BAB III
METODOLOGI PENELITIAN

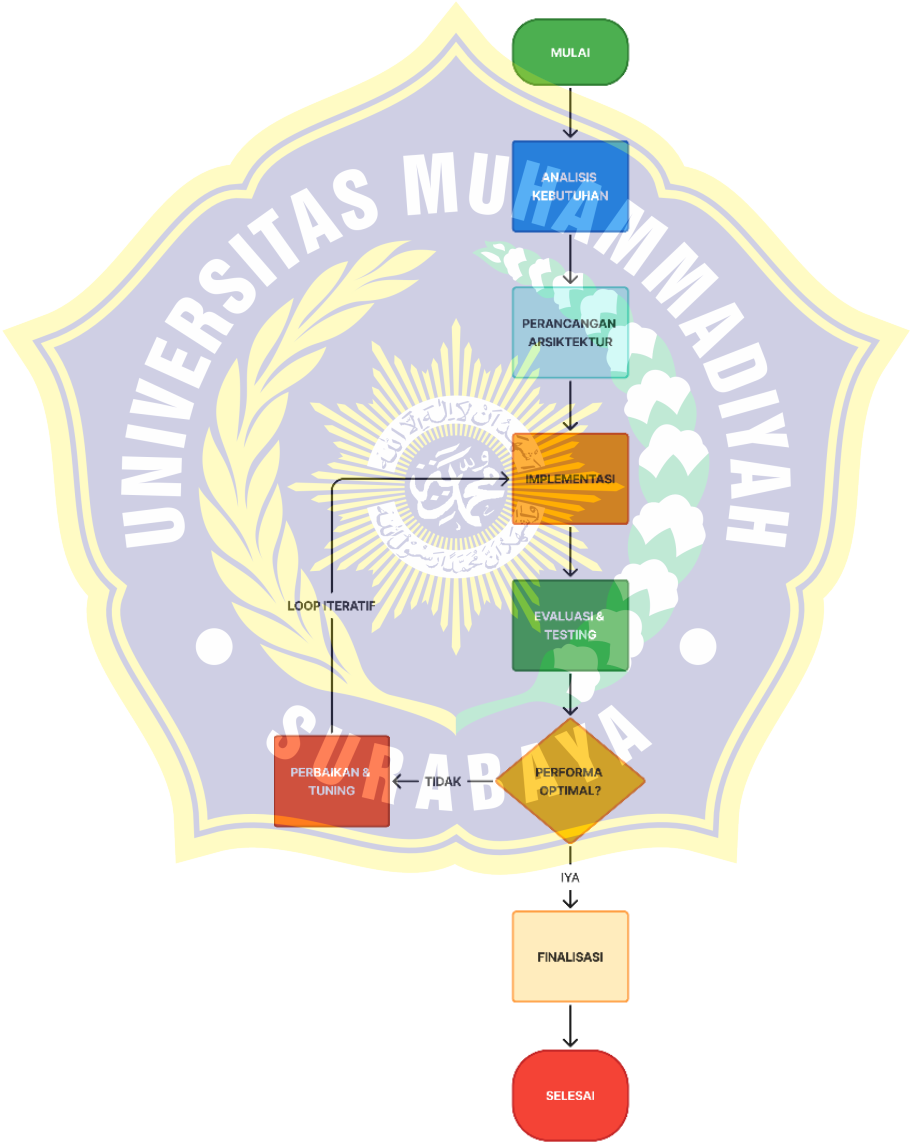
Metodologi penelitian ini dirancang mengikuti kerangka kerja iteratif berbasis Software Development Life Cycle (SDLC) seperti yang ditunjukkan pada Gambar 3.1. Kerangka kerja mencakup enam tahapan utama: analisis kebutuhan sistem, perancangan arsitektur hibrida, implementasi prototip, evaluasi dan *testing*, perbaikan dan *tuning*, serta finalisasi model. Proses iteratif memungkinkan penyempurnaan berkelanjutan hingga mencapai performa optimal yang diukur melalui metrik rekomendasi top-N.

Dalam setiap siklus iterasi, dilakukan pengumpulan dan pengolahan data multimodal yang bersumber dari *dataset* MovieLens dan metadata The Movie Database (TMDb). Ekstraksi fitur visual dari poster film menggunakan Vision Transformer (ViT) untuk menangkap informasi estetika dan komposisi visual. Ekstraksi fitur semantik dari sinopsis menggunakan Bidirectional *Encoder* Representations from Transformers (BERT) untuk memahami konten naratif. Metadata terstruktur meliputi interaksi pengguna-item, skor popularitas film dari TMDb, informasi pemeran (*cast*), dan kategori genre. Representasi dari berbagai modalitas ini kemudian diintegrasikan melalui modul fusi berbasis mekanisme *cross-attention* untuk membentuk *embedding* item yang kaya dan informatif.

Embedding multimodal yang dihasilkan dikombinasikan secara aditif dengan *embedding* kolaboratif item melalui parameter α yang dipelajari, kemudian diproses dalam arsitektur graf LightGCN sehingga sinyal konten dan sinyal kolaboratif saling memperkaya selama propagasi graf.

Model diimplementasikan menggunakan *framework* PyTorch dan dievaluasi menggunakan metrik rekomendasi top-N yang mencakup Precision@K, Recall@K, NDCG@K, dan Hit Ratio@K. Evaluasi juga mencakup analisis mendalam terhadap potensi *overfitting*, keseimbangan kontribusi antar modalitas melalui studi ablasi, serta stratifikasi performa berdasarkan kategori popularitas film untuk mengukur kemampuan model dalam menangani item populer maupun *long-tail*.

Proses iteratif berlangsung melalui siklus implementasi, evaluasi, dan perbaikan hingga diperoleh model akhir yang *robust* dan menunjukkan peningkatan kinerja signifikan dibandingkan pendekatan *baseline*. Metode *baseline* meliputi *collaborative filtering* klasik berbasis *similarity* (ItemKNN), LightGCN tanpa konten multimodal, dan model multimodal tanpa komponen graf. Pendekatan ini memastikan model yang dihasilkan tidak hanya unggul secara empiris, namun juga mudah direproduksi dan dapat diadaptasi untuk penelitian sistem rekomendasi film berbasis *dataset* publik.



Gambar 3. 1 Kerangka Kerja SDLC Iteratif

3.1 Pengumpulan Data

Sumber data yang digunakan dalam penelitian ini meliputi dua *dataset* publik yang telah divalidasi secara luas sebagai standar evaluasi dalam riset sistem rekomendasi film. *Dataset* utama berupa MovieLens dalam dua varian ukuran, yaitu MovieLens 100K dan MovieLens 1M, yang diperoleh melalui portal resmi GroupLens Research (GroupLens, n.d.). Data tersimpan dalam format Comma-Separated Values (CSV) yang mencakup empat atribut esensial: identifikasi pengguna (*userId*), identifikasi film (*movieId*), nilai rating pada skala 1 hingga 5, serta stempel waktu (*timestamp*) yang merekam momen pemberian rating. Struktur data interaksi pengguna-item ini memungkinkan konstruksi graf bipartit yang menjadi fondasi arsitektur LightGCN dalam memodelkan hubungan kolaboratif antar entitas (Harper & Konstan, 2016; LeeGeon et al., 2025).

Komponen data kedua merupakan metadata deskriptif film yang diperoleh melalui Application Programming Interface (API) The Movie Database (TMDb), sebuah repositori informasi film yang terkelola secara komunitas (*The Movie Database (TMDB)*, n.d.). Proses pemetaan dilakukan dengan menyelaraskan identifikator film MovieLens terhadap sistem identifikasi TMDb berdasarkan kesesuaian judul dan tahun produksi. Pemetaan berhasil dilakukan dengan tingkat keberhasilan minimum 85% dari total film dalam *dataset*.

Ekstraksi metadata dari TMDb mencakup beberapa komponen penting. Pertama, jalur citra poster beresolusi tinggi yang digunakan sebagai input representasi visual Vision Transformer (ViT). Kedua, ringkasan naratif (*overview*) berbahasa Inggris sebagai sumber fitur semantik melalui BERT. Ketiga, skor popularitas (*popularity*) dan jumlah voting (*vote_count*) sebagai indikator popularitas film untuk stratifikasi evaluasi. Keempat, informasi lima aktor utama (*cast*) melalui *endpoint credits*. Kelima, atribut tambahan berupa genre dan tahun rilis. Kedua sumber data bersifat terbuka untuk riset akademik dan telah menjadi rujukan dalam berbagai publikasi rekomendasi multimodal (Mu & Wu, 2023; Xia et al., 2024) sebagaimana disajikan pada Tabel 3.1.

Tabel 3. 1 Karakteristik *Dataset* MovieLens

Atribut	MovieLens 100K	MovieLens 1M
Jumlah Pengguna	943	6,040
Jumlah Film	1,682	3,706
Jumlah Rating	100,000	1,000,209
Sparsitas (%)	93.7	95.8
Skala Rating	1–5	1–5

Sumber:(Harper & Konstan, 2016)

Sebelum digunakan dalam tahap pemodelan, data interaksi pengguna-item menjalani preprocessing awal untuk mengurangi *noise* dari entitas yang jarang muncul. Filtering dilakukan dengan menghapus pengguna yang memiliki jumlah rating kurang dari 5 dan film yang memiliki rating kurang dari 10. Threshold ini dipilih untuk memastikan setiap entitas memiliki cukup informasi untuk pembelajaran model, sekaligus menjaga ukuran *dataset* tetap representatif. Metadata TMDb yang berhasil diekstraksi kemudian diintegrasikan dengan data MovieLens melalui *movieId* sebagai kunci penggabungan, menghasilkan *dataset* multimodal lengkap yang siap digunakan untuk eksperimen.

3.2 Pengolahan Data

Tahapan awal pengolahan data mencakup prosedur pembersihan dan integrasi kedua sumber *dataset* untuk menghasilkan korpus multimodal yang terstandarisasi. Film yang gagal dalam proses pemetaan identifikasi MovieLens ke TMDb, tidak memiliki poster yang valid, atau tidak memiliki sinopsis berbahasa Inggris dihapus dari *dataset* eksperimen. Data interaksi pengguna-item menjalani filtering dengan menghapus pengguna yang memiliki kurang dari 5 rating dan film yang memiliki kurang dari 10 rating. Threshold ini dipilih untuk mengurangi noise dari entitas yang terlalu jarang muncul, sekaligus memastikan setiap entitas memiliki informasi yang cukup untuk pembelajaran model (Rodríguez-Baena et al., 2025). Setelah filtering, *dataset* diintegrasikan dengan metadata TMDb

menggunakan movieId sebagai kunci penggabungan, menghasilkan korpus yang menggabungkan data interaksi dengan informasi multimodal untuk setiap film.

Ekstraksi representasi visual dilakukan melalui tahapan preprocessing di mana citra poster film diseragamkan ke resolusi 224×224 piksel dan dinormalisasi menggunakan statistik mean dan standar deviasi dari *dataset* ImageNet. Normalisasi citra poster dilakukan dengan transformasi yang dinyatakan dalam persamaan (3.1):

$$x_{norm} = \frac{x - \mu}{\sigma} \tag{3.1}$$

di mana $\boldsymbol{\mu} = [0.485, 0.456, 0.406]$ dan $\boldsymbol{\sigma} = [0.229, 0.224, 0.225]$ untuk kanal RGB. Prosedur ini mema/stikan kompatibilitas dengan model Vision Transformer yang telah di-pretrain pada ImageNet. Poster yang telah dinormalisasi kemudian diproses melalui Vision Transformer (ViT) dengan arsitektur ViT-Base yang menghasilkan *embedding* visual berdimensi 768. Proses ekstraksi menggunakan model *pre-trained* pada ImageNet yang di-*fine-tune* pada dua *layer encoder* terakhir melalui tugas klasifikasi genre untuk menghasilkan representasi visual yang lebih spesifik terhadap karakteristik poster film (Palanisamy et al., 2025).

Representasi tekstual dari sinopsis film diperoleh melalui preprocessing menggunakan tokenizer BERT (bert-base-uncased). Setiap sinopsis di-tokenize menjadi subword tokens dengan panjang maksimum 128 token, mencakup token khusus [CLS] di awal dan [SEP] di akhir sesuai format input BERT. Sinopsis yang melebihi panjang maksimum dipotong, sementara sinopsis yang lebih pendek di-pad dengan token [PAD]. Attention mask dibuat untuk membedakan token aktual dari padding, memungkinkan model fokus hanya pada konten yang valid. *Embedding* tekstual dengan dimensi 768 diekstraksi dari output hidden state token [CLS] pada *layer* terakhir BERT, yang merepresentasikan makna global dari keseluruhan sinopsis (Devlin et al., n.d.).

Metadata terstruktur berupa informasi cast direpresentasikan melalui *embedding layer* yang dapat dipelajari, di mana setiap aktor dalam vocabulary

memiliki vektor *embedding* unik. Untuk setiap film, lima aktor utama diambil dan *embedding* mereka diagregasi menggunakan operasi mean pooling untuk membentuk representasi cast dengan dimensi 64. Genre film di-encode menggunakan multi-hot encoding, menghasilkan vektor biner dengan dimensi sejumlah total kategori genre dalam *dataset*. Skor popularitas dari TMDb dinormalisasi menggunakan min-max scaling ke rentang [0, 1] dengan persamaan (3.2):

$$\text{popularity}_{\text{norm}} = \frac{\text{popularity} - \text{popularity}_{\text{min}}}{\text{popularity}_{\text{max}} - \text{popularity}_{\text{min}}} \tag{3.2}$$

Normalisasi ini memastikan semua fitur numerik berada dalam skala yang sebanding, mencegah dominasi fitur dengan magnitude besar dalam proses pembelajaran.

Strategi partisi data mengadopsi pendekatan *leave-last-out* per pengguna untuk mensimulasikan skenario rekomendasi temporal. Untuk setiap pengguna, interaksi terakhir berdasarkan timestamp dipisahkan sebagai set testing, interaksi kedua terakhir sebagai set validasi, dan sisanya sebagai set training. Pendekatan ini memastikan model dilatih pada interaksi historis dan dievaluasi pada interaksi masa depan, mencerminkan kondisi *deployment* sistem rekomendasi yang sesungguhnya. *rating* bernilai 3 atau lebih dianggap sebagai interaksi positif mengikuti pendekatan binarisasi *implicit feedback* yang diterapkan pada *dataset* MovieLens oleh (Anelli et al., 2021), di mana nilai 3 merupakan titik tengah skala yang membedakan preferensi positif dari netral atau negatif, sementara pasangan pengguna-item tanpa rating dianggap sebagai interaksi negatif (unobserved). Konversi ini memungkinkan penggunaan Bayesian Personalized Ranking (BPR) sebagai fungsi objektif, yang dirancang khusus untuk pembelajaran dari *implicit feedback* dengan paradigma pairwise ranking (Rendle et al., 2009).

Untuk setiap batch training, triplet (u, i, j) dikonstruksi di mana u adalah pengguna, i adalah item positif yang berinteraksi dengan u, dan j adalah item negatif yang dipilih secara *random* dari item yang tidak berinteraksi dengan u. Strategi *negative sampling* menggunakan pendekatan *popularity-aware* yang

memprioritasikan item populer sebagai sampel negatif, sehingga model dilatih terhadap pasangan yang lebih menantang dan menghasilkan konvergensi yang lebih cepat. Seluruh proses preprocessing dan feature extraction dilakukan secara offline sebelum training, dengan hasil *embedding* disimpan dalam format *cache* untuk mempercepat proses *training* iteratif. Pipeline preprocessing dirancang modular dan dapat direplikasi, dengan dokumentasi lengkap untuk memastikan reproducibility penelitian.

3.2.1 Parameter dan Algoritma Ekstraksi Fitur

Sistem rekomendasi yang diusulkan mengintegrasikan berbagai sumber informasi melalui pemrosesan multimodal dan pemodelan graf kolaboratif. Setiap modalitas memiliki karakteristik unik yang memerlukan teknik ekstraksi fitur spesifik. Tabel 3.2 menyajikan ringkasan komprehensif parameter input, algoritma ekstraksi, proses pemrosesan, dan representasi output yang dihasilkan.

Pemilihan algoritma untuk setiap modalitas didasarkan pada efektivitas yang telah terbukti dalam literatur sistem rekomendasi. Vision Transformer (ViT) dipilih untuk memproses poster karena kemampuannya menangkap dependensi global pada representasi visual melalui mekanisme self-attention (Álvarez-Sánchez et al., 2025; Palanisamy et al., 2025). BERT digunakan untuk sinopsis karena keunggulannya dalam pemahaman konteks semantik bidirectional (Devlin et al., 2019). *Embedding layer* trainable digunakan untuk cast dan genre untuk fleksibilitas representasi dalam sistem rekomendasi (Mu & Wu, 2023). Yang krusial, LightGCN berperan sebagai pemroses histori interaksi dan pola kolaboratif pengguna melalui propagasi sinyal pada graf bipartit user-item (X. He et al., 2020; LeeGeon et al., 2025), bukan hanya sebagai backbone *collaborative filtering* pasif.

Tabel 3. 2 Parameter Multimodal dan Algoritma Ekstraksi Fitur

No	Parameter	Sumber Data	Algoritma Ekstraktor	Dimensi Output	Keterangan
1	Poster Film	TMDb API (gambar JPEG/PNG 224×224 px)	ViT-Base/16 (*fine-tuned* 2 *layer* terakhir)	768-dim → 256-dim (setelah proyeksi)	Token [CLS] diambil sebagai representasi visual global poster
2	Sinopsis Film	TMDb API (teks bahasa Inggris, maks. 128 token)	BERT (bert-base-uncased, *fine-tuned* 2 *layer* terakhir)	768-dim → 256-dim (setelah proyeksi)	Token [CLS] diambil sebagai representasi semantik sinopsis
3	Informasi Cast	TMDb API (top-5 aktor per film)	Trainable Cast *Embedding* Layer	64-dim (rata-rata 5 *embedding* aktor)	*Embedding* aktor diinisialisasi acak dan dilatih bersama model
4	Genre Film	MovieLens + TMDb (label multi-genre biner)	One-Hot Encoding + Linear Projection Layer	32-dim (setelah proyeksi linear)	Genre dikodekan sebagai vektor biner lalu diproyeksikan ke ruang laten

No	Parameter	Sumber Data	Algoritma Ekstraktor	Dimensi Output	Keterangan
5	Histori Interaksi Pengguna	MovieLens *Dataset* (matriks rating user-item)	LightGCN Graph Propagation	256-dim (*embedding* akhir user dan item)	Data interaksi membentuk user-item bipartite graph; digunakan sebagai sinyal kolaboratif untuk memperbarui embedding

Integrasi seluruh modalitas dilakukan melalui modul Fusion Transformer dengan mekanisme *cross-attention* multi-head (4 heads, hidden dimension 64). Fusion Transformer mempelajari korespondensi antar modalitas secara adaptif, memungkinkan setiap sumber informasi untuk saling memperkaya (Kim et al., 2023; Xia et al., 2024). *Output* fusi berupa *embedding* multimodal berdimensi 256 (m_i) sesuai persamaan (3.3) yang kemudian dikombinasikan secara aditif dengan *embedding* kolaboratif item sebelum memasuki propagasi graf LightGCN. Dengan mekanisme ini, *embedding* item pada graf tidak hanya bergantung pada pola interaksi historis, tetapi juga diperkaya oleh informasi konten dari berbagai modalitas tanpa menghilangkan sinyal kolaboratif yang sudah ada.

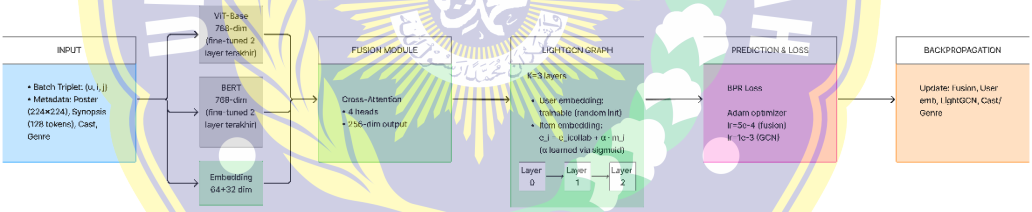
Arsitektur hibrida ini memungkinkan sistem memanfaatkan kelebihan dari dua paradigma*: content-based filtering* melalui representasi multimodal yang kaya, dan *collaborative filtering* melalui pemodelan graf yang menangkap preferensi kolektif (Liu et al., 2025). Pendekatan ini efektif mengatasi *cold-start problem* pada item baru yang memiliki konten visual dan tekstual namun belum memiliki histori interaksi, sekaligus memanfaatkan sinyal kolaboratif untuk item dengan interaksi yang cukup (Rodríguez-Baena et al., 2025; H. Zhou et al., 2023).

3.3 Rancangan atau Design yang diusulkan

3.3.1 Arsitektur Model Rekomendasi

Arsitektur yang dirancang dalam penelitian ini mengadopsi pendekatan hibrida yang mengintegrasikan pemrosesan konten multimodal dengan pemodelan relasi kolaboratif secara *end-to-end*. Mekanisme Transformer digunakan untuk memproses representasi konten heterogen dari berbagai modalitas, sementara struktur graf LightGCN memodelkan pola preferensi kolektif antar pengguna dan item. Integrasi kedua komponen ini bertujuan menghasilkan sistem rekomendasi film dengan akurasi superior dan ketahanan terhadap skenario *cold-start*.

Visualisasi lengkap arsitektur *training flow* disajikan pada Gambar 3.2, yang mengilustrasikan alur pemrosesan data melalui enam fase utama: input data, ekstraksi fitur multimodal, fusi representasi, propagasi graf kolaboratif, komputasi prediksi dan *loss*, serta *backpropagation* untuk pembaruan parameter. Arsitektur ini memungkinkan pembelajaran *end-to-end* di mana informasi konten multimodal dan sinyal kolaboratif saling memperkaya dalam satu kerangka terintegrasi. Visualisasi lengkap arsitektur *training flow* disajikan pada Gambar 3.2.



Gambar 3. 2 Arsitektur *Training Flow* Model Hibrida Transformer-LightGCN Yang Di Usulkan

Setiap komponen dalam arsitektur memiliki peran spesifik dalam transformasi data dari input mentah hingga menghasilkan model yang terlatih. Berikut adalah penjelasan detail untuk setiap komponen:

a. Lapisan Masukan

Fase input menerima data *training* dalam format *batch triplet* (u, i, j) di mana u merepresentasikan *user ID*, i merupakan item positif (film yang telah diberi

rating oleh pengguna), dan j adalah item negatif yang dipilih secara *random* dari film yang tidak pernah diberi rating oleh pengguna tersebut. Paradigma triplet ini memungkinkan model belajar melalui *pairwise ranking*, yaitu memberikan skor lebih tinggi pada item relevan dibandingkan item yang tidak relevan (Rendle et al., 2009).

Untuk setiap item dalam triplet (baik i maupun j), sistem mengambil metadata lengkap yang mencakup empat modalitas: poster film berukuran 224×224 pixel dalam format RGB, sinopsis naratif dengan maksimal 128 token, informasi lima aktor utama, dan vektor *multi-hot* yang merepresentasikan genre film. Kelengkapan metadata multimodal ini menjadi fondasi bagi model untuk membentuk representasi item yang kaya informasi dari berbagai perspektif.

b. Modul Ekstraksi Fitur

Modul ekstraksi fitur terdiri dari tiga sub-komponen yang memproses modalitas berbeda secara paralel, menghasilkan representasi vektor berkualitas tinggi untuk setiap jenis informasi.

Vision Transformer (ViT-Base) memproses poster film dengan membagi gambar menjadi 196 *patches* berukuran 16×16 pixel, kemudian memodelkan hubungan global antar *patches* melalui 12 *layer* Transformer *encoder* dengan 12 *attention heads*. Model ViT yang digunakan merupakan arsitektur *pre-trained* pada *dataset* ImageNet yang di-*fine-tune* pada dua *layer encoder* terakhir melalui tugas klasifikasi genre untuk menghasilkan representasi visual yang lebih spesifik terhadap karakteristik poster film yang telah dipelajari dari jutaan gambar. Output dari ViT adalah *embedding* visual v_i berdimensi 768 yang menangkap informasi estetika, komposisi warna, dan karakteristik visual poster film (Palanisamy et al., 2025; Saha & Xu, 2025).

BERT (bert-base-uncased) bertanggung jawab untuk memproses sinopsis film melalui mekanisme *bidirectional encoding* yang memungkinkan pemahaman konteks dari kedua arah secara simultan. Teks sinopsis di-*tokenize* menjadi *subword units* dengan token khusus [CLS] di awal dan [SEP] di akhir, kemudian diproses melalui 12 *layer* Transformer. *Hidden state* dari token [CLS] pada *layer* terakhir diambil sebagai representasi semantik global dari keseluruhan sinopsis,

menghasilkan *embedding* tekstual t_i berdimensi 768 yang menangkap plot dan tema naratif film. Model BERT di-*fine-tune* pada dua *layer encoder* terakhir melalui tugas klasifikasi genre untuk menghasilkan representasi semantik yang lebih spesifik terhadap domain perfilman, sementara *layer* sebelumnya tetap *frozen* guna mempertahankan pengetahuan bahasa yang telah dipelajari dari korpus berskala besar (Devlin et al., 2019). Pendekatan Transformer untuk ekstraksi fitur dari metadata telah menunjukkan keunggulan dalam sistem rekomendasi berbasis *collaborative filtering* (Khan et al., 2025).

Embedding Layers untuk cast dan genre merupakan komponen yang dapat dipelajari (*trainable*) selama proses *training*. Setiap aktor dalam *vocabulary* memiliki *embedding vector* unik yang di-*lookup* dari tabel *embedding*. Untuk setiap film, *embedding* dari lima aktor utama diagregasi menggunakan *mean pooling* untuk membentuk representasi cast c_i berdimensi 64. Genre film di-*encode* sebagai *multi-hot vector* dan diproyeksikan melalui *linear layer* menghasilkan *embedding* genre g_i berdimensi 32. Kedua *embedding layer* ini dipelajari dari data spesifik domain rekomendasi film untuk menangkap pola preferensi terkait aktor dan genre (Mu & Wu, 2023). Ekstraksi fitur dilakukan dalam mode *inference* dengan parameter model *pretrained* yang di-*freeze* untuk efisiensi komputasi pada komponen ViT dan BERT (Koren et al., 2009).

c. Modul Fusi

Modul fusi mengintegrasikan keempat representasi fitur $[v_i, t_i, c_i, g_i]$ melalui mekanisme *Cross-attention Transformer* dengan 4 *attention heads* dan *hidden dimension* 64. Mekanisme *cross-attention* memungkinkan setiap modalitas untuk "berkomunikasi" dengan modalitas lain, mempelajari korespondensi antar sumber informasi secara adaptif (Xia et al., 2024). Sebagai contoh, modalitas visual dapat "bertanya" kepada modalitas tekstual tentang relevansi antara tampilan poster dengan isi cerita film, sementara modalitas cast dapat berkontribusi pada pemahaman karakteristik aktor yang sesuai dengan tema naratif.

Representasi item multimodal m_i yang dihasilkan diformulasikan dalam persamaan (3.3):

$$m_i = \text{FusionTransformer}(v_i, t_i, c_i, g_i) \quad (3.3)$$

di mana m_i merupakan *embedding* multimodal berdimensi 256 yang merupakan representasi terintegrasi dari seluruh informasi film. Dimensi output yang lebih kecil dari total dimensi input (768+768+64+32 = 1,632) mengindikasikan adanya kompresi informasi yang efisien, di mana model belajar mengekstraksi fitur paling relevan dari setiap modalitas dan menggabungkannya dalam ruang representasi yang lebih *compact*. Seluruh parameter *Fusion Transformer* bersifat *trainable*, memungkinkan adaptasi terhadap karakteristik spesifik data rekomendasi film. Penelitian terkini tentang integrasi bertahap antara representasi multimodal dengan graf kolaboratif menunjukkan bahwa pendekatan *dual-stage fusion* dapat meningkatkan performa rekomendasi secara signifikan (Xu et al., 2025). *Embedding* multimodal ini kemudian dikombinasikan secara aditif dengan *embedding* kolaboratif item sebelum memasuki propagasi graf LightGCN.

d. Pemfilteran Kolaboratif Graf LightGCN

Komponen LightGCN memodelkan struktur kolaboratif dari interaksi pengguna-item melalui graf bipartit dengan K=3 *layer* propagasi. *Embedding* pengguna diinisialisasi secara *random* sebagai parameter yang dapat dipelajari, di mana preferensi pengguna terbentuk melalui agregasi *embedding* item yang pernah berinteraksi dengannya. *Embedding* item diperoleh melalui mekanisme aditif yang menggabungkan *embedding* kolaboratif dengan representasi multimodal menggunakan formula $e_i = e_i^{collab} + \alpha \cdot m_i$, di mana e_i^{collab} adalah *embedding* kolaboratif yang dipelajari dari data interaksi, m_i adalah representasi multimodal dari modul fusi, dan α diimplementasikan melalui fungsi *sigmoid* pada parameter *trainable* sehingga nilainya berada dalam rentang [0, 1]. Mekanisme ini memastikan sinyal kolaboratif tetap dipertahankan dan diperkaya oleh informasi multimodal, bukan digantikan sepenuhnya. Poin krusial dari komponen ini:

- *User embedding*: *trainable* (*random init*)
- *Item embedding*: *init from fusion* (m_i)

Perbedaan ini memungkinkan item *cold-start* (film baru tanpa riwayat interaksi) tetap memiliki representasi yang informatif dari konten multimodalnya,

sementara *user embedding* sepenuhnya dipelajari dari pola interaksi historis. Integrasi LightGCN dengan mekanisme *attention* untuk memperkaya sinyal kolaboratif telah menunjukkan hasil yang menjanjikan dalam penelitian terkini (Hassanzadeh et al., 2025).

Proses propagasi dilakukan melalui tiga *layer* (*Layer* 0, *Layer* 1, *Layer* 2) dengan agregasi linear *neighborhood* dan normalisasi simetris sebagaimana persamaan (2.2) dan persamaan (2.3) pada Bab II. *Embedding* akhir diperoleh melalui agregasi *weighted sum* dari semua *layer* propagasi sebagaimana persamaan (2.4) pada Bab II, dengan αk = 1/(K+1) untuk agregasi *uniform* yang menangkap informasi dari berbagai tingkat kedalaman hubungan kolaboratif(Lee et al., 2024; LeeGeon et al., 2025). Pendekatan ini memungkinkan sistem memanfaatkan pola preferensi tidak hanya dari pengguna dengan interaksi langsung yang sama, tetapi juga dari pengguna serupa hingga beberapa derajat hubungan.

e. Perhitungan Prediksi dan Fungsi Loss

Fase prediksi menghitung skor relevansi melalui operasi *dot product* antara *embedding* pengguna dan *embedding* item hasil agregasi. Skor prediksi preferensi pengguna u terhadap item i dihitung melalui persamaan (3.4):

$$\hat{y}_{ui} = \mathrm{e}_u^T \mathrm{e}_i \tag{3.4}$$

Skor ini merepresentasikan tingkat kesesuaian antara preferensi pengguna u dengan karakteristik item i dalam ruang *embedding* bersama berdimensi 256.

Model dioptimasi menggunakan fungsi objektif *Bayesian Personalized Ranking* (BPR) sebagaimana persamaan (2.5) pada Bab II, di mana skor item positif dioptimasi agar lebih tinggi dari item negatif melalui paradigma *pairwise ranking* dengan regularisasi L2 untuk mencegah *overfitting*. Model dioptimasi menggunakan algoritma Adam dengan dua *learning rate* terpisah: 5×10^{-4} untuk modul fusi yang memproses keluaran model *pre-trained* dan 1×10^{-3} untuk komponen LightGCN yang diinisialisasi secara acak. *Batch size* ditetapkan sebesar

2.048 untuk MovieLens 1M dan 512 untuk MovieLens 100K, dengan koefisien regularisasi L2 $\lambda = 1\times10^{-5}$ (Rendle et al., 2009).

f. Propagasi Balik

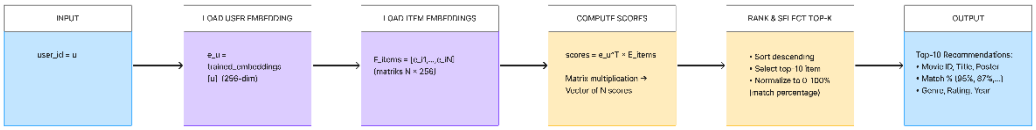
Fase *backpropagation* mendistribusikan gradien *loss* ke seluruh parameter yang *trainable* untuk memperbarui bobot model melalui algoritma optimasi Adam (Rendle et al., 2009). Parameter yang di-update mencakup seluruh bobot *Fusion Transformer* yang mempelajari cara optimal menggabungkan modalitas, *embedding* pengguna yang membentuk profil preferensi individual, bobot agregasi LightGCN yang menentukan pentingnya setiap *layer* propagasi, serta *embedding layers* untuk cast dan genre yang beradaptasi dengan pola preferensi domain-spesifik.

Komponen Vision Transformer dan BERT di-*fine-tune* pada dua *layer encoder* terakhir melalui tugas klasifikasi genre film agar representasi yang dihasilkan lebih spesifik terhadap domain perfilman, dibandingkan representasi umum dari *pre-training* pada ImageNet dan Wikipedia. *Layer* selain dua terakhir tetap dalam kondisi *frozen* untuk mempertahankan pengetahuan dasar yang sudah dipelajari. Strategi *partial fine-tuning* ini menyeimbangkan antara adaptasi terhadap domain target dan pencegahan *overfitting* akibat keterbatasan data pelatihan.

Proses *training* berlangsung secara *end-to-end* di mana seluruh komponen *trainable* dioptimasi secara bersamaan, memungkinkan informasi konten multimodal dan sinyal kolaboratif saling memperkaya dalam satu kerangka terintegrasi. Pendekatan ini berbeda dari metode *two-stage* yang memisahkan *feature extraction* dan *collaborative filtering*, sehingga menghasilkan representasi yang lebih koheren dan performa rekomendasi yang lebih superior (Pan & Yang, 2010).

Setelah model selesai dilatih melalui proses yang dijelaskan di atas, sistem memasuki fase *inference* untuk memberikan rekomendasi kepada pengguna. Berbeda dengan fase *training* yang melibatkan komputasi intensif dan update parameter, fase *inference* dirancang untuk efisiensi dan kecepatan respons real-time. Gambar 3.3 menunjukkan arsitektur *inference* flow yang menyederhanakan

proses menjadi lima langkah sekuensial untuk menghasilkan rekomendasi top-K dengan latency minimal. Alur *inference flow* untuk menghasilkan rekomendasi disajikan pada Gambar 3.3.



Gambar 3. 3 Arsitektur *Inference Flow* Model Hibrida Transformer-LightGCN Yang Di Usulkan

Fase *inference* memanfaatkan model yang telah terlatih untuk menghasilkan rekomendasi personal secara efisien tanpa memerlukan rekomputasi fitur multimodal atau propagasi graf. Seluruh *embedding* pengguna dan item yang telah dipelajari selama *training* di-*cache* dalam memori untuk akses instant, memungkinkan sistem memberikan respons dalam waktu kurang dari dua detik bahkan untuk katalog berisi puluhan ribu film. Berikut adalah penjelasan detail untuk setiap langkah dalam proses inference:

a. Masukan Permintaan Pengguna

Proses *inference* dimulai ketika sistem menerima permintaan rekomendasi dari pengguna, yang direpresentasikan melalui user_id = u. Input ini merupakan identifier unik pengguna yang akan digunakan untuk mengambil *embedding* preferensi yang telah dipelajari selama fase training. Berbeda dengan fase *training* yang memerlukan triplet data (u, i, j) beserta metadata lengkap dari setiap item (Rendle et al., 2009), fase *inference* hanya membutuhkan identitas pengguna sebagai input tunggal. Kesederhanaan input ini menjadikan proses sangat efisien dari sisi kompleksitas data dan memungkinkan sistem melayani ribuan request secara konkuren tanpa bottleneck pada *layer* input processing.

b. Pemuatan *Embedding* Pengguna

Sistem mengambil *embedding* pengguna yang telah terlatih dari memori atau *cache* menggunakan operasi lookup sederhana: e_u =

trained_user_embeddings[u]. *Embedding* ini merupakan vektor berdimensi 256 yang merepresentasikan preferensi agregat pengguna yang telah dipelajari melalui propagasi graf LightGCN selama *training* (X. He et al., 2020; LeeGeon et al., 2025). Vektor ini mengandung informasi preferensi laten yang terbentuk dari pola interaksi historis pengguna dengan item-item yang pernah diberi rating, serta pengaruh dari pengguna lain dengan pola preferensi serupa melalui mekanisme *collaborative filtering* multi-hop. Operasi load ini sangat cepat karena hanya melibatkan pengambilan vektor dari struktur data terindeks, tanpa memerlukan komputasi neural network atau propagasi graf. *Embedding* pengguna bersifat statis selama sesi *inference* dan hanya akan berubah ketika model dilatih ulang dengan data interaksi baru atau ketika pengguna memberikan rating pada film tambahan yang memicu incremental update.

c. Pemuatan *Embedding* Item

Sistem mengambil matrix *embedding* seluruh item dalam katalog: E_{items} = [e_{i1}, e_{i2} , ..., e_i N] dengan dimensi N × 256, di mana N adalah jumlah total film dalam database. Setiap *embedding* item e_i merupakan hasil dari proses *end-to-end training* yang menggabungkan representasi multimodal—dari poster melalui Vision Transformer (Palanisamy et al., 2025), sinopsis melalui BERT (Devlin et al., 2019), serta cast dan genre melalui *embedding* layers—dengan refinement kolaboratif dari propagasi LightGCN.

Embedding item mengandung dua jenis informasi yang terintegrasi: informasi konten multimodal dari poster, sinopsis, cast, dan genre; serta informasi kolaboratif dari pola interaksi pengguna yang dipelajari melalui propagasi graf LightGCN yang dikandungnya: pertama, informasi konten yang kaya dari berbagai modalitas yang telah diekstraksi dan difusikan melalui *cross-attention* transformer (Xia et al., 2024); kedua, sinyal kolaboratif yang menangkap pola preferensi kolektif dari struktur graf bipartit pengguna-item. Kombinasi informasi ini memungkinkan sistem memberikan rekomendasi yang akurat bahkan untuk item dengan interaksi terbatas (*cold-start* scenario), karena representasi multimodal tetap informatif meskipun tanpa riwayat kolaboratif yang ekstensif (Rodríguez-Baena et al., 2025).

Matrix E_{items} biasanya di-*cache* dalam memori untuk akses cepat mengingat *embedding* item relatif stabil dan hanya berubah ketika ada item baru ditambahkan ke katalog atau model dilatih ulang dengan data terbaru. Strategi caching ini merupakan trade-off antara memory footprint dengan kecepatan inference, di mana sistem mengalokasikan RAM untuk menyimpan *embedding* pre-computed dibandingkan dengan recompute on-demand yang akan menambah latency signifikan dan membuat sistem tidak praktis untuk deployment produksi.

d. Perhitungan Skor Prediksi

Sistem menghitung skor prediksi untuk seluruh item secara paralel menggunakan operasi matrix multiplication: scores = $e_u^T \times E_{items}$. Operasi ini menghasilkan vektor scores dengan panjang N, di mana setiap elemen $scores_i$ merepresentasikan tingkat kesesuaian antara preferensi pengguna u dengan karakteristik item i dalam ruang *embedding* bersama berdimensi 256.

Secara matematis, scores [i] = $e_u \cdot e_i$ adalah dot product yang mengukur similarity (dengan interpretasi cosine similarity jika *embedding* dinormalisasi) antara vektor pengguna dan vektor item. Skor tinggi mengindikasikan bahwa item memiliki karakteristik yang selaras dengan preferensi pengguna berdasarkan pola yang dipelajari model selama *training* phase. Komputasi ini sangat efisien karena dapat di-vectorize sepenuhnya dan di-accelerate menggunakan GPU atau optimized linear algebra libraries, memungkinkan pemrosesan katalog berisi puluhan ribu film dalam waktu kurang dari satu detik pada hardware standar.

Efisiensi komputasi merupakan keunggulan fundamental dari arsitektur *inference* ini dibandingkan dengan pendekatan yang memerlukan re-extraction fitur multimodal untuk setiap request (Mu & Wu, 2023). Dengan memanfaatkan *embedding* yang telah di-precompute selama training, sistem mengeliminasi overhead dari *forward pass* Vision Transformer dan BERT yang masing-masing memerlukan ratusan millisecond per item, sehingga dapat memberikan respons real-time bahkan pada scale produksi dengan jutaan pengguna aktif.

e. Pemeringkatan dan Seleksi Top-K

Vektor scores diurutkan secara descending untuk mengidentifikasi item dengan skor tertinggi, kemudian sistem memilih top-K item (dalam implementasi ini, K=10) sebagai kandidat rekomendasi final. Operasi sorting dapat dilakukan secara efisien menggunakan algoritma partial sort atau heap-based selection yang hanya mencari K elemen terbesar tanpa mengurutkan seluruh array, mengurangi kompleksitas waktu dari O(N log N) menjadi O(N log K) yang signifikan ketika K jauh lebih kecil dari N.

Skor prediksi mentah kemudian dinormalisasi ke skala 0-100% untuk menghasilkan *match percentage* (persentase kesesuaian antara preferensi pengguna dan karakteristik film) yang intuitif dan mudah dipahami oleh pengguna akhir. Normalisasi dilakukan menggunakan min-max scaling pada top-K scores, di mana skor tertinggi di-map ke 100% dan skor terendah dalam top-K di-map ke threshold minimum (misalnya 70% atau 75%), memberikan representasi visual yang meaningful tentang confidence level sistem terhadap setiap rekomendasi (Cremonesi et al., 2010). Formula normalisasi didefinisikan sebagai: match_percentage = ((score - min_score) / (max_score - min_score)) × 100%.

f. Keluaran Rekomendasi Top-K

Output Top-K rekomendasi sistem mengembalikan daftar 10 film rekomendasi yang telah di-ranking beserta metadata lengkap untuk ditampilkan kepada pengguna melalui antarmuka aplikasi. Setiap item dalam list mencakup: movie_id untuk referensi unik, title dan poster image untuk identifikasi visual, *match percentage* hasil normalisasi skor prediksi, genre tags untuk kategorisasi, rating rata-rata dari pengguna lain, dan tahun rilis. Informasi tambahan ini diambil dari database metadata film dan digabungkan dengan hasil prediksi model untuk membentuk response yang komprehensif dan *user-friendly*.

Match percentage memberikan transparansi kepada pengguna tentang tingkat confidence sistem terhadap setiap rekomendasi, memungkinkan pengguna membuat keputusan yang lebih informed dalam memilih film untuk ditonton (Herlocker et al., 2004). Film dengan match 95% mengindikasikan alignment yang sangat kuat antara karakteristik film dengan profil preferensi pengguna yang telah

dipelajari dari interaksi historis, sementara match 75% tetap relevan namun dengan confidence yang lebih moderate. Output ini kemudian dikirim ke *layer* presentasi—dalam implementasi ini menggunakan framework Streamlit (Akkem et al., 2024) untuk ditampilkan dalam format yang menarik dengan visualisasi *match percentage* menggunakan progress bar, rating stars, dan genre badges sesuai dengan desain mockup yang telah dijelaskan pada bagian Prototipe Antarmuka Sistem.

Perbedaan fundamental antara *training* flow dan *inference* flow terletak pada kompleksitas komputasi, tujuan, dan durasi eksekusi masing-masing fase. *Training* flow melibatkan serangkaian operasi komputasi intensif: ekstraksi fitur multimodal melalui *forward pass* Vision Transformer dan BERT yang telah di-*fine-tune*, fusion transformer untuk integrasi modalitas, propagasi graf LightGCN multi-layer, perhitungan BPR loss, dan *backpropagation* untuk update parameter seluruh komponen trainable (Kim et al., 2023; Rendle et al., 2009). Keseluruhan proses *training* memakan waktu berjam-jam hingga berhari-hari tergantung ukuran *dataset* dan konfigurasi hardware, dengan total waktu *training* pada *dataset* MovieLens 1M diperkirakan mencapai 6-12 jam pada single GPU modern.

Sebaliknya, *inference* flow hanya melibatkan tiga operasi lightweight: lookup *embedding* pengguna dari *cache* (complexity O(1)), lookup matrix *embedding* item dari *cache* (complexity O(1)), dan matrix multiplication untuk menghitung similarity scores (complexity O(N×d) di mana d=256 adalah dimensi embedding). Tidak ada feature extraction yang perlu dilakukan karena *embedding* item sudah mengandung informasi multimodal yang telah difusikan; tidak ada graph propagation yang perlu dijalankan karena *embedding* final sudah merupakan hasil agregasi multi-layer; dan tidak ada parameter updates karena model dalam mode *inference* read-only. Total latency dari request hingga response biasanya di bawah 2 detik untuk katalog berisi 10,000 film pada hardware consumer-grade, dengan bottleneck utama seringkali berasal dari database *query* untuk metadata tambahan bukan dari komputasi model itu sendiri (Fayyaz et al., 2020; Steck et al., 2021).

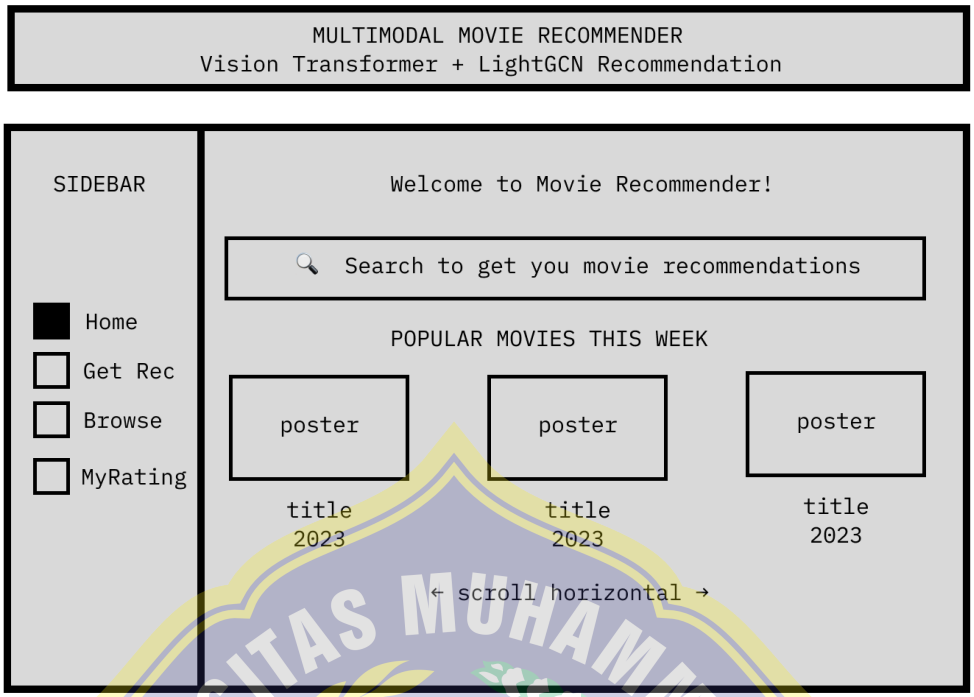
Strategi pre-computation dan caching *embedding* merupakan design decision yang mengoptimalkan trade-off antara memory footprint dengan kecepatan inference. Menyimpan *embedding* seluruh pengguna dan item dalam

RAM memerlukan alokasi memori signifikan—untuk 10,000 pengguna dan 10,000 item dengan dimensi 256 dan precision float32, total memory requirement sekitar 20,000 × 256 × 4 bytes ≈ 20 MB yang sangat manageable pada sistem modern. Investasi memory ini memungkinkan eliminasi recomputation overhead dan menjamin response time yang konsisten bahkan pada peak load, menjadikan sistem scalable untuk deployment produksi yang melayani ribuan concurrent users dengan requirement latency strict (Fayyaz et al., 2020).

3.3.2 Prototipe Antarmuka Sistem

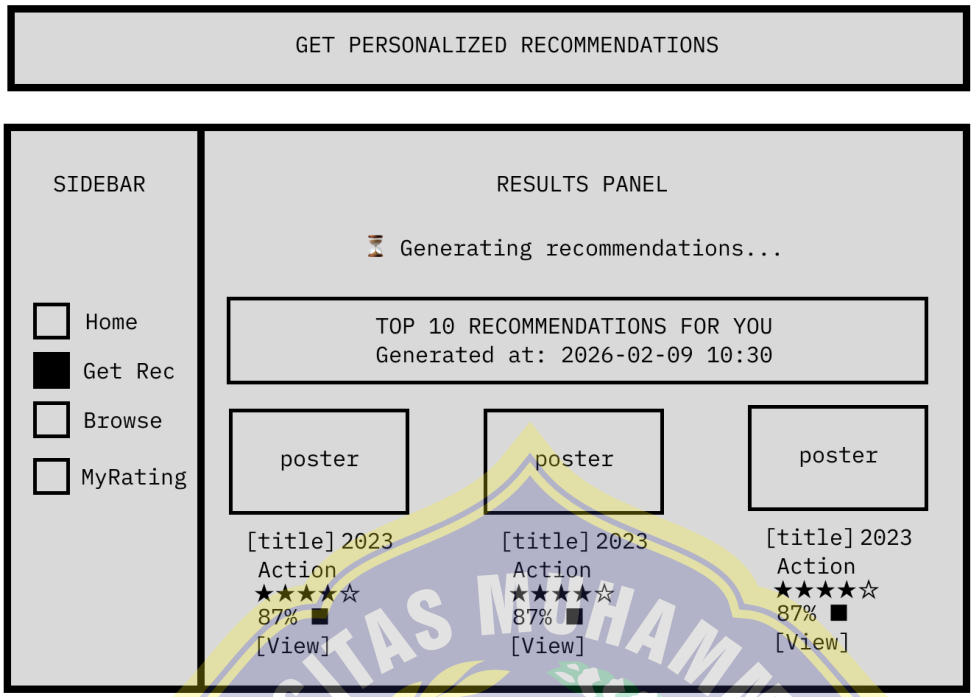
Prototipe antarmuka dibangun menggunakan Streamlit, sebuah *framework* Python untuk pembuatan aplikasi *machine learning* berbasis web (Akkem et al., 2024). Antarmuka terdiri dari enam halaman utama. Halaman Beranda menyediakan kolom pencarian film dengan fitur saran otomatis serta menampilkan film populer dan terbaru. Halaman Rekomendasi menampilkan daftar *top-10* film personal dari model Transformer-LightGCN dengan persentase kecocokan untuk setiap film. Halaman Detail Film menyajikan informasi lengkap meliputi poster, sinopsis, genre, *cast*, serta dua jenis *rating* prediksi model dan rata-rata pengguna lain dilengkapi fitur pemberian *rating*. Halaman Katalog memungkinkan pengguna menelusuri seluruh koleksi film dengan filter genre dan rentang tahun. Halaman Riwayat *Rating* menampilkan histori interaksi pengguna yang tersimpan dalam file *user_ratings.csv*. Halaman Pengaturan menyediakan konfigurasi preferensi pengguna.

Gambar 3.4 menunjukkan prototipe halaman Beranda yang berfungsi sebagai titik masuk pengguna ke sistem, menampilkan film populer dan kolom pencarian untuk eksplorasi awal.



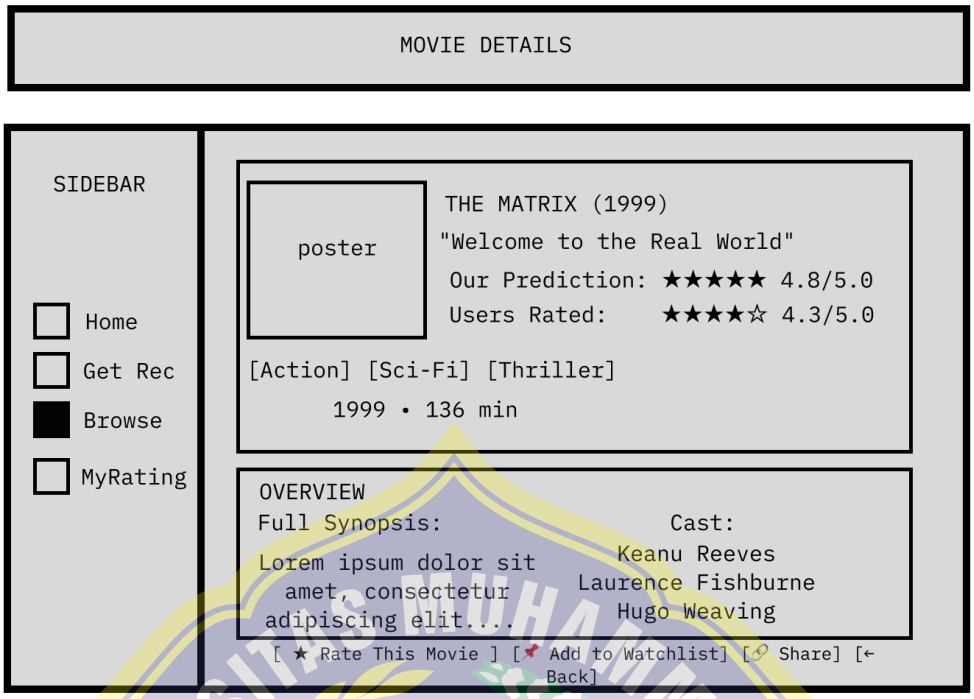
Gambar 3. 4 Prototipe Halaman Beranda

Gambar 3.5 menunjukkan prototipe halaman Rekomendasi yang merupakan fitur utama sistem, menampilkan daftar *top-10* film personal dengan persentase kecocokan dari model Transformer-LightGCN.



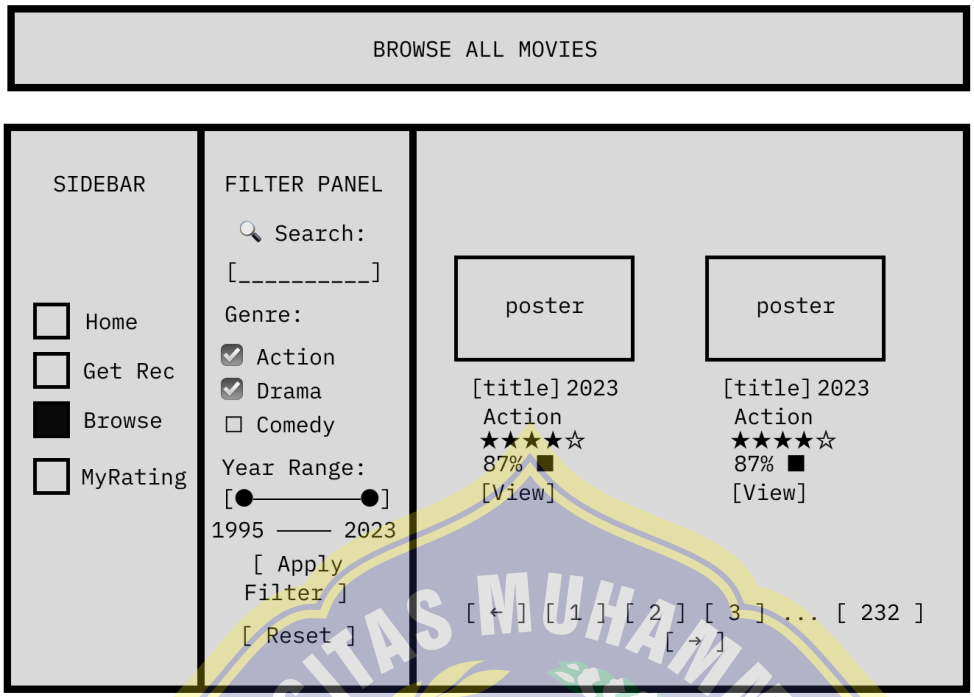
Gambar 3. 5 Prototipe Halaman Rekomendasi Personal

Gambar 3.6 menunjukkan prototipe halaman Detail Film yang menyajikan informasi lengkap setiap film beserta fitur pemberian *rating* oleh pengguna.



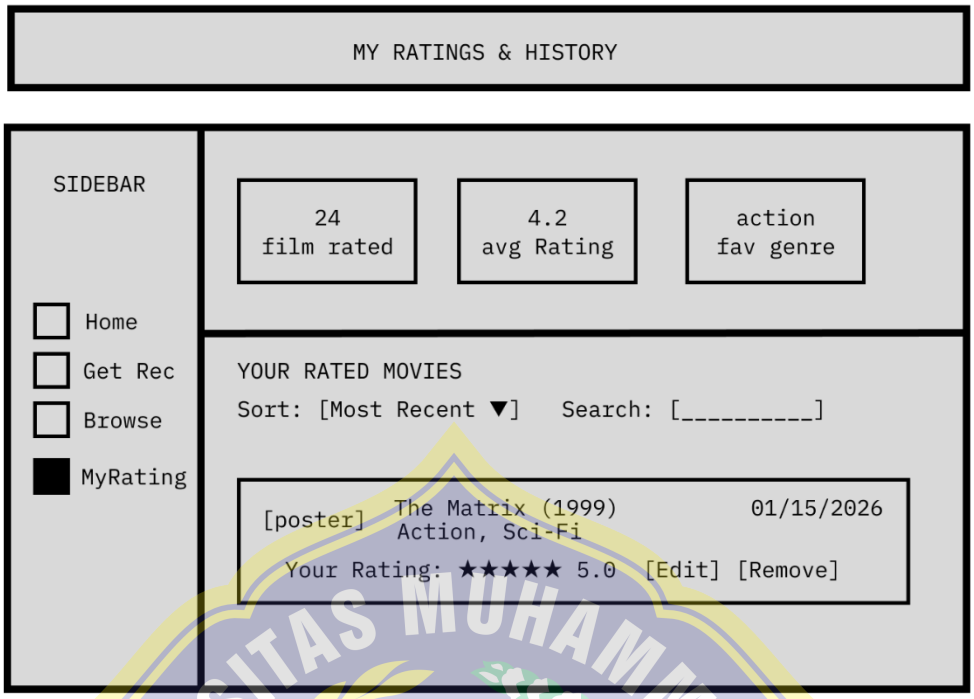
Gambar 3. 6 Prototipe Halaman Detail Film

Gambar 3.7 menunjukkan prototipe halaman Katalog yang memungkinkan pengguna menelusuri seluruh koleksi film dengan filter pencarian.



Gambar 3. 7 Prototipe Halaman Katalog Film

Gambar 3.8 menunjukkan prototipe halaman Riwayat *Rating* yang mencatat seluruh histori interaksi pengguna terhadap film.



Gambar 3. 8 Prototipe Halaman Riwayat *Rating*

3.3.3 Input dan Output Sistem

Sistem rekomendasi film yang dikembangkan dirancang dengan spesifikasi input dan output yang terstruktur untuk mendukung proses pembelajaran model dan penyajian rekomendasi kepada pengguna akhir. Bagian ini menjelaskan secara eksplisit format data yang diterima sistem pada berbagai fase operasional, struktur output yang dihasilkan, serta alur transformasi data dari input hingga menjadi hasil rekomendasi yang dapat digunakan.

Input Sistem

Sistem menerima dua kategori input yang berbeda berdasarkan fase operasionalnya, yaitu input untuk fase *training* dan input untuk fase inferensi.

a) Input Fase Training

Pada fase training, sistem memerlukan data interaksi pengguna-film dan metadata multimodal yang diperoleh dari dua sumber utama. Data interaksi berasal dari MovieLens dalam format Comma-Separated Values (CSV) yang mencakup

empat atribut esensial (Harper & Konstan, 2016) Struktur data input dari file *ratings.csv* dan contoh format *output* rekomendasi disajikan pada Lampiran 3.

b) Input Fase Inferensi

Pada fase *inference*, sistem menerima *user ID* sebagai input utama dengan parameter opsional berupa jumlah rekomendasi (default 10), filter genre, dan rentang tahun rilis.

Output Sistem

Sistem menghasilkan tiga kategori output yang disesuaikan dengan konteks penggunaannya, meliputi output fase training, output fase inferensi, dan output evaluasi.

a) Output Fase Training

Selama proses training, sistem menghasilkan beberapa artefak teknis untuk monitoring dan reproduksibilitas. Model *checkpoint* disimpan dalam format PyTorch (.pth) yang berisi state dictionary model terlatih dengan ukuran berkisar 200-500 MB bergantung pada ukuran *dataset*. *Training* logs mencatat metrik pembelajaran per *epoch* dalam format CSV atau TensorBoard events, mencakup nilai BPR loss, metrik validasi (Precision@K, Recall@K, NDCG@K), *learning rate* aktual, dan waktu komputasi per epoch. File *embedding* (.npy) menyimpan representasi vektor terlatih untuk seluruh pengguna dan item dalam bentuk matrix NumPy yang digunakan untuk mempercepat proses inferensi.

b) Output Fase Inferensi

Output utama sistem adalah daftar rekomendasi top-N yang disajikan kepada pengguna akhir dalam format JSON terstruktur Contoh format *output* rekomendasi disajikan pada Lampiran 3.

c) Output Evaluasi

Untuk keperluan validasi penelitian, sistem menghasilkan output evaluasi komprehensif dalam format tabel yang mencakup nilai metrik standar rekomendasi

top-N. Hasil evaluasi dilaporkan sebagai rata-rata dari tiga replikasi eksperimen dengan *random seed* berbeda untuk memastikan validitas statistik (X. He et al., 2020). Format output evaluasi mengikuti konvensi Metric@K: mean ± std di mana mean merepresentasikan rata-rata performa dan std menunjukkan standar deviasi antar replikasi. Sebagai contoh, hasil evaluasi dapat dilaporkan sebagai "Precision@10: 0.3245 ± 0.0089" yang mengindikasikan performa stabil dengan variasi minimal antar eksperimen.

Transformasi data dari input hingga menjadi output rekomendasi melibatkan beberapa tahapan pemrosesan yang terintegrasi. Pada fase training, data ratings.csv dan metadata TMDb menjalani preprocessing untuk menghasilkan representasi multimodal melalui ekstraksi fitur menggunakan ViT dan BERT. Representasi ini kemudian difusikan melalui modul *cross-attention* dan dikombinasikan secara aditif dengan *embedding* kolaboratif item melalui parameter α sebelum memasuki propagasi graf LightGCN. Model dilatih menggunakan fungsi objektif BPR (Rendle et al., 2009) hingga menghasilkan model terlatih dan file *embedding* yang disimpan untuk inferensi.

Pada fase inferensi, sistem menerima userId dari pengguna, kemudian melakukan retrieval *embedding* pengguna dan *embedding* seluruh item dari file yang telah disimpan. Skor preferensi dihitung melalui matrix multiplication antara *embedding* pengguna dengan matriks *embedding* item, menghasilkan vektor skor untuk seluruh film dalam katalog. Sistem kemudian melakukan seleksi top-K film dengan skor tertinggi, memperkaya hasil dengan metadata lengkap dari database, dan memformat output dalam struktur JSON yang siap ditampilkan di antarmuka pengguna. Keseluruhan pipeline dirancang untuk memastikan latensi inferensi yang rendah sambil mempertahankan kualitas rekomendasi yang tinggi.

Output rekomendasi yang dihasilkan sistem divisualisasikan kepada pengguna melalui antarmuka aplikasi berbasis Streamlit dalam beberapa bentuk tampilan. Daftar top-10 rekomendasi ditampilkan sebagai kartu film (*movie card*) yang memuat poster film, judul, tahun rilis, label genre, rating bintang, dan persentase kesesuaian (*match percentage*) yang merepresentasikan nilai *confidence score* dalam skala 0–100% sebagaimana ditunjukkan pada (Gambar 3.5). Detail lebih lanjut mengenai setiap film rekomendasi dapat diakses melalui halaman detail

film (Gambar 3.6), yang menampilkan prediksi rating sistem secara berdampingan dengan rata-rata rating pengguna lain, sehingga memberikan perspektif ganda kepada pengguna dalam mengambil keputusan. Seluruh interaksi pengguna terhadap hasil rekomendasi, termasuk pemberian rating dan penambahan ke daftar tonton, dicatat secara otomatis dan dapat ditinjau kembali pada halaman riwayat rating (Gambar 3.8).

3.4 Rencana Evaluasi

Protokol evaluasi dirancang dengan fokus pada skenario rekomendasi top-N berbasis *implicit feedback* (data interaksi tidak langsung seperti pemberian rating) menggunakan korpus MovieLens dalam varian 100K dan 1M yang telah diintegrasikan dengan metadata deskriptif dari TMDb. Evaluasi mencakup pengukuran performa kuantitatif melalui metrik standar, analisis stratifikasi berdasarkan karakteristik item, studi ablasi komponen arsitektur, serta validasi kualitatif melalui antarmuka demonstrasi interaktif.

Performa model dianggap mencapai hasil optimal apabila memenuhi dua kriteria utama. Kriteria pertama, model hibrida yang diusulkan harus mengungguli seluruh metode *baseline* pada metrik utama NDCG@K dan HR@K di kedua *dataset* yang digunakan. Ketiga *baseline* dipilih karena mewakili pendekatan yang berbeda: ItemKNN mewakili *collaborative filtering* klasik berbasis kemiripan, PureLightGCN mewakili pemodelan graf kolaboratif tanpa informasi konten, dan VBPR mewakili integrasi fitur visual dengan *collaborative filtering*. Dengan mengungguli ketiganya, arsitektur hibrida dapat dinyatakan memberikan nilai tambah dibandingkan pendekatan yang sudah ada. Kriteria kedua, peningkatan performa tersebut harus terbukti signifikan secara statistik melalui *paired t-test* dengan tingkat signifikansi $\alpha = 0{,}01$ dan didukung oleh pengukuran *effect size* menggunakan Cohen's d (Cohen, 2013). Pengujian signifikansi statistik diperlukan untuk memastikan bahwa perbedaan performa bukan disebabkan oleh variasi acak dari inisialisasi parameter. Pendekatan evaluasi berbasis perbandingan *baseline* dengan uji signifikansi ini mengikuti standar yang diterapkan pada penelitian NCF, di mana seluruh peningkatan performa diverifikasi melalui *paired t-test* (X. He et al., 2017).

Threshold konversi *rating* eksplisit menjadi *implicit feedback* ditetapkan pada nilai ≥3 mengikuti pendekatan binarisasi yang diterapkan oleh (Anelli et al., 2021) pada *dataset* MovieLens 1M, di mana *rating* bernilai 3 atau lebih dipertahankan sebagai *implicit feedback* positif. Pemilihan titik tengah skala ini didasarkan pada pertimbangan bahwa *rating* bernilai 1 dan 2 mengindikasikan ketidaksukaan pengguna terhadap film, sehingga tidak tepat diperlakukan sebagai sinyal positif.

Proses pencapaian kriteria keberhasilan dilakukan secara iteratif sesuai kerangka kerja pada Gambar 3.1. Pada setiap iterasi, konfigurasi *hyperparameter* dievaluasi dan disesuaikan berdasarkan performa yang diperoleh. Apabila kriteria belum tercapai setelah suatu iterasi, langkah perbaikan yang dilakukan meliputi penyesuaian *learning rate* dan *batch size* untuk menstabilkan proses pelatihan, perubahan strategi *negative sampling* untuk meningkatkan kualitas sampel pelatihan, modifikasi mekanisme integrasi multimodal untuk memperbaiki representasi item, serta penyesuaian *threshold* data *implicit feedback* untuk memperbanyak sinyal pelatihan. Iterasi dihentikan ketika kedua kriteria keberhasilan terpenuhi secara konsisten pada minimal tiga replikasi dengan *random seed* yang berbeda.

Performa sistem diukur menggunakan empat metrik standar rekomendasi *top-N* yang dievaluasi pada nilai K = 5, 10, 20 sebagaimana telah didefinisikan pada Bab II: Precision@K persamaan (2.6) mengukur proporsi item relevan dalam daftar rekomendasi, Recall@K persamaan (2.7) menilai cakupan terhadap seluruh item relevan, NDCG@K persamaan (2.8) dan (2.9) mengukur kualitas *ranking* dengan mempertimbangkan posisi item relevan, serta Hit Ratio@K persamaan (2.10) mengukur probabilitas keberhasilan rekomendasi di mana *hit* terjadi jika minimal satu item relevan muncul dalam *top-K* (Cremonesi et al., 2010; Herlocker et al., 2004).

Protokol evaluasi menggunakan pendekatan *sampled evaluation* yang merupakan standar dalam literatur sistem rekomendasi sebagaimana diterapkan oleh He et al. (2017) pada NCF. Untuk setiap pengguna dalam set *testing*, satu item positif (*test item*) di-*ranking* bersama 99 item negatif yang diambil secara acak dari item yang belum pernah berinteraksi dengan pengguna tersebut. Metrik

Precision@K, Recall@K, NDCG@K, dan HR@K kemudian dihitung berdasarkan posisi item positif dalam *ranking* 100 item ini. Pendekatan *sampled evaluation* dipilih karena lebih efisien secara komputasi dibandingkan *full ranking* terhadap seluruh item dalam katalog, dan memungkinkan hasil yang dapat dibandingkan secara langsung dengan penelitian lain yang menggunakan protokol serupa. Ringkasan parameter evaluasi yang digunakan dalam penelitian ini disajikan pada Tabel 3.3.

Tabel 3. 3 Parameter Evaluasi

Parameter	Nilai
Protokol evaluasi	*Sampled evaluation* (1 positif + 99 negatif)
Metrik utama	NDCG@K, HR@K
Metrik pendukung	Precision@K, Recall@K
Nilai K	5, 10, 20
Strategi partisi	*Leave-last-out* per pengguna
Jumlah replikasi	3 (*seed* 42, 123, 456)
Format pelaporan	*Mean* ± standar deviasi
Uji signifikansi	*Paired t-test* (α = 0,01)
Effect size	Cohen's d
Kriteria keberhasilan	Mengungguli seluruh *baseline* + signifikan secara statistik

Strategi partisi data mengadopsi pendekatan *leave-last-out* per pengguna untuk mensimulasikan kondisi deployment temporal yang realistis. Untuk setiap pengguna, interaksi terakhir berdasarkan *timestamp* dipisahkan sebagai set testing, interaksi kedua terakhir sebagai set validasi, dan sisanya sebagai set training. Pendekatan ini memastikan model dilatih pada interaksi historis dan dievaluasi pada interaksi masa depan.

Evaluasi komprehensif diperluas untuk menganalisis ketahanan model terhadap berbagai skenario *deployment*. Analisis *cold-start* dilakukan dengan mengevaluasi performa pada subset item yang memiliki jumlah interaksi terbatas

(kurang dari 20 rating dalam set training) untuk memvalidasi kontribusi representasi multimodal dalam mengatasi keterbatasan data interaksi. *Stratified evaluation* berdasarkan lima kategori popularitas dilakukan dengan membagi item menurut distribusi jumlah interaksi: *very long-tail* (0-20 ratings), *long-tail* (21-50 ratings), *mid-popularity* (51-100 ratings), *popular* (101-200 ratings), dan *very popular* (>200 ratings). Stratifikasi ini mengukur kemampuan model dalam menangani distribusi *long-tail* dan memastikan performa tidak bias terhadap item populer (Klimashevskaia et al., 2024). Pengukuran efisiensi komputasional mencakup waktu pelatihan per epoch, latensi inferensi per pengguna, dan konsumsi memori GPU selama *training* dan inference.

Komparasi performa dilakukan terhadap tiga metode baseline yang representatif untuk mengevaluasi kontribusi komponen multimodal dan graf secara independen. Pertama, Item-based K-Nearest Neighbor (ItemKNN) (Sarwar et al., 2001) sebagai *collaborative filtering* klasik berbasis similarity yang tidak menggunakan representasi multimodal maupun struktur graf. Kedua, LightGCN (X. He et al., 2020) sebagai baseline berbasis graf murni tanpa informasi konten multimodal untuk mengukur kontribusi sinyal kolaboratif dari propagasi graf. Ketiga, Visual Bayesian Personalized Ranking (VBPR) (R. He & McAuley, 2015) sebagai baseline multimodal berbasis CNN yang mengintegrasikan fitur visual dari poster film tanpa komponen propagasi graf. Pemilihan ketiga baseline ini memungkinkan evaluasi komprehensif untuk memvalidasi bahwa arsitektur hibrida yang diusulkan mengungguli pendekatan *graph-only*, *multimodal-only*, dan metode *collaborative filtering* klasik.

Studi ablasi sistematis dilakukan untuk memvalidasi kontribusi setiap modalitas dan komponen arsitektur. Komponen dinonaktifkan secara progresif dalam konfigurasi berikut: (1) tanpa representasi visual dari poster, (2) tanpa representasi tekstual dari sinopsis, (3) tanpa *embedding cast*, (4) tanpa *embedding* genre, (5) tanpa modul fusi *cross-attention* (diganti dengan konkatenasi sederhana), dan (6) tanpa mekanisme propagasi LightGCN (hanya menggunakan *embedding* multimodal). Setiap konfigurasi dievaluasi menggunakan metrik yang sama untuk mengukur dampak bertahap masing-masing elemen terhadap peningkatan akurasi rekomendasi.

Untuk memastikan validitas statistik dan reprodusibilitas hasil eksperimen, seluruh rangkaian evaluasi direplikasi sebanyak tiga kali menggunakan *random seed* yang berbeda pada setiap replikasi, yaitu *seed* = 42, *seed* = 123, dan *seed* = 456. Penetapan *seed* secara eksplisit mengontrol seluruh sumber stokastisitas dalam *pipeline* eksperimen, mencakup inisialisasi parameter model, urutan *shuffling* data training, dan proses *negative sampling* untuk konstruksi triplet BPR. Praktik ini mengikuti prosedur yang lazim diterapkan dalam evaluasi sistem rekomendasi berbasis *deep learning*, sebagaimana ditunjukkan oleh (Ji et al., 2023) yang melaporkan rata-rata dari tiga percobaan acak dengan *seed* berbeda pada evaluasi model BPR dan LightGCN menggunakan *dataset* MovieLens untuk mereduksi *noise* dari faktor-faktor acak. Performa akhir dilaporkan dalam format mean ± standar deviasi untuk setiap metrik, misalnya "Precision@10: 0.3245 ± 0.0089", di mana standar deviasi yang rendah mengkonfirmasi konsistensi performa model terlepas dari kondisi inisialisasi. Prosedur replikasi yang sama diterapkan pada seluruh model baseline dan konfigurasi studi ablasi untuk memastikan kesetaraan kondisi evaluasi (*fair comparison*) antar metode yang dibandingkan.

Signifikansi statistik perbedaan performa divalidasi menggunakan *paired t-test* dengan tingkat signifikansi $\alpha = 0.01$. Pengukuran *effect size* dilakukan melalui Cohen's d yang diformulasikan dalam persamaan (3.5):

$$d = \frac{\bar{x}_1 - \bar{x}_2}{s_{\text{pooled}}}, s_{\text{pooled}} = \sqrt{\frac{(n_1 - 1)s_1^2 + (n_2 - 1)s_2^2}{n_1 + n_2 - 2}} \quad (3.5)$$

di mana $\bar{x}_1$dan $\bar{x}_2$ adalah rata-rata performa dua metode, s_1 dan s_2 adalah standar deviasi, serta n_1 dan n_2 adalah ukuran sampel. Cohen's d diinterpretasikan sebagai *small effect* ($d \approx 0.2$), *medium effect* ($d \approx 0.5$), atau *large effect* ($d \geq 0.8$) untuk mengkuantifikasi magnitude perbaikan (Cohen, 2013).

Selain evaluasi kuantitatif berbasis metrik, validasi fungsionalitas sistem dilakukan melalui antarmuka demonstrasi interaktif yang dikembangkan menggunakan Streamlit. Framework ini memungkinkan pengembangan prototip sistem *machine learning* secara cepat dengan antarmuka web yang responsif

(Akkem et al., 2024). Melalui antarmuka ini, pengguna dapat memilih film referensi sebagai input preferensi, kemudian sistem menampilkan daftar rekomendasi top-K yang mencakup poster film, judul, informasi genre, tahun rilis, dan skor prediksi hasil inferensi model Transformer-LightGCN. Pendekatan ini memungkinkan evaluasi kualitatif terhadap relevansi, diversitas, dan interpretabilitas output rekomendasi sebagai pelengkap pengukuran metrik formal. Antarmuka juga menyediakan visualisasi distribusi skor prediksi dan perbandingan rekomendasi dari berbagai model untuk analisis komparatif.

Hasil evaluasi kuantitatif divisualisasikan melalui serangkaian grafik komparatif menggunakan matplotlib dan seaborn untuk memfasilitasi interpretasi performa (Hunter, 2007; Waskom, 2021). Visualisasi mencakup bar chart perbandingan metrik evaluasi antar model dengan *error bars* yang merepresentasikan standar deviasi dari tiga replikasi, grouped bar chart stratifikasi popularitas untuk menganalisis performa pada lima kategori item (very *long-tail* hingga very popular), serta heatmap studi ablasi yang menampilkan kontribusi relatif setiap komponen arsitektur melalui intensitas warna. Precision-recall curve untuk setiap model pada *dataset* MovieLens 100K dan 1M disajikan untuk memberikan gambaran *trade-off metrik*, sementara *confusion matrix* menampilkan distribusi prediksi pada kategori relevan dan tidak relevan untuk analisis komprehensif (Herlocker et al., 2004).

Seluruh eksperimen dieksekusi pada infrastruktur GPU dengan konfigurasi *hyperparameter* yang terstandarisasi untuk menjamin ekuitas komparatif dan *reproducibility* hasil eksperimen. Hyperparameter utama meliputi: dimensi *embedding* 256, *learning rate* 10^{-3}, *batch size* 1024, jumlah *layer* LightGCN $K = 3$, *dropout rate* 0.1, dan koefisien regularisasi L2 $\lambda = 10^{-5}$. Seluruh kode eksperimen dan konfigurasi akan didokumentasikan secara komprehensif untuk memfasilitasi replikasi penelitian.