomi et al., 2021; Zardari et al., 2022). Metode ini sangat efektif dalam mengidentifikasi berbagai jenis kesalahan, terutama pada tahap awal pengembangan perangkat lunak. Salah satu manfaat besar dari *black box testing* adalah kemampuannya untuk digunakan dalam berbagai jenis sistem, termasuk aplikasi *web* dan sistem informasi, di mana pengujian dapat dilakukan dengan mengirimkan berbagai masukan dan menganalisis keluaran (Baniş et al., 2021; Shahid et al., 2022). Teknik ini juga sering digunakan dalam kombinasi dengan pengujian lain, seperti *white box testing*, untuk mengoptimalkan deteksi kerentanan dan meningkatkan keamanan (Moghadam et al., 2023). Dengan pendekatan ini, pengujian menjadi lebih komprehensif, memastikan bahwa berbagai aspek dari aplikasi teruji dengan baik.

2.6 Tinjauan Studi

Penelitian 1, oleh (Putri & Faisal, 2023) yang berjudul “*Analyzing the Effectiveness of Collaborative Filtering and Content-Based Filtering Methods in Anime Recommendation Systems*” berfokus pada pengembangan sistem rekomendasi *Anime* dengan tujuan menemukan metode yang paling efektif antara *Collaborative Filtering (CF)* dan *Content-Based Filtering (CBF)*. Metode yang digunakan mencakup eksplorasi dan pra-pemrosesan data, pengembangan model menggunakan algoritma seperti *Singular Value Decomposition (SVD)* untuk *CF* dan analisis atribut konten untuk *CBF*. *Dataset* yang digunakan berasal dari *Kaggle*, berisi lebih dari 12.000 entri *Anime* dan lebih dari 7 juta *Rating* pengguna. Pengujian

dilakukan melalui metode *A/B testing* terhadap dua kelompok pengguna yang menerima rekomendasi dari masing-masing metode, dengan evaluasi menggunakan survei skala *Likert* dan analisis statistik (*uji-t* atau *ANOVA*). Hasilnya menunjukkan bahwa *CBF* memperoleh rata-rata kepuasan lebih tinggi (4,7) dibandingkan *CF* (4,3), dengan nilai $p = 0,04$ yang menunjukkan perbedaan signifikan. Kesimpulannya, meskipun keduanya memiliki keunggulan, *CBF* lebih efektif dalam memberikan rekomendasi personal, dan integrasi keduanya dalam pendekatan *hybrid* disarankan untuk hasil yang lebih optimal.

Penelitian 2, oleh (Girsang et al., 2020) yang berjudul “*Collaborative Recommendation System in Users of Anime Films*” berfokus pada pengembangan sistem rekomendasi *anime* berbasis *Collaborative Filtering* (*CF*), dengan tujuan memberikan rekomendasi film *anime* berdasarkan riwayat penilaian pengguna. Metode yang digunakan adalah *Collaborative Filtering* dengan pendekatan *Alternating Least Squares* (*ALS*) dan *SimRank* untuk menghitung kemiripan antar pengguna. *Dataset* yang digunakan berasal dari *Kaggle*, mencakup 73.516 pengguna dan 12.294 judul *anime* beserta *Rating*-nya. Sistem diuji dengan membagi data menjadi 70% untuk pelatihan dan 30% untuk pengujian, serta mengevaluasi model menggunakan metrik *Root Mean Square Error* (*RMSE*). Hasil pengujian menunjukkan nilai *RMSE* sebesar 2,5375. Penelitian ini menyimpulkan bahwa pendekatan *ALS* dapat membangun sistem rekomendasi yang efisien dan layak digunakan, meskipun eksperimen dilakukan pada *subset* 100 ribu data karena keterbatasan sumber daya komputasi.

Penelitian 3, oleh (Mukti & Mardhiyah, 2022) yang berjudul “*Sistem Rekomendasi Pembelian Lisensi Film Menggunakan Pendekatan Hybrid Filtering (Studi Kasus: Film Animasi Jepang)*” berfokus pada pengembangan sistem rekomendasi untuk membantu perusahaan *streaming* dalam memilih lisensi film yang potensial, dengan tujuan meminimalkan risiko kesalahan pembelian lisensi. Metode yang digunakan adalah *Hybrid*

Filtering yang menggabungkan *Content-Based Filtering* menggunakan *Cosine Similarity* terhadap sinopsis dan *Collaborative Filtering* menggunakan algoritma *Singular Value Decomposition (SVD)*. Data yang digunakan berasal dari *MyAnimeList* dan tersedia di *Kaggle*, mencakup 17.562 judul *anime* dan preferensi dari 325.772 pengguna. Sistem diuji dengan *confusion matrix* untuk memperoleh nilai presisi, serta menggunakan metrik evaluasi *Mean Absolute Error (MAE)* dan *Root Mean Square Error (RMSE)*. Hasil pengujian menunjukkan nilai presisi 71%, *MAE* sebesar 0,9, dan *RMSE* sebesar 1,1985, yang menunjukkan bahwa sistem mampu memberikan rekomendasi dengan akurasi yang cukup baik. Penelitian ini menyimpulkan bahwa metode *hybrid* efektif dalam menghasilkan daftar rekomendasi yang relevan dan berhasil diimplementasikan dalam bentuk *website* menggunakan *framework Flask*.

Penelitian 4, oleh (Arfisko & Wibowo, 2022) yang berjudul “*Sistem Rekomendasi Film Menggunakan Metode Hybrid Collaborative Filtering dan Content-Based Filtering*” berfokus pada pengembangan sistem rekomendasi film dengan tujuan meningkatkan akurasi dan relevansi hasil rekomendasi melalui pendekatan *hybrid*. Metode yang digunakan adalah *Hybrid Filtering* dengan pendekatan *cascade*, yaitu menggabungkan *Item-Based Collaborative Filtering (CF)* sebagai filter awal dan *Content-Based Filtering (CBF)* sebagai pengurut akhir berdasarkan profil pengguna dan konten. *Dataset* yang digunakan adalah *MovieLens 100K*, mencakup 100.836 *Rating* dari 610 pengguna terhadap 9.742 film, serta 32.977 *tag* sebagai konten item. Sistem diuji menggunakan *5-fold cross-validation* dan dievaluasi dengan metrik *Hit Rate* dan *Mean Average Precision (MAP@10)*. Hasilnya, metode *Hybrid CF-CBF* menunjukkan performa terbaik dengan nilai *Hit Rate* sebesar 0,8042 dan *MAP@10* sebesar 0,3159, mengungguli metode *CF* (0,7904 dan 0,3054), *CBF* (0,7009 dan 0,2219), serta *Hybrid CBF-CF* (0,74 dan 0,2572). Kesimpulan dari penelitian ini adalah bahwa pendekatan *Hybrid CF-CBF* secara signifikan meningkatkan kualitas

rekomendasi dibandingkan metode individual maupun urutan *hybrid* yang dibalik, terutama dalam memprioritaskan item relevan di posisi atas daftar rekomendasi.

Penelitian 5, oleh (Juni Permana & Wibowo, 2023) yang berjudul “*Movie Recommendation System Based on Synopsis Using Content-Based Filtering with TF-IDF and Cosine Similarity*” berfokus pada pengembangan sistem rekomendasi film berbasis sinopsis, dengan tujuan memberikan rekomendasi yang relevan melalui ekstraksi kata kunci dari sinopsis film. Metode yang digunakan adalah *Content-Based Filtering* dengan pendekatan *TF-IDF* dan *Cosine Similarity*, serta ekstraksi kata kunci menggunakan pustaka *Gensim*. Data yang digunakan berasal dari *MovieLens*, terdiri atas 45.443 data film dan 99.806 riwayat penayangan oleh pengguna. Evaluasi dilakukan dalam dua skenario—berdasarkan film pertama yang ditonton dan berdasarkan film yang disukai pengguna—dengan metrik evaluasi berupa *Precision*, *Recall*, dan *F1-Score*. Hasil evaluasi menunjukkan performa rendah pada skenario berbasis kata kunci, dengan *F1-Score* tidak mencapai 0,1, sedangkan pendekatan berbasis genre menunjukkan *F1-Score* tertinggi sebesar 0,15 pada rekomendasi *top-20*. Penelitian ini menyimpulkan bahwa penggunaan *TF-IDF* dan *Cosine Similarity* dengan ekstraksi kata kunci dari sinopsis belum efektif dalam menghasilkan rekomendasi yang relevan, serta menyarankan pengembangan metode *hybrid* dan perluasan data referensi untuk peningkatan performa sistem di masa mendatang.

Penelitian 6, oleh (Velamentosa & Zuliarso, 2025) yang berjudul “Sistem Rekomendasi Film Menggunakan Metode *Content-Based Filtering*” berfokus pada pengembangan sistem rekomendasi film berdasarkan kesamaan konten, dengan tujuan membantu pengguna menemukan film yang sesuai preferensi mereka secara efisien. Metode yang digunakan adalah *Content-Based Filtering (CBF)* dengan teknik *Natural Language Processing (NLP)*, khususnya *TF-IDF* untuk ekstraksi fitur dan *Cosine Similarity* untuk mengukur kesamaan antar film. *Dataset* yang digunakan terdiri dari 100 film yang

dikumpulkan dari *TMDb* dan *MovieLens*, mencakup atribut seperti judul, genre, sinopsis, sutradara, dan *rating*. Sistem diuji menggunakan metrik evaluasi *Precision*, *Recall*, dan *F1-Score*, dan menghasilkan nilai masing-masing sebesar 85%, 78%, dan 81%. Hasil tersebut menunjukkan bahwa sistem efektif memberikan rekomendasi yang relevan, meskipun masih menghadapi tantangan seperti *over-specialization* dan *cold start*. Penelitian ini memberikan kontribusi signifikan dalam pengembangan sistem rekomendasi berbasis konten untuk industri hiburan digital.

Penelitian 7, oleh (Wiputra & Shandi, 2021) yang berjudul “Perancangan Sistem Rekomendasi Menggunakan Metode *Collaborative Filtering* dengan Studi Kasus Perancangan *Website* Rekomendasi Film” berfokus pada pembangunan sistem rekomendasi film berbasis web dengan tujuan memberikan rekomendasi yang terpersonalisasi berdasarkan preferensi pengguna. Metode yang digunakan adalah *Collaborative Filtering*, baik *user-based* maupun *item-based*, dengan teknik perhitungan kemiripan menggunakan *Euclidean distance* dan prediksi *rating* menggunakan *similarity-based prediction*. *Dataset* berupa data *rating* pengguna terhadap film yang dikumpulkan langsung dari interaksi pengguna di dalam sistem. Sistem diuji dengan pendekatan kualitatif melalui implementasi *website* yang menampilkan rekomendasi berdasarkan kesamaan pengguna maupun kemiripan antar film. Hasil implementasi menunjukkan bahwa sistem mampu mengelompokkan pengguna atau film berdasarkan kesamaan preferensi, serta memberikan rekomendasi relevan tanpa perlu menganalisis konten secara langsung. Penelitian ini berkontribusi dalam menyediakan solusi sistem rekomendasi yang murah, terjangkau, dan dapat diadaptasi untuk berbagai platform layanan digital skala kecil.

Penelitian 8, oleh (Sari et al., 2024) yang berjudul “Implementasi *Content-Based Filtering* Menggunakan *TF-IDF* and *Cosine Similarity* untuk Sistem Rekomendasi Resep Masakan” berfokus pada pengembangan sistem rekomendasi resep masakan berbasis

konten, dengan tujuan untuk membantu pengguna—khususnya ibu rumah tangga—menemukan resep yang sesuai dengan bahan yang tersedia di rumah. Metode yang digunakan adalah *Content-Based Filtering* dengan pendekatan *TF-IDF* untuk pembobotan kata dan *Cosine Similarity* untuk mengukur kemiripan antar resep berdasarkan *query* bahan makanan yang dimasukkan pengguna. *Dataset* yang digunakan terdiri dari 30 data resep yang diambil dari situs masakapahariini.com dan mencakup 311 kata kunci unik hasil *preprocessing*. Sistem diuji menggunakan metrik *Root Mean Square Error (RMSE)*, dan menghasilkan nilai *RMSE* sebesar 0,356 yang menunjukkan tingkat kesalahan prediksi cukup rendah. Hasil tersebut menunjukkan bahwa sistem yang dikembangkan mampu memberikan rekomendasi resep yang akurat dan relevan sesuai kebutuhan pengguna.

Penelitian 9, oleh (Pramesti & Santiyasa, 2022) yang berjudul “Penerapan Metode *Content-Based Filtering* dalam Sistem Rekomendasi Video Game” berfokus pada pengembangan sistem rekomendasi untuk membantu pengguna memilih *video game* berdasarkan *genre* yang sesuai dengan preferensi mereka. Tujuan penelitian ini adalah mengurangi kebingungan pengguna dalam menentukan *game* berikutnya yang akan dimainkan di tengah banyaknya pilihan *video game*. Metode yang digunakan adalah *Content-Based Filtering* dengan pembobotan menggunakan *TF-IDF* dan pengukuran kemiripan menggunakan *Cosine Similarity*. *Dataset* yang digunakan terdiri dari 16.026 data *video game* dari situs *Kaggle*, dengan atribut berupa ID, nama, dan *genre*. Data tersebut diproses melalui tahapan penghapusan data kosong dan duplikat. Sistem diuji menggunakan metrik *Precision* dengan mengambil 10 sampel *game*, menghasilkan nilai rata-rata *Precision* sebesar 87,75%. Hasil tersebut menunjukkan bahwa sistem rekomendasi yang dibangun mampu memberikan saran *game* yang relevan dan sesuai dengan preferensi pengguna, sehingga efektif dalam mempermudah pengguna memilih *video game* yang sesuai.

Penelitian 10, oleh (Noordi et al., 2024) yang berjudul “*Marvel Movie Recommendation System Using Hybrid Item-Based and Content-Based Filtering Methods*” berfokus pada pengembangan sistem rekomendasi film Marvel dengan tujuan memberikan rekomendasi yang relevan dan personal berdasarkan preferensi pengguna. Metode yang digunakan adalah *Hybrid Filtering*, yaitu kombinasi dari *item-based Collaborative Filtering* dan *Content-Based Filtering* dengan pendekatan *TF-IDF* dan *Cosine Similarity*. Data yang digunakan berasal dari *IMDb* dan *Movielens*, terdiri atas dua *dataset* utama, yaitu *marvel_movies.csv* yang memuat informasi film (judul, *genre*, aktor, sutradara, dan sinopsis) serta *user_rating.csv* yang berisi *rating* pengguna terhadap film. Proses pengujian dilakukan melalui analisis *similarity* antar item menggunakan *Pearson correlation (item-based)* dan *Cosine Similarity (content-based)*, serta evaluasi performa sistem menggunakan metrik *precision*. Hasil evaluasi menunjukkan nilai *precision* sebesar 0,8 (80%), *similarity item-based* sebesar 0,68, dan *similarity tertinggi content-based* sebesar 0,14. Temuan ini menunjukkan bahwa model *hybrid* yang digunakan mampu memberikan rekomendasi film Marvel yang relevan dan akurat, meskipun tetap memiliki keterbatasan pada ketergantungan data yang lengkap dan proses komputasi yang kompleks.

Penelitian 11, oleh (Jena et al., 2022) yang berjudul “*Recommendation System for Anime Using Machine Learning Algorithms*” berfokus pada pengembangan sistem rekomendasi *anime* yang dapat menyesuaikan preferensi pengguna dengan tujuan meningkatkan pengalaman menonton melalui rekomendasi yang lebih personal. Penelitian ini menerapkan pendekatan kombinasi dari *Content-Based Filtering*, *Collaborative Filtering* (menggunakan *K-Nearest Neighbors* dan *Singular Value Decomposition*), serta *popularity-based filtering*. *Dataset* yang digunakan diambil dari *Kaggle*, terdiri dari dua bagian: satu berisi 12.294 entri *anime* dan satu lagi mencakup 7.813.737 data *rating* dari 73.516 pengguna. Evaluasi dilakukan dengan menguji efektivitas masing-masing metode, seperti

prediksi akurasi *rating* pada *SVD* dan kemiripan *genre* pada *Content-Based Filtering* menggunakan *Cosine Similarity*. Hasilnya menunjukkan bahwa pendekatan kombinasi memberikan rekomendasi yang relevan dan akurat, seperti pada kasus *anime* “*Kimi no Na wa*” yang berhasil direkomendasikan *anime* serupa secara tepat. Kesimpulannya, sistem yang dirancang menunjukkan kontribusi signifikan terhadap peningkatan akurasi rekomendasi *anime* dengan menggabungkan berbagai pendekatan *machine learning*.

Penelitian 12, oleh (Rianti et al., 2024) yang berjudul “*Machine Learning Journal Article Recommendation System Using Content-Based Filtering*” berfokus pada pengembangan sistem rekomendasi artikel jurnal *machine learning* untuk mempermudah pengguna menemukan referensi yang relevan, khususnya dalam konteks keterbatasan jumlah publikasi di Indonesia. Metode yang digunakan adalah *Content-Based Filtering* dengan algoritma *TF-IDF* untuk ekstraksi fitur dan *Cosine Similarity* untuk pengukuran kemiripan antar artikel. Proses pengembangan mengikuti tahapan *CRISP-DM*, dengan *dataset* berjumlah 100 artikel jurnal *machine learning* berbahasa Indonesia dan Inggris yang diambil dari *Google Scholar* periode 2013–2023. Fitur utama *dataset* mencakup judul, sektor, pendekatan (gabungan metode dan algoritma), serta tautan artikel. Sistem diuji menggunakan metrik *Precision* terhadap lima pendekatan populer, yaitu *K-Means*, *Naive Bayes*, *K-Nearest Neighbor*, *Multiple Linear Regression*, dan *Linear Regression*. Hasil evaluasi menunjukkan bahwa empat pendekatan memperoleh nilai *Precision* sebesar 80% dan satu pendekatan (*KNN*) sebesar 60%, dengan rata-rata *Precision* 76%. Hasil ini menunjukkan bahwa sistem mampu memberikan rekomendasi yang cukup relevan, meskipun keterbatasan jumlah data menghambat kinerja optimal model untuk seluruh pendekatan. Penelitian ini menyimpulkan bahwa metode *Content-Based Filtering* dapat diterapkan secara efektif untuk rekomendasi artikel ilmiah, serta menyarankan perluasan *dataset* dan eksplorasi pendekatan *hybrid* untuk pengembangan lanjutan.

Penelitian 13, oleh (Salmani & Kulkarni, 2021) yang berjudul “*Hybrid Movie Recommendation System Using Machine Learning*” berfokus pada peningkatan akurasi sistem rekomendasi film dengan tujuan memberikan saran tontonan yang lebih relevan dan personal melalui kombinasi berbagai algoritma *filtering*. Metode yang digunakan adalah pendekatan *hybrid* dengan menggabungkan *Content-Based Filtering* berbasis genre menggunakan *TF-IDF* dan *Cosine Similarity*, serta *Collaborative Filtering* dengan algoritma *SVD* dan *SVD++*. Dataset yang digunakan adalah *MovieLens-100k*, terdiri dari file *movies.csv* (9.744 baris data film) dan *ratings.csv* (100.837 data *rating* pengguna). Sistem diuji menggunakan metrik evaluasi *Root Mean Square Error (RMSE)* dan *Hit Ratio*. Hasil pengujian menunjukkan bahwa model *content-based* memiliki *hit ratio* sebesar 0,9325, *item-based Collaborative Filtering* sebesar 1,0, *user-based* sebesar 0,88, dan model *SVD* serta *SVD++* masing-masing menghasilkan *RMSE* sebesar 0,87 dan 0,9387. Model *hybrid* yang menggabungkan *Content-Based* dan *SVD* menghasilkan *RMSE* terbaik sebesar 0,76, menunjukkan peningkatan akurasi sekitar 20% dibandingkan model individual. Penelitian ini menyimpulkan bahwa pendekatan *hybrid* efektif dalam meningkatkan performa sistem rekomendasi, serta mampu mengatasi kelemahan model tunggal seperti masalah *cold-start* dan ketergantungan pada data eksplisit.

Penelitian 14, oleh Addini (Yusmar et al., 2021) yang berjudul “*Restaurant Recommender System Using Item-Based Collaborative Filtering and Adjusted Cosine Algorithm Similarity*” berfokus pada pengembangan sistem rekomendasi restoran dengan tujuan membantu pengguna dalam memilih restoran sesuai preferensi mereka secara efisien. Metode yang digunakan adalah *item-based Collaborative Filtering* dengan algoritma *Adjusted Cosine Similarity* dan pendekatan prediksi berbasis *weighted sum*. Data yang digunakan terdiri dari 28 restoran dan *rating* dari 40 pengguna yang dikumpulkan melalui kuesioner menggunakan teknik *purposive sampling*. Sistem diuji menggunakan dua metrik

evaluasi, yaitu *Precision* dan *Mean Absolute Error (MAE)*, dalam tiga skenario jumlah pengguna (5, 20, dan 40). Hasil terbaik diperoleh pada skenario dengan 40 pengguna, yaitu *Precision* sebesar 0,732 dan *MAE* sebesar 0,623. Hal ini menunjukkan bahwa sistem semakin akurat dengan jumlah data *rating* yang lebih besar. Kesimpulannya, metode yang digunakan terbukti efektif dalam menghasilkan rekomendasi dengan tingkat akurasi yang baik dan tingkat kesalahan yang rendah, serta layak diterapkan dalam sistem rekomendasi restoran.

Penelitian 15, oleh (Anwar & Uma, 2021) yang berjudul “*Comparative Study of Recommender System Approaches and Movie Recommendation Using Collaborative Filtering*” berfokus pada pengembangan dan evaluasi sistem rekomendasi film dengan tujuan membandingkan efektivitas beberapa pendekatan *Collaborative Filtering (CF)*. Metode yang digunakan adalah *Collaborative Filtering* dengan empat pendekatan utama, yaitu *K-Nearest Neighbor (K-NN)*, *Singular Value Decomposition (SVD)*, *SVD++*, dan *Co-Clustering*. Dataset yang digunakan adalah *MovieLens 100K*, yang terdiri dari 100.000 *rating* terhadap 1.682 film oleh 943 pengguna. Sistem diuji menggunakan dua metrik evaluasi utama, yaitu *Root Mean Square Error (RMSE)* dan *Mean Absolute Error (MAE)*. Hasil evaluasi menunjukkan bahwa model *SVD++* menghasilkan kinerja terbaik dengan *RMSE* sebesar 0,9201 dan *MAE* sebesar 0,7219, dibandingkan *K-NN* (*RMSE* 0,979, *MAE* 0,7732), *SVD* (*RMSE* 0,9358, *MAE* 0,737), dan *Co-Clustering* (*RMSE* 0,9675, *MAE* 0,7582). Kesimpulannya, pendekatan *SVD++* efektif dalam meningkatkan akurasi prediksi *rating* dan mengurangi *error*, serta mampu mengatasi sebagian masalah *cold start* dan *data sparsity* dalam sistem rekomendasi film.

2.7 Kesimpulan Tinjauan Studi

Penelitian terkait sistem rekomendasi film dan *anime* menunjukkan perkembangan yang signifikan melalui penerapan berbagai pendekatan, seperti *Collaborative Filtering*

(CF), *Content-Based Filtering* (CBF), serta kombinasi keduanya dalam *Hybrid Filtering*. Pendekatan CF banyak memanfaatkan algoritma, antara lain *Singular Value Decomposition* (SVD), *SVD++*, *Alternating Least Squares* (ALS), dan *K-Nearest Neighbor* (K-NN), yang terbukti efektif dalam mempersonalisasi rekomendasi berdasarkan riwayat interaksi pengguna, meskipun tetap menghadapi tantangan seperti *cold start* dan *data sparsity*. Sementara itu, pendekatan CBF umumnya mengandalkan *Term Frequency-Inverse Document Frequency* (TF-IDF) dan *Cosine Similarity* untuk mengekstraksi kesamaan dari konten film atau *anime*, seperti sinopsis, genre, hingga daftar kreator, namun memiliki kelemahan berupa risiko *over-specialization*.

Metode *Hybrid Filtering* yang menggabungkan keunggulan CF dan CBF terbukti menjadi solusi efektif dalam meningkatkan akurasi dan relevansi sistem rekomendasi, sebagaimana ditunjukkan dalam penelitian (Arfisko & Wibowo, 2022; Mukti & Mardhiyah, 2022; Salmani & Kulkarni, 2021). Evaluasi sistem dilakukan melalui berbagai metrik seperti *RMSE*, *MAE*, *Precision*, *Recall*, dan *F1-Score*, yang masing-masing memberikan gambaran menyeluruh atas performa dan stabilitas model. Temuan menarik dari beberapa studi juga menyoroti pentingnya struktur data, skala *Dataset*, serta penggunaan teknologi seperti *Flask* dan evaluasi *A/B testing* dalam membuktikan efektivitas sistem.

Berdasarkan tinjauan studi, dapat disimpulkan bahwa pendekatan *hybrid* merupakan strategi paling menjanjikan untuk mengatasi keterbatasan metode tunggal dan menghasilkan sistem rekomendasi yang lebih personal, akurat, dan efisien. Oleh karena itu, penelitian ini akan mengadopsi *Hybrid Filtering* berbasis *SVD* (*Collaborative Filtering*) dan *TF-IDF Cosine Similarity* (*Content-Based Filtering*), serta menerapkan skema *Weighted Hybrid*. Pendekatan ini diharapkan mampu memberikan kontribusi baru dalam pengembangan sistem rekomendasi yang tidak hanya akurat, namun juga adaptif terhadap karakteristik pengguna dan dinamika konten digital.

BAB III

METODOLOGI PENELITIAN

3.1 Analisa Kebutuhan

Analisis kebutuhan merupakan langkah awal yang esensial dalam mengidentifikasi setiap komponen yang diperlukan agar proses penelitian dan pengembangan sistem rekomendasi dapat berjalan sesuai dengan tujuan yang telah ditetapkan. Dalam konteks penelitian ini, proses analisis kebutuhan melibatkan dua aktor utama, yaitu *user* (pengguna) dan *admin* (pengelola sistem). Meskipun memiliki tanggung jawab yang berbeda, kedua aktor tersebut berperan saling melengkapi dalam mendukung operasional sistem rekomendasi berbasis *Hybrid Filtering*. Rincian kebutuhan dari masing-masing aktor tersebut dapat dilihat pada Tabel 3.1 di bawah ini:

Tabel 3.1 Tabel Analisa Kebutuhan

No	Aktor	Keterangan
1	User	- <i>Login</i> untuk mendapatkan rekomendasi <i>anime</i> personal.
		- Memberi <i>rating</i> pada <i>anime</i> yang sudah ditonton.
		- Melihat rekomendasi berbasis popularitas meskipun belum <i>login</i> .
		- Melihat rekomendasi populer (dapat diakses tanpa <i>login</i>).
		- Melihat <i>rating</i> yang sudah diberikan sebelumnya jika sudah <i>login</i>
2	Admin	- Mengelola data <i>anime</i> dan pengguna.
		- Meninjau dan memvalidasi <i>rating</i> pengguna.
		- Melihat hasil evaluasi sistem rekomendasi.
		- Melakukan <i>retraining</i> jika performa menurun.

Tabel di atas menunjukkan bahwa sistem harus mampu mengakomodasi dua kebutuhan utama: kebutuhan interaktif dari sisi pengguna, yang berkaitan dengan fungsionalitas akses dan preferensi konten; dan kebutuhan pengelolaan dari sisi *admin*, yang bertugas memastikan data tetap valid serta sistem tetap berjalan optimal. Kebutuhan-kebutuhan tersebut menjadi dasar dalam penyusunan kebutuhan fungsional dan nonfungsional sistem, yang dijabarkan pada subbab berikut.

3.1.1 Kebutuhan Fungsional

Kebutuhan fungsional adalah fitur atau layanan yang wajib disediakan oleh sistem untuk memenuhi kebutuhan pengguna dan *admin*. Adapun kebutuhan fungsional dalam penelitian ini adalah sebagai berikut:

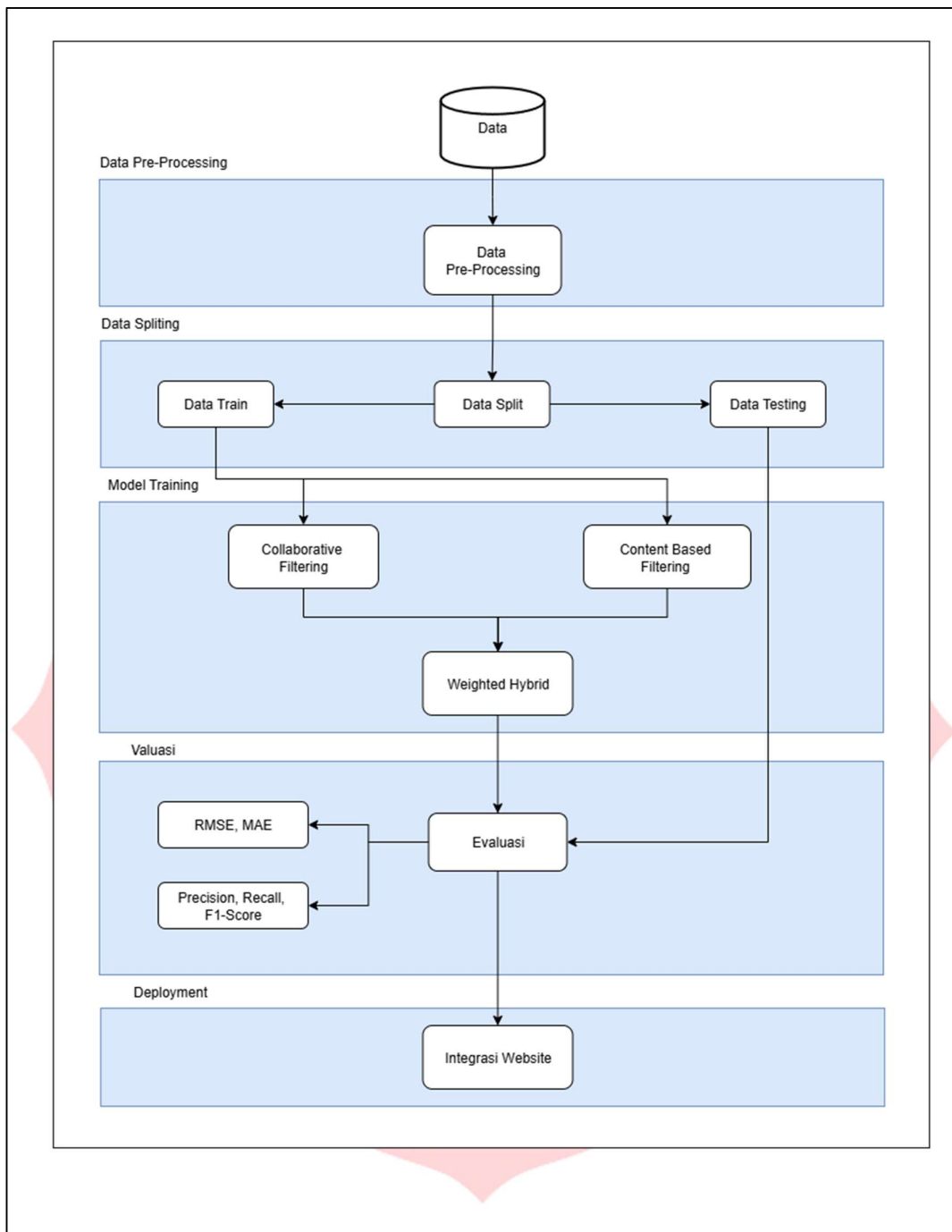
- a. Pengguna dapat melakukan *login* dan *logout* untuk mendapatkan rekomendasi yang dipersonalisasi.
- b. Pengguna dapat memberikan *rating* terhadap *anime* yang telah ditonton.
- c. Sistem dapat menghasilkan rekomendasi berdasarkan kombinasi metode *Collaborative Filtering* dan *Content-Based Filtering*.
- d. Pengguna yang belum *login* tetap dapat melihat rekomendasi berbasis popularitas (*popularity-based*).
- e. *Admin* dapat melakukan pengelolaan data *anime* dan data pengguna, serta mengakses laporan evaluasi performa sistem.

3.1.2 Kebutuhan Non Fungsional

Kebutuhan nonfungsional berkaitan dengan kualitas sistem secara keseluruhan dan spesifikasi teknis yang harus dipenuhi. Kebutuhan nonfungsional sistem rekomendasi ini adalah sebagai berikut:

- a. Sistem memiliki akurasi prediksi yang tinggi, dengan nilai *RMSE* di bawah 1,7.
- b. Model rekomendasi harus mampu menangani permasalahan *cold start* dan *data sparsity*.
- c. Sistem dapat dijalankan secara lokal pada perangkat dengan spesifikasi minimum: RAM 8 GB, prosesor *Intel Core i5*, dan tanpa *GPU* tambahan.

3.2 Kerangka Pemikiran



Gambar 3.1 Kerangka Pemikiran

3.3 Data Understanding

Penelitian ini menggunakan *dataset* berjudul “*Anime Recommendation Database*” yang tersedia secara publik di platform *Kaggle*. *Dataset* tersebut memuat data interaksi antara 73.516 pengguna dengan 12.294 judul *anime*, dan terdiri dari dua file utama, yaitu:

3.3.1 *Anime.csv*

Tabel ini memuat metadata terkait judul-judul *anime* yang tercatat di *MyAnimeList*, dengan total entri mencapai kurang lebih 12.294. Setiap baris pada tabel merepresentasikan satu judul *anime*, dilengkapi dengan informasi seperti *ID*, judul, *genre*, tipe (*TV*, *Movie*, *OVA*, dll.), skor rata-rata, jumlah episode, dan jumlah pengguna yang menambahkan *anime* tersebut ke daftar mereka. Tabel ini berfungsi sebagai sumber utama dalam penerapan *Content-Based Filtering* serta menjadi referensi dalam proses penggabungan dan transformasi data.

Tabel 3.2 Informasi *Dataset anime.csv*

Kolom	Deskripsi
<i>anime_id</i>	ID unik setiap <i>anime</i>
<i>name</i>	Judul <i>anime</i>
<i>genre</i>	Daftar <i>genre</i> (misalnya: <i>Action</i> , <i>Adventure</i> , <i>Comedy</i> , dll)
<i>type</i>	Tipe <i>anime</i> (misalnya: <i>TV</i> , <i>Movie</i> , <i>OVA</i> , dll.)
<i>episodes</i>	Jumlah <i>episode</i>
<i>rating</i>	Skor rata-rata dari pengguna
<i>members</i>	Jumlah <i>user</i> yang menambahkan <i>anime</i> ke daftar mereka

Tabel 3.3 Sampel Dataset anime.csv

No	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9,37	200.630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Military...	TV	64	9,26	793.665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9,25	114.262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9,17	673572
4	9969	Gintama'	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9,16	151.266
...	...	...	...	...	...	...	...
12.289	9316	Toushindai My Lover: Minami tai Mecha-Minami	Hentai	OVA	1	4,15	211
12.290	5543	Under World	Hentai	OVA	1	4,28	183
12.291	5621	Violence Gekiga David no Hoshi	Hentai	OVA	4	4,88	219
12.292	6133	Violence Gekiga Shin David no Hoshi: Inma Dens...	Hentai	OVA	1	4,98	175
12.293	26081	Yasuji no Pornorama: Yacchimae!!	Hentai	Movie	1	5.46	142

3.3.2 Rating.csv

Tabel ini berisi data interaksi antara pengguna dan judul *anime*, berupa nilai *rating* yang diberikan oleh pengguna terhadap *anime* tertentu. Total entri pada tabel ini mencapai lebih dari 7 juta baris, yang mencakup *ID* pengguna, *ID anime*, serta nilai *rating* dengan skala 1–10. Nilai -1 menandakan bahwa pengguna belum memberikan *rating*. Tabel ini menjadi komponen utama dalam pembentukan model *Collaborative Filtering* dan juga digunakan sebagai dasar dalam evaluasi prediksi sistem rekomendasi.

Tabel 3.4 informasi *Dataset rating.csv*

Kolom	Deskripsi
<i>user_id</i>	ID unik pengguna
<i>anime_id</i>	ID unik <i>anime</i>
<i>rating</i>	Skor yang diberikan oleh <i>user</i>

Tabel 3.5 Sampel *Dataset rating.csv*

No	<i>user_id</i>	<i>anime_id</i>	<i>rating</i>
0	1	20	-1
1	1	24	-1
2	1	79	-1
3	1	226	-1
4	1	241	-1
...	...	...	...
7813732	73515	16512	7
7813733	73515	17187	9
7813734	73515	22145	10
7813735	73516	790	9
7813736	73516	8074	9

Dataset yang digunakan adalah *dataset* sekunder dan telah banyak digunakan di beberapa penelitian sebelumnya (Girsang et al., 2020). *Dataset* ini tersedia di *Kaggle* pada tautan : <https://www.Kaggle.com/Datasets/CooperUnion/Anime-recommendations-database>.

3.3.3 Merge Dataset

Untuk keperluan analisis dan pemodelan sistem rekomendasi, kedua file *dataset—rating.csv* dan *anime.csv*—digabungkan menggunakan atribut kunci *anime_id*. Proses ini menghasilkan sebuah tabel gabungan yang mengintegrasikan data interaksi pengguna (*user_id* dan *rating*) dengan informasi konten *anime* (*judul*, *genre*, *tipe*, dan sebagainya). Hasil *merge* ini menjadi dasar bagi penerapan metode *Hybrid*

Filtering, karena memungkinkan sistem untuk memanfaatkan baik data historis pengguna maupun karakteristik konten dari setiap *anime* yang di-*rating*.

Tabel 3.6 Informasi *Dataset merge.csv*

Kolom	Deskripsi
<i>user_id</i>	ID unik dari pengguna yang memberikan <i>rating</i> terhadap <i>anime</i> .
<i>anime_id</i>	ID unik untuk setiap judul <i>anime</i> , menjadi kunci penghubung antar <i>dataset</i> .
<i>rating_x</i>	Skor yang diberikan oleh pengguna terhadap <i>anime</i> tertentu dalam skala 1–10. Nilai -1 menunjukkan bahwa pengguna belum memberikan <i>rating</i> .
<i>name</i>	Judul dari <i>anime</i> yang di- <i>rating</i> oleh pengguna.
<i>genre</i>	Daftar <i>genre</i> dari <i>anime</i> , seperti <i>Action</i> , <i>Comedy</i> , <i>Romance</i> , dan sebagainya.
<i>type</i>	Tipe format <i>anime</i> , misalnya <i>TV</i> , <i>Movie</i> , <i>OVA</i> , <i>Special</i> , atau <i>Music</i> .
<i>episodes</i>	Jumlah total <i>episode</i> dari <i>anime</i> yang bersangkutan. Nilai ini dapat berupa angka atau “ <i>Unknown</i> ” jika tidak tersedia.
<i>rating_y</i>	Skor rata-rata dari semua pengguna untuk <i>anime</i> tersebut, dihitung oleh platform sumber (<i>MyAnimeList</i>), dengan skala 1–10.
<i>members</i>	Jumlah total pengguna yang menambahkan <i>anime</i> tersebut ke dalam daftar mereka.

Tabel 3.7 Sampel *Dataset merge.csv*

No	<i>user_id</i>	<i>anime_id</i>	<i>rating_x</i>	<i>name</i>	<i>genre</i>	<i>type</i>	<i>episodes</i>	<i>rating_y</i>	<i>members</i>
1	1	20	-1	<i>Naruto</i>	<i>Action, Comedy, Martial Arts, Shounen, Super Power</i>	<i>TV</i>	220	7.81	683.297
2	1	24	-1	<i>School Rumble</i>	<i>Comedy, Romance, School, Shounen</i>	<i>TV</i>	26	8.06	178.553
3	1	79	-1	<i>Shuffle!</i>	<i>Comedy, Drama, Ecchi, Fantasy, Harem, Magic, Romance, School</i>	<i>TV</i>	24	7.31	158.772
4	1	226	-1	<i>Elfen Lied</i>	<i>Action, Drama, Horror, Psychological, Romance, Seinen, Supernatural</i>	<i>TV</i>	13	7.85	623.511
5	1	241	-1	<i>Girls Bravo: First Season</i>	<i>Comedy, Ecchi, Fantasy, Harem, Romance, School</i>	<i>TV</i>	11	6.69	84.395

...	...	...	...	...	...	...	...	...	...
7,8 13, 73 3	73 51 5	165 12	7	<i>Devil Survivor 2 The Animation</i>	<i>Action, Demons, Supernatural</i>	<i>TV</i>	13	7.0 6	101 .26 6
7,8 13, 73 4	73 51 5	171 87	9	<i>Ghost in the Shell: Arise - Border:1 Ghost Pain</i>	<i>Mecha, Police, Psychological, Sci- Fi</i>	<i>Mov ie</i>	1	7.6 4	31. 747
7,8 13, 73 5	73 51 5	221 45	10	<i>Kuroshitsuji: Book of Circus</i>	<i>Comedy, Demons, Fantasy, Historical, Shounen, Supernatural</i>	<i>TV</i>	10	8.3 7	122 .89 5
78 13 73 5	73 51 6	790	9	<i>Ergo Proxy</i>	<i>Mystery, Psychological, Sci- Fi</i>	<i>TV</i>	23	8.0 3	265 005 .0
78 13 73 6	73 51 6	807 4	9	<i>Highschool of the Dead</i>	<i>Action, Ecchi, Horror, Supernatural</i>	<i>TV</i>	12	7.4 6	535 892 .0

3.4 Pre-processing Data

Pre-processing Data merupakan tahap awal dalam proses pengolahan data yang bertujuan untuk menyiapkan data mentah agar layak digunakan dalam proses pelatihan dan pemodelan sistem rekomendasi. Tahap ini sangat krusial untuk memastikan bahwa data yang digunakan memiliki kualitas yang baik, bersih dari *noise*, konsisten dalam format, serta sesuai dengan kebutuhan *input* dari algoritma *machine learning*.

Dalam penelitian ini, proses *pre-processing* dilakukan terhadap dua *dataset* utama yang telah digabung sebelumnya, yaitu *dataset rating* pengguna (*rating.csv*) dan *dataset informasi anime* (*anime.csv*). Proses *pre-processing* dilakukan secara bertahap, meliputi penghapusan nilai tidak valid, normalisasi teks, pembersihan *genre* tidak relevan, penanganan nilai kosong, seleksi pengguna aktif, serta penyimpanan *dataset* akhir. Tahapan lengkapnya dijelaskan sebagai berikut:

- a. Penghapusan *Rating* Tidak Valid : Pada *dataset* hasil penggabungan, terdapat nilai *rating* = -1 yang menandakan bahwa pengguna tidak memberikan penilaian terhadap suatu *anime*. Nilai-nilai ini dihapus karena tidak memberikan informasi yang berguna, terutama bagi pendekatan *Collaborative Filtering* yang memerlukan data eksplisit sebagai dasar prediksi.
- b. Normalisasi Teks : Atribut teks seperti *name*, *genre*, dan *type* dinormalisasi dengan cara mengubah seluruh karakter menjadi huruf kecil (*lowercase*), menghapus tanda baca, serta karakter non-alfabet. Langkah ini dilakukan agar *input* teks menjadi konsisten dan dapat diproses secara optimal dalam tahap ekstraksi fitur berbasis *TF-IDF*.
- c. Pembersihan *Genre* Tidak Relevan : Beberapa *genre* seperti *unknown*, *other*, dan *none* dianggap tidak relevan dalam konteks sistem rekomendasi publik, sehingga dihapus dari kolom *genre*. Apabila setelah proses ini sebuah entri tidak memiliki *genre*, maka baris tersebut juga dihapus dari *dataset*.
- d. Penghapusan Nilai Kosong pada Kolom Penting : *Dataset* dibersihkan dari entri yang memiliki nilai kosong (*missing values*) pada kolom-kolom kunci seperti *name*, *genre*, *type*, dan *rating_y*. Hal ini dilakukan untuk memastikan hanya data yang valid dan lengkap yang digunakan dalam proses pelatihan model.
- e. Seleksi Pengguna aktif : Untuk meningkatkan kualitas hasil rekomendasi dan mengurangi masalah *data sparsity*, hanya pengguna yang telah memberikan minimal 50 *rating* yang dipertahankan dalam *dataset*. Langkah ini bertujuan agar matriks interaksi yang dibentuk memiliki kepadatan yang cukup untuk dilatih oleh model.
- f. Penyimpanan *Dataset* : *Dataset* yang telah melalui seluruh proses *pre-processing* disimpan kembali dalam bentuk file *merged_clean.csv*. *Dataset* ini menjadi basis utama dalam pelatihan sistem rekomendasi berbasis *Hybrid Filtering* yang dikembangkan dalam penelitian ini.

Dengan melalui proses *pre-processing* ini, data menjadi lebih terstruktur dan representatif, sehingga dapat meningkatkan performa model dalam memberikan rekomendasi yang lebih akurat dan relevan kepada pengguna.

Tabel 3.8 Sampel *Dataset merge_clean.csv*

No	<i>user_id</i>	<i>anime_id</i>	<i>rating_x</i>	<i>name</i>	<i>genre</i>	<i>type</i>	<i>episodes</i>	<i>rating_y</i>	<i>members</i>
1	1	11	8	<i>naruto</i>	<i>action, comedy, martial arts, shounen, super power</i>	<i>tv</i>	220	7.81	683297
2	1	132	6	<i>shaman king</i>	<i>action, adventure, comedy, drama, shounen, supernatural</i>	<i>tv</i>	64	7.83	169517
3	1	148	9	<i>slam dunk</i>	<i>comedy, drama, school, shounen, sports</i>	<i>tv</i>	101	8.56	82570
4	1	176	10	<i>sen to chihiro no kamikakushi</i>	<i>adventure, drama, supernatural</i>	<i>movie</i>	1	8.93	466254
5	1	200	9	<i>dragon ball gt</i>	<i>action, adventure, comedy, fantasy, magic, sci-fi</i>	<i>tv</i>	64	6.72	226625
...	...	...	...	...	...	...	...	...	...
5,646,459	32888	6139	7	<i>btroom</i>	<i>action, psychological, sci-fi, seinen</i>	<i>tv</i>	12	7.68	329561
5,646,460	32888	6413	7	<i>devil survivor 2 the animation</i>	<i>action, demons, supernatural</i>	<i>tv</i>	13	7.06	101266
5,646,461	32888	6513	9	<i>ghost in the shell arise border1 ghost pain</i>	<i>mecha, police, psychological, sci-fi</i>	<i>movie</i>	1	7.64	31747
5,646,462	32888	7161	10	<i>kurohitsuji book of circus</i>	<i>comedy, demons, fantasy, historical, shounen, supernatural</i>	<i>tv</i>	10	8.37	122895

Tabel 3.8 berikut menampilkan cuplikan hasil akhir *dataset* setelah melalui seluruh tahapan *pre-processing*, seperti penghapusan *rating* tidak valid, pembersihan nilai kosong, normalisasi teks, serta penggabungan data interaksi dan konten. Data yang ditampilkan telah siap digunakan dalam proses pemodelan sistem rekomendasi berbasis *Hybrid Filtering*.

3.5 Data Splitting

Pada tahap ini, *dataset* hasil *pre-processing* dibagi menjadi dua *subset* utama, yaitu data pelatihan (*training set*) dan data pengujian (*testing set*), dengan proporsi 80% untuk pelatihan dan 20% untuk pengujian. Pembagian dilakukan menggunakan metode *holdout* per pengguna (*user-level splitting*), di mana interaksi setiap pengguna dibagi secara acak namun tetap mempertahankan identitas pengguna yang sama di kedua *subset*.

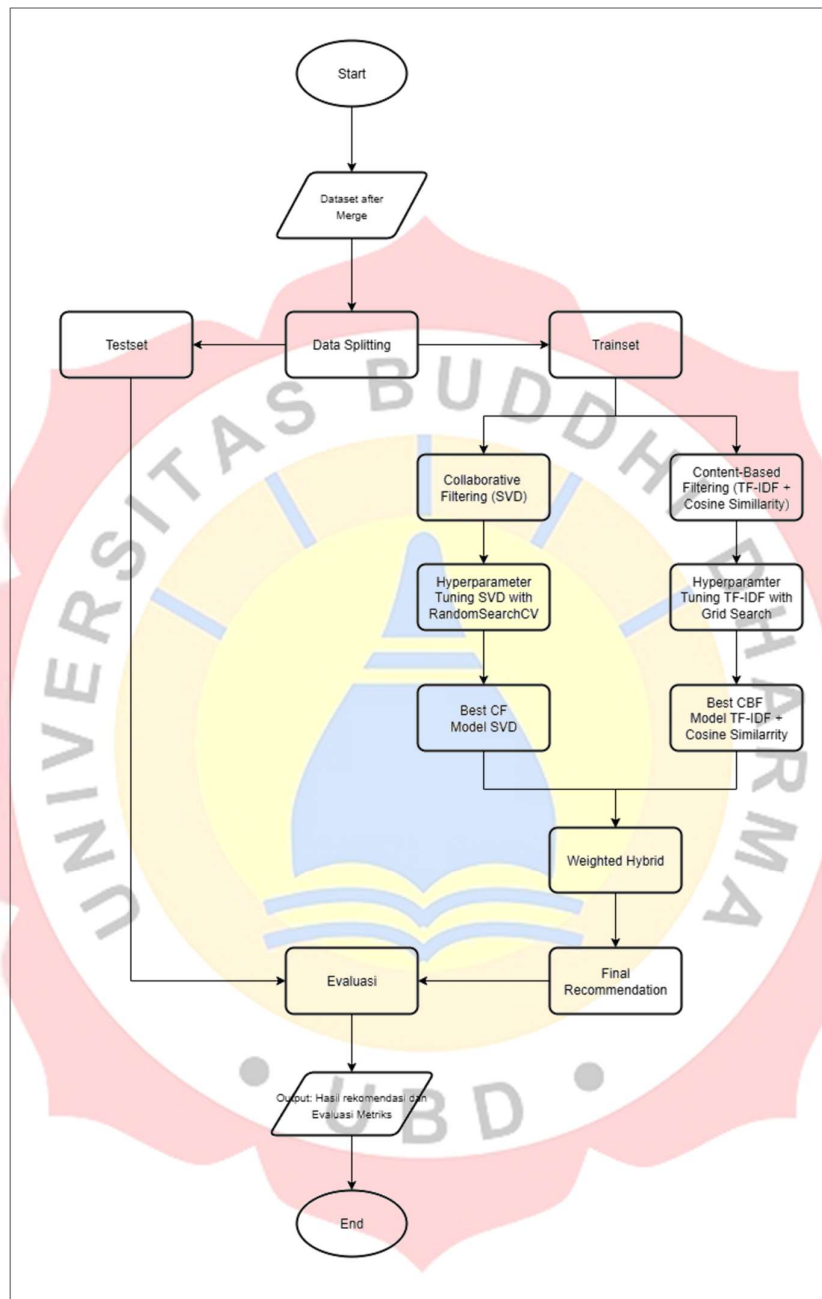
Pendekatan ini dipilih untuk memastikan bahwa model dapat dilatih dan diuji dengan representasi interaksi yang proporsional pada setiap pengguna, serta menghindari kebocoran data antar *subset*. Dengan demikian, data pelatihan digunakan untuk mengenali pola preferensi pengguna berdasarkan histori interaksi dan karakteristik konten *anime*, sedangkan data pengujian digunakan untuk mengevaluasi kemampuan generalisasi model terhadap data baru yang belum pernah dilihat sebelumnya. Distribusi hasil pembagian data ditunjukkan pada Tabel 3.9 berikut ini:

Tabel 3.9 Distribusi *Training* dan *Testing*

<i>Subset</i>	Jumlah Interaksi	Jumlah <i>User</i>	Jumlah <i>Anime</i>
Data <i>Training</i>	4.503.940	32.888	8.941
Data <i>Testing</i>	1.142.523	32.888	7.686
Total	5.646.463	32.888	8.941

3.6 Model Training

3.6.1 Alur Proses Pelatihan Model



Gambar 3.2 Alur Proses Pelatihan Model

3.6.2 Hyperparameter Tuning

Proses *Hyperparameter Tuning* merupakan langkah penting dalam pengembangan sistem rekomendasi, karena dapat memengaruhi akurasi model dalam memprediksi *rating*

maupun memberikan rekomendasi. *Hyperparameter* adalah parameter yang tidak dipelajari langsung oleh model dari data, tetapi ditentukan secara manual dan diuji melalui percobaan sistematis untuk mendapatkan kombinasi terbaik.

Dalam penelitian ini, *tuning* dilakukan pada dua model utama: *Collaborative Filtering* menggunakan algoritma *Singular Value Decomposition (SVD)*, dan *Content-Based Filtering* menggunakan *TF-IDF Vectorizer*. *Tuning* pada *SVD* dilakukan dengan metode *RandomizedSearchCV*, yaitu proses pencarian parameter secara acak dari ruang parameter tertentu, menggunakan validasi silang (*cross-validation*) 5-fold dan dua metrik evaluasi, yaitu *RMSE* dan *MAE*.

Adapun ruang parameter yang diuji pada model *SVD* meliputi jumlah dimensi laten (*n_factors*), jumlah iterasi pelatihan (*n_epochs*), laju pembelajaran (*lr_all*), dan tingkat regularisasi (*reg_all*). Rincian ruang parameter tersebut ditampilkan pada Tabel 3.11. Proses ini bertujuan untuk mencari kombinasi parameter yang menghasilkan nilai *RMSE* serendah mungkin pada data validasi.

Untuk model *TF-IDF Vectorizer*, *tuning* dilakukan terhadap parameter yang memengaruhi pembobotan kata, seperti batas minimum dan maksimum dokumen (*min_df*, *max_df*), ukuran *n-gram* (*ngram_range*), dan jumlah fitur maksimum (*max_features*). Tujuan *tuning* ini adalah untuk memaksimalkan kualitas perhitungan kemiripan antar *anime* menggunakan *Cosine Similarity*.

Tabel 3.10 Konfigurasi Parameter *RandomizedSearchCV* Model *SVD*

<i>Hyperparameter</i>	Nilai
<i>estimator</i>	[<i>SVD()</i>]
<i>param_distributions</i>	<i>param_grid</i>
<i>n_iter</i>	100
<i>scoring</i>	[<i>RMSE</i> , <i>mae</i>]
<i>cv</i>	<i>K-Fold</i> (<i>n_splits</i> = 5)
<i>verbose</i>	1

<i>n_jobs</i>	-1
<i>random_state</i>	42

Tabel 3.11 Parameter *Grid Model SVD*

<i>Hyperparameter</i>	Nilai
<i>n_factors</i>	50, 100, 150, 200
<i>n_epochs</i>	20, 30, 50, 100
<i>lr_all</i>	0.002, 0.005, 0.01
<i>reg_all</i>	0.02, 0.05, 0.1

Tabel 3.12 Parameter *Grid* untuk *Content-Based Filtering Model TF-IDF*

Hyperparameter	Nilai
<i>min_df</i>	2, 3
<i>max_df</i>	0.85, 0.9
<i>ngram_range</i>	(1,1), (1,2)
<i>max_features</i>	800, 1200, 1500
<i>stop_words</i>	'english'

3.6.3 Collaborative Filtering

Collaborative Filtering (CF) merupakan pendekatan rekomendasi yang didasarkan pada pola interaksi pengguna terhadap *item* tertentu, tanpa memperhatikan konten dari *item* tersebut. Dalam penelitian ini, pendekatan *CF* digunakan untuk memodelkan kesamaan preferensi antar pengguna terhadap judul *anime*, dengan memanfaatkan metode *matrix factorization* berbasis *Singular Value Decomposition (SVD)*.

Penerapan model *CF* dilakukan menggunakan pustaka *Surprise*, dengan menggunakan kolom *user_id*, *anime_id*, dan *rating_x* dari *dataset training* sebagai *input*. Tabel 3.13 menunjukkan hasil penyaringan data yang digunakan untuk pelatihan model *CF*:

Tabel 3.13 Hasil Penyaringan untuk model CF

Kolom	Non-null Count	Dtype
<i>user_id</i>	4.508.393 non-null	<i>int64</i>
<i>anime_id</i>	4.508.393 non-null	<i>int64</i>
<i>rating_x</i>	4.508.393 non-null	<i>int64</i>

Pengembangan model *Collaborative Filtering* dilakukan melalui sejumlah tahapan sistematis sebagai berikut:

1. *Dataset* hasil penyaringan digunakan untuk membentuk matriks *user-item*, di mana setiap baris mewakili pengguna dan kolom mewakili *anime*. Nilai dalam matriks merupakan *rating* eksplisit dari pengguna terhadap *anime* tertentu, sedangkan sel kosong menunjukkan absennya interaksi.
2. Matriks *user-item* kemudian didekomposisi menggunakan metode *SVD*, menjadi tiga buah matriks terpisah yang merepresentasikan relasi laten antara pengguna dan *item*.
3. Matriks hasil dekomposisi digunakan untuk merekonstruksi nilai prediksi *rating* yang belum tersedia. Nilai prediksi ini akan digunakan untuk menyusun daftar rekomendasi *anime* bagi masing-masing pengguna.
4. Model *SVD* diimplementasikan dengan parameter utama yang ditentukan melalui proses *RandomizedSearchCV*, meliputi:
 - a. *n_factors* : jumlah *latent features*,
 - b. *n_epochs*: jumlah iterasi pelatihan,
 - c. *lr_all*: *learning rate*,
 - d. *reg_all*: regularisasi.

Proses pelatihan dilakukan menggunakan validasi silang *5-fold* untuk mengevaluasi performa model pada berbagai kombinasi parameter.

3.6.4 Content-Based Filtering

Content-Based Filtering (CBF) adalah metode dalam sistem rekomendasi yang menyajikan saran kepada pengguna berdasarkan tingkat kemiripan konten antar *item*. Pendekatan ini menitikberatkan pada karakteristik dari *item* yang sebelumnya telah diminati atau diberi *rating* tinggi oleh pengguna. Pada penelitian ini, *CBF* diterapkan untuk menilai kesamaan antara *anime* dengan memanfaatkan berbagai informasi seperti judul (*name*), *genre*, dan jenis *anime* (*type*).

Dalam tahap ini, perhatian utama diarahkan pada kolom-kolom dalam *dataset* yang merepresentasikan fitur konten *anime*, antara lain *genre* dan *type*, serta data *metadata* tambahan seperti *episodes* dan *members* apabila diperlukan. Seluruh fitur teks yang tersedia kemudian diproses lebih lanjut dan diubah ke dalam bentuk numerik agar kemiripannya dapat dihitung secara matematis. Tabel 3.14 menunjukkan hasil penyaringan.

Tabel 3.14 Hasil penyaringan untuk Model CBF

Kolom	Non-null Count	Dtype
<i>anime_id</i>	8.941 non-null	<i>int64</i>
<i>name</i>	8.941 non-null	<i>object</i>
<i>genre</i>	8.941 non-null	<i>object</i>
<i>type</i>	8.941 non-null	<i>object</i>

Adapun langkah-langkah pengembangan model *Content-Based Filtering (CBF)* adalah sebagai berikut:

1. Penggabungan Fitur Konten : Kolom *name*, *genre*, dan *type* digabung menjadi satu kolom teks baru sebagai representasi deskripsi konten *anime*. Tujuannya

adalah untuk menangkap kombinasi semantik dari *genre*, judul, dan format *anime* dalam satu representasi.

2. Vektorisasi Teks Menggunakan *TF-IDF Vectorizer* : Representasi teks hasil penggabungan kemudian diproses menggunakan *TF-IDF Vectorizer* dari pustaka *Scikit-learn*. *TF-IDF* digunakan untuk mengubah teks menjadi vektor angka yang mencerminkan pentingnya sebuah kata dalam dokumen tertentu relatif terhadap seluruh *corpus*.
3. Perhitungan *Cosine Similarity* : Setelah proses vektorisasi, dihitung nilai *cosine similarity* antar *item* untuk menentukan sejauh mana kesamaan antara dua *anime* berdasarkan fitur kontennya. Nilai *similarity* disimpan dalam bentuk matriks segitiga simetris.
4. Penyusunan Daftar *Anime* Serupa : Untuk setiap *anime*, sistem menyimpan daftar *anime* lain yang paling mirip secara konten berdasarkan skor *similarity* tertinggi. Informasi ini akan digunakan untuk menghasilkan rekomendasi konten kepada pengguna, terutama jika pengguna merupakan *cold-start user* yang belum memiliki cukup histori interaksi.

Proses *CBF* ini bersifat *item-to-item* dan tidak bergantung pada interaksi pengguna secara langsung. Oleh karena itu, model ini menjadi komponen penting dalam sistem *hybrid*, terutama untuk menangani kasus *cold-start* atau pengguna baru.

3.6.5 Hybrid Approach

Pendekatan *hybrid* yang digunakan dalam penelitian ini dirancang untuk menggabungkan keunggulan dari dua teknik sistem rekomendasi utama guna meningkatkan akurasi prediksi, relevansi rekomendasi, serta kemampuan sistem dalam menangani permasalahan *data sparsity* dan *cold-start user*. Sistem ini mengintegrasikan *Collaborative Filtering* berbasis algoritma *Singular Value Decomposition (SVD)* dan *Content-Based*

Filtering berbasis *TF-IDF Vectorizer* melalui pendekatan *late fusion*, di mana skor prediksi dari masing-masing model digabungkan menggunakan skema penggabungan berbobot (*weighted hybrid*).

Nilai skor dari model *CF* dan *CBF* terlebih dahulu dinormalisasi, lalu dikombinasikan menggunakan rumus linier berbobot dengan parameter α , yang ditentukan melalui proses *tuning* berdasarkan hasil evaluasi performa individual model terhadap metrik *RMSE* dan *Precision@10*. Semakin baik performa suatu model, semakin besar kontribusinya dalam proses penggabungan skor akhir.

Selain itu, pendekatan ini juga memungkinkan sistem untuk memberikan rekomendasi yang lebih adaptif, karena tetap dapat menghasilkan saran personal kepada pengguna aktif maupun pengguna baru yang belum memiliki histori interaksi. Dengan demikian, sistem *hybrid* yang dibangun dalam penelitian ini tidak hanya efektif dalam menangkap pola preferensi pengguna secara eksplisit maupun implisit, tetapi juga fleksibel untuk diterapkan pada skenario skala mikro seperti simulasi platform *streaming anime MJ CELL*.

3.7 Metode Evaluasi

Evaluasi dalam penelitian ini dilakukan untuk mengukur dan membandingkan performa dari tiga pendekatan sistem rekomendasi, yaitu *Collaborative Filtering (SVD)*, *Content-Based Filtering (TF-IDF)*, dan *Hybrid Filtering*. Terdapat dua pendekatan evaluasi yang digunakan, yaitu evaluasi prediktif berbasis *rating* dan evaluasi rekomendasi *Top-N*.

3.7.1 Evaluasi Akurasi Prediksi

Evaluasi ini dilakukan dengan mengukur selisih antara *rating* yang diprediksi oleh model dengan *rating* aktual pada data uji. Metrik yang digunakan adalah *RMSE* dan *MAE*. Kedua metrik ini dihitung menggunakan fungsi `accuracy.RMSE()` dan `accuracy.mae()` dari pustaka *Surprise*. Proses evaluasi dilakukan terhadap model *CF*, *CBF*, dan *Hybrid* menggunakan data *test.csv*, yang telah dibagi secara *holdout* per pengguna.

3.7.2 Evaluasi Rekomendasi Top-N

Evaluasi ini bertujuan untuk mengukur kualitas hasil rekomendasi yang diberikan kepada pengguna. Evaluasi dilakukan dengan membandingkan daftar rekomendasi teratas dari masing-masing model terhadap interaksi aktual pengguna di data uji. Dalam penelitian ini digunakan pendekatan *Top-N* dengan parameter $k = 10$, dan ambang batas *rating* relevan adalah 8 (*rating* > 8 dianggap relevan).

Metrik evaluasi yang digunakan adalah *Precision@10*, *Recall@10*, dan *F1@10*, yang dihitung berdasarkan seberapa banyak rekomendasi dalam *Top-10* yang benar-benar relevan. Seluruh metrik ini dihitung menggunakan fungsi evaluasi kustom yang telah diimplementasikan dalam kode *Python*, dengan *input* berupa hasil prediksi model terhadap pasangan *user_id* dan *anime_id* pada data uji. Hasil evaluasi digunakan sebagai dasar untuk menentukan bobot optimal pada model *Hybrid*, serta untuk membandingkan performa akhir antar model.

3.8 Ruang Lingkup Pengembangan

3.8.1 Spesifikasi Perangkat Keras

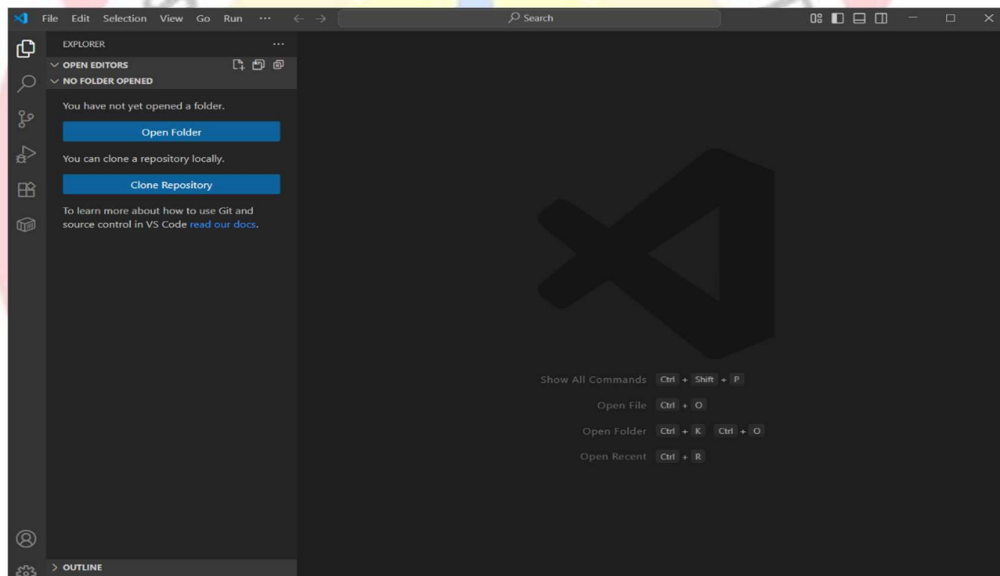
Perangkat keras merupakan komponen krusial dalam proses pelatihan model *machine learning*. Spesifikasi perangkat keras yang memadai dapat mempercepat waktu pelatihan serta memastikan proses komputasi berjalan lancar dan efisien. Dalam penelitian ini, proses pelatihan dan pengujian model sistem rekomendasi dilakukan pada perangkat dengan spesifikasi sebagaimana ditampilkan pada Tabel 3.15.

Tabel 3.15 Spesifikasi Perangkat Keras

Jenis Perangkat Keras	Nama Perangkat Keras
<i>Processor</i>	AMD Ryzen 5 4600H
<i>RAM</i>	16 GB DDR4 3200 MHz
Kartu Grafis	NVIDIA GeForce GTX 1650
Penyimpanan	SSD 512 GB
<i>Peripheral</i>	<i>Keyboard, Mouse</i>

3.8.2 Spesifikasi Perangkat Lunak

Perangkat lunak yang digunakan dalam penelitian ini terdiri dari sistem operasi *Windows 11*, bahasa pemrograman *Python 3.12*, serta sejumlah pustaka seperti *Pandas*, *NumPy*, *Scikit-learn*, dan *Surprise* untuk pemrosesan data dan pembentukan model sistem rekomendasi berbasis *Content-Based Filtering* dan *Collaborative Filtering*. Pengembangan antarmuka *backend* dilakukan menggunakan *framework Flask*, dengan basis data ringan *SQLite* untuk menyimpan data pengguna dan hasil rekomendasi. Seluruh proses pengembangan dilakukan dalam lingkungan *Visual Studio Code*, seperti ditampilkan pada Gambar 3.3.

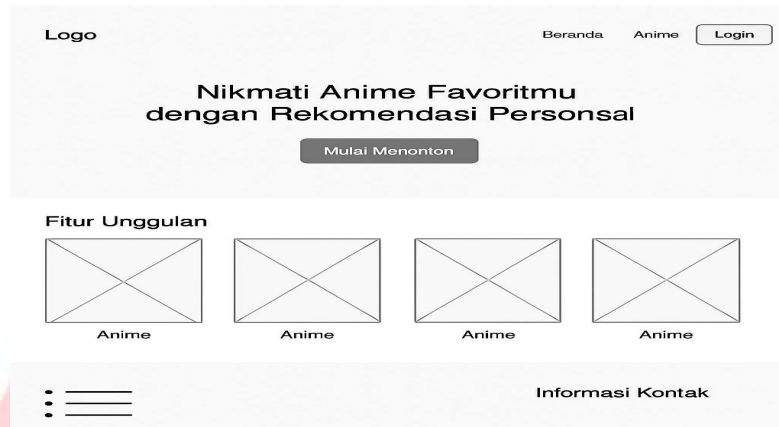


Gambar 3.3 *Visual Studio Code*

3.9 Perancangan Layar, Menu

3.9.1 Perancangan Layar, Menu

a. Wireframe Halaman Awal



Gambar 3.4 Wireframe Halaman Awal

Gambar 3.4 merupakan *wireframe* dari halaman awal sistem *MJ CELL Streaming Anime*. Desain awal ini menggambarkan struktur dasar antarmuka pengguna sebelum tahap implementasi visual. Terdapat elemen judul dan navigasi sederhana di bagian atas, seperti *Beranda*, *Anime*, dan tombol *Login*. Di bagian tengah, terdapat slogan utama "*Nikmati Anime Favoritmu dengan Rekomendasi Personal*" yang disertai tombol *Mulai Menonton* sebagai *call-to-action*. Bagian bawah halaman menampilkan empat kolom *Fitur Unggulan*, masing-masing mewakili kategori atau *anime* populer. *Footer* disusun dengan menu navigasi tambahan dan informasi kontak sebagai elemen penutup halaman.

b. Wireframe Halaman Login



Gambar 3.5 Wireframe Halaman Login

Gambar 3.5 merupakan *wireframe* untuk halaman *login* sistem *MJ CELL Streaming Anime*. Halaman ini dirancang dengan fokus pada kesederhanaan dan kemudahan akses bagi pengguna. Terdapat dua kolom *input* utama, yaitu *Input Email* dan *Input Password*, yang digunakan untuk mengautentikasi pengguna ke dalam sistem. Di bawah *form*, terdapat tombol *Login* untuk mengirimkan kredensial yang telah diisi. Selain itu, tersedia tautan *Daftar* di bagian bawah untuk mengarahkan pengguna baru yang belum memiliki akun ke halaman registrasi.

c. Wireframe Halaman Registrasi

Gambar 3.6 Wireframe Halaman Registrasi

Gambar 3.6 memperlihatkan desain *wireframe* dari halaman registrasi pada sistem *MJ CELL Streaming Anime*. Halaman ini disediakan bagi pengguna baru yang ingin membuat akun sebelum dapat mengakses sistem. Formulir registrasi terdiri dari dua kolom *input*, yaitu *Input Nama* dan *Input Email*, yang harus diisi oleh pengguna. Setelah data terisi, pengguna dapat menekan tombol *Daftar* untuk menyelesaikan proses pendaftaran.

d. Wireframe Halaman Histori Penilaian User

Logo & Judul: MJ CELL Streaming Anime			Search Bar	Menu
Riwayat Penilaian Anda				
ID Anime: XXXX	Rating: X	Tanggal: dd/mm/yyyy		
ID Anime: XXXX	Rating: X	Tanggal: dd/mm/yyyy		
ID Anime: XXXX	Rating: X	Tanggal: dd/mm/yyyy		
ID Anime: XXXX	Rating: X	Tanggal: dd/mm/yyyy		
ID Anime: XXXX	Rating: X	Tanggal: dd/mm/yyyy		
ID Anime: XXXX	Rating: X	Tanggal: dd/mm/yyyy		
ID Anime: XXXX	Rating: X	Tanggal: dd/mm/yyyy		

Gambar 3.7 Wireframe Halaman Histori Penilaian User

Gambar 3.7 menampilkan desain *wireframe* untuk halaman histori penilaian *user* pada sistem *MJ CELL Streaming Anime*. Halaman ini berfungsi untuk menampilkan daftar penilaian yang telah diberikan pengguna terhadap *anime* yang tersedia di platform. Setiap entri riwayat disajikan dalam bentuk kartu yang memuat informasi *ID anime*, nilai *rating* yang diberikan, serta tanggal penilaian. Desain ini bertujuan agar pengguna dapat dengan mudah meninjau kembali histori penilaiannya, sekaligus menjadi referensi bagi sistem dalam memperbarui rekomendasi secara berkelanjutan.

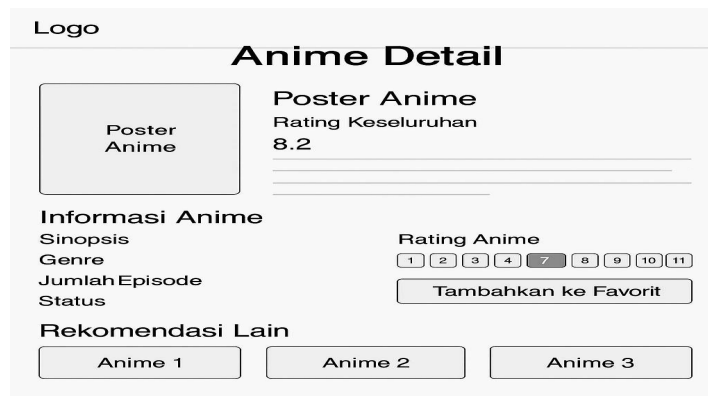
e. Wireframe Halaman Utama



Gambar 3.8 Wireframe Halaman Utama

Gambar 3.8 memperlihatkan rancangan *wireframe* dari halaman utama dalam sistem *MJ CELL Streaming Anime*. Halaman ini merupakan bagian utama yang menampilkan hasil personalisasi berdasarkan histori dan preferensi pengguna. Desain Halaman utama ini menampilkan daftar *anime* yang direkomendasikan dalam bentuk kartu, yang memuat informasi seperti judul, *genre*, skor *rating*, dan gambar sampul. Selain itu, terdapat navigasi kategori di bagian atas untuk mempermudah pengguna menelusuri rekomendasi berdasarkan jenis atau tema *anime*.

f. Wireframe Halaman Detail Anime



Gambar 3.9 Wireframe Halaman Detail Anime

Gambar 3.9 menunjukkan rancangan *wireframe* halaman detail *anime* pada sistem *MJ CELL Streaming Anime*. Halaman ini berfungsi untuk menampilkan informasi lengkap mengenai *anime* yang dipilih oleh pengguna. Informasi yang ditampilkan mencakup judul, gambar sampul, *genre*, skor *rating*, sinopsis, serta tombol untuk memberikan *rating*. Terdapat juga bagian rekomendasi terkait (*related anime*) yang menampilkan daftar *anime* serupa berdasarkan hasil perhitungan kemiripan konten dari model *CBF*.

g. Wireframe Halaman Pengguna



Gambar 3.10 Wireframe Halaman Pengguna

Gambar 3.10 menampilkan *wireframe* halaman pengguna, yang berfungsi untuk menampilkan informasi akun pengguna secara personal. Pada bagian atas halaman terdapat nama pengguna, alamat *email*, dan tombol Edit Profil.

h. Wireframe Halaman Utama Admin



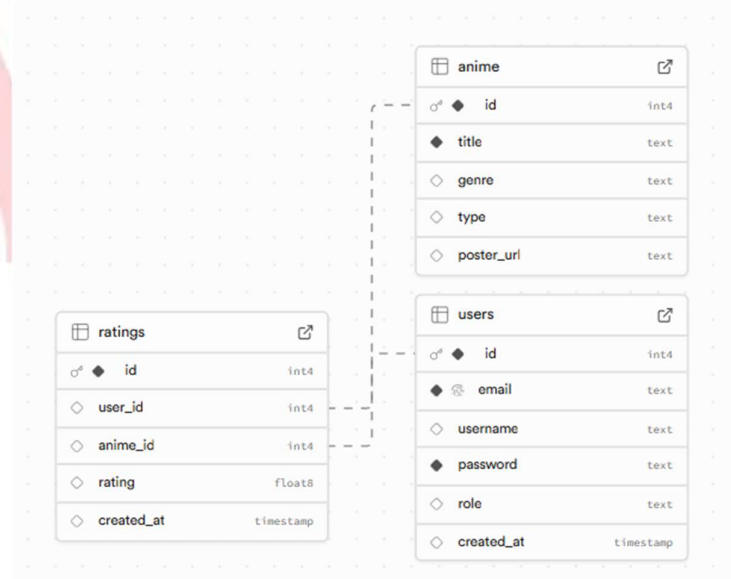
Gambar 3.11 Wireframe Halaman Utama Admin

Gambar 3.11 menunjukkan desain *wireframe* halaman Utama *Admin* yang hanya dapat diakses oleh pengguna dengan peran *admin*. Halaman Utama ini berfungsi sebagai pusat pengelolaan sistem.

3.9.2 Perancangan Database

Database dirancang untuk *database* yang dibuat serta digunakan dalam membantu aplikasi dalam menyimpan datanya.

a. Desain Entity-Relation Diagram (ERD)



Gambar 3.12 Rancangan ERD Database

b. Spesifikasi Tabel

Tabel 3.16 Tabel *User*

Nama Atribut	Tipe Data	Keterangan
<i>id</i>	<i>BIGINT (PK)</i>	ID unik pengguna (<i>auto increment</i>)
<i>email</i>	<i>TEXT (UNIQUE)</i>	Email unik pengguna
<i>username</i>	<i>TEXT</i>	Username pengguna
<i>password</i>	<i>TEXT</i>	Password terenkripsi
<i>created_at</i>	<i>TIMESTAMP</i>	Tanggal pendaftaran
<i>is_admin</i>	<i>BOOLEAN</i>	Status admin (<i>true/false</i>)

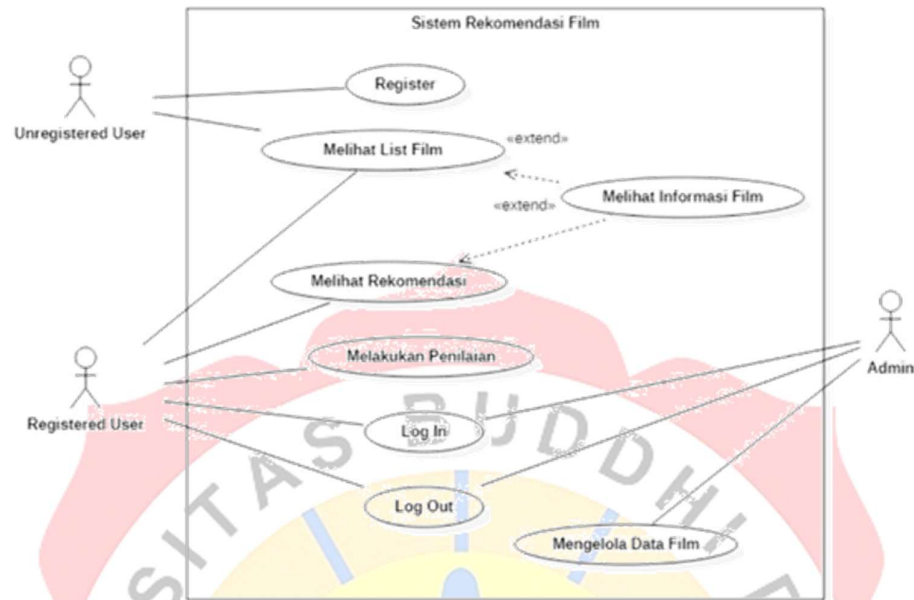
Tabel 3.17 Tabel *Anime*

Nama Atribut	Tipe Data	Keterangan
<i>id</i>	<i>BIGINT (PK)</i>	ID unik anime (<i>auto increment</i>)
<i>title</i>	<i>TEXT</i>	Judul anime
<i>genre</i>	<i>TEXT</i>	Genre anime
<i>type</i>	<i>TEXT</i>	Jenis anime (<i>TV, OVA, Movie, dll</i>)
<i>poster_url</i>	<i>TEXT</i>	URL gambar poster
<i>ratings</i>	<i>FLOAT</i>	Rating rata-rata anime
<i>members</i>	<i>INTEGER</i>	Jumlah anggota/pengikut

Tabel 3.18 Tabel *Rating*

Nama Atribut	Tipe Data	Keterangan
<i>id</i>	<i>BIGINT (PK)</i>	ID unik rating (<i>auto increment</i>)
<i>user_id</i>	<i>BIGINT (FK)</i>	Relasi ke tabel <i>Users</i>
<i>anime_id</i>	<i>BIGINT (FK)</i>	Relasi ke tabel <i>Anime</i>
<i>rating</i>	<i>FLOAT</i>	Nilai rating (skala 0–10)
<i>created_at</i>	<i>TIMESTAMP</i>	Waktu pemberian rating

3.9.3 Use Case Diagram



Gambar 3.13 Use Case Diagram

Pada Gambar 3.13 *Use Case Diagram* pada sistem rekomendasi film berbasis *Hybrid Filtering* ini menggambarkan interaksi antara dua aktor utama, yaitu pengguna (*user*) dan *admin*, dengan berbagai fitur sistem yang tersedia pada platform *streaming*. Pengguna dapat melakukan registrasi atau *login*, melihat daftar film, memberikan *rating* pada film yang telah ditonton, menerima rekomendasi film yang dipersonalisasi, melihat histori interaksi, serta melakukan *logout* dari akun. Sementara itu, *admin* memiliki kewenangan untuk *login* ke sistem, mengelola data film dan data *user*, melihat laporan penggunaan sistem, serta melakukan *logout* dari halaman *dashboard admin*. Dengan demikian, *Use Case Diagram* ini merepresentasikan seluruh kebutuhan fungsional inti yang menunjang proses personalisasi dan pengelolaan sistem rekomendasi film secara efisien dan terstruktur.