

**SISTEM REKOMENDASI FILM MENGGUNAKAN DATA USER-END  
DAN KNOWLEDGE GRAPH CONVOLUTIONAL NETWORK**

**PROPOSAL**

Diajukan untuk Memenuhi Salah Satu Syarat  
Matakuliah Tugas Akhir 1  
Jenjang Strata 1 pada Program Studi Informatika  
Universitas Jenderal Achmad Yani

Oleh:

Nama : Muhammad Rizki Yanuar

NIM : 3411211062

Pembimbing 1

Fajri Rakhmat Umbara, S.T., M.T

NID: 412185888

Pembimbing 2

Agus Komarudin, S.Kom, M.T.

NID: 412175878



PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN INFORMATIKA  
UNIVERSITAS JENDERAL ACHMAD YANI

2025

## ABSTRAK

Dalam era digital, sistem rekomendasi memainkan peran penting dalam membantu pengguna menemukan film sesuai preferensi mereka. Namun, metode tradisional seperti collaborative filtering dan content-based filtering sering menghadapi masalah sparsity dan cold start, terutama untuk pengguna baru. Penelitian ini mengusulkan implementasi Knowledge Graph Convolutional Network (KGCN) yang memanfaatkan data user-end untuk mengatasi keterbatasan tersebut. Knowledge Graph (KG) digunakan untuk mengintegrasikan atribut film dan informasi pengguna dalam struktur graf yang terarah, memungkinkan sistem untuk menghasilkan rekomendasi lebih relevan. Metode Graph Convolutional Network (GCN) diaplikasikan untuk mengagregasi informasi dari tetangga terdekat, memanfaatkan hubungan antar entitas dalam graf.

Penelitian ini bertujuan mengembangkan sistem rekomendasi film yang lebih responsif terhadap preferensi pengguna, dengan mengevaluasi performa KGCN dibandingkan metode tradisional. Dataset MovieLens digunakan sebagai sumber data utama untuk menguji efektivitas pendekatan ini. Hasil penelitian diharapkan mampu meningkatkan akurasi rekomendasi melalui integrasi umpan balik pengguna secara eksplisit maupun implisit. Dengan pendekatan ini, sistem rekomendasi tidak hanya memanfaatkan atribut item, tetapi juga memperhitungkan preferensi dan interaksi pengguna secara komprehensif. Penelitian ini memberikan kontribusi signifikan terhadap pengembangan sistem rekomendasi berbasis KG, terutama dalam konteks permasalahan cold start dan sparsity.

Kata Kunci: Knowledge Graph, Graph Convolutional Network, Sistem Rekomendasi Film, Cold Start, Sparsity, Data User-End

## 1. Latar Belakang Masalah

Dalam industri film, sistem rekomendasi sangat penting untuk membantu pengguna menemukan film yang sesuai dengan preferensi mereka. Dengan rilisnya ribuan film setiap tahun, menimbulkan tantangan besar yaitu bagaimana menyaring pilihan sehingga pengguna dengan mudah menemukan film yang cocok bagi mereka. Meskipun telah digunakan secara luas, metode tradisional seperti *collaborative filtering* dan *content-based filtering* sering kali menghadapi masalah *sparsity* dan *cold start* [1], dikarenakan data interaksi pengguna dengan item terbatas atau tidak ada untuk pengguna baru.

Untuk mengatasi masalah ini, penelitian telah beralih ke pendekatan *deep learning*, termasuk penggunaan *Knowledge Graph Convolutional Networks* (KGCN). *Knowledge Graph* (KG) merupakan sebuah grafik yang menyimpan informasi terarah dengan simpul yang mewakili entitas (seperti pengguna, item atau atribut) dan tepi yang mewakili hubungan antar entitas. Salah satu keunggulan KG adalah kemampuan dalam menghubungkan berbagai atribut dan informasi yang relevan untuk pengguna dan item. *Graph Convolutional Network* (GCN) merupakan metode yang dirancang untuk bekerja dengan data yang terstruktur dalam bentuk graf. Pada GCN terdapat fitur utama yaitu kemampuan dalam mengagregasi dan menggabungkan informasi dari tetangga terdekat, desain semacam ini memiliki 2 keuntungan: (1) Struktur kedekatan lokal secara efektif dicatat dan disimpan di setiap entitas melalui metode agregasi. (2) Tetangga diberi bobot berdasarkan skor yang bergantung pada pengguna tertentu dan hubungan antar entitas. Skor ini menggambarkan informasi konten dari KG serta minat relasional pengguna itu sendiri [2].

Berbagai penelitian telah dilakukan dalam pengembangan sistem rekomendasi yang memanfaatkan data item-end dan user-item dengan menggunakan berbagai metode. Pada penelitian [2] penggunaan KGCN dengan memanfaatkan data item-ends dan menggunakan *uniformly sampling fixed-size neighborhood* untuk mengatasi jumlah nodes tetangga yang sangat besar, yang menghasilkan rekomendasi berdasarkan atribut-atribut item dan mengurangi waktu

komputasi karena menentukan jumlah nodes tetangga yang akan diambil. Penelitian ini menunjukkan bahwa KGCN dapat efektif dalam mengidentifikasi keterkaitan antar item dari atribut yang terdapat dalam *Knowledge Graph* dan *uniformly sampling fixed-size neighborhood* menunjukkan peningkatan rata-rata AUC 4,4% karena dapat menangkap informasi yang jauh lebih banyak. Meskipun penelitian ini memberikan hasil signifikan terhadap pengembangan sistem rekomendasi berbasis item, masih terdapat celah dalam pemanfaatan data user-end dan teknik sampling. Penelitian lebih lanjut diperlukan untuk mengeksplorasi bagaimana umpan balik pengguna, baik bersifat eksplisit maupun implisit dapat diintegrasikan ke dalam model KGCN dan bagaimana teknik *importance sampling* dapat meningkatkan akurasi model, dengan demikian sistem rekomendasi tidak hanya bergantung pada atribut item tetapi juga mempertimbangkan preferensi dan pengalaman pengguna serta menangkap informasi yang jauh lebih relevan ketika pemilihan nodes tetangga.

Pada penelitian [3] penggunaan sistem rekomendasi hybrid yang menggabungkan *Collaborative Filtering*, *Content-Based Filtering*, dan teknik *Self-Organizing* (SOM) dengan memanfaatkan data user-item. Pendekatan hybrid ini memanfaatkan metode *weighted* dengan menggabungkan skor dari collaborative filtering dan content-based filtering untuk menghasilkan satu rekomendasi dan *Feature Augmentation* dengan memanfaatkan output dari hasil rekomendasi *collaborative* dan *content-based* kemudian menjadikan hasil rekomendasi tersebut sebagai informasi tambahan dalam menciptakan model rekomendasi. Penelitian ini menyarankan penerapan *Deep Learning* seperti *Graph Neural Network* (GNN) untuk menangkap pola interaksi yang lebih kompleks, pengembangan strategi untuk menangani masalah cold start bagi pengguna baru atau item baru, dan mengintegrasikan data user-end untuk mengeksplorasi bagaimana umpan balik eksplisit dan implisit dari pengguna dapat meningkatkan akurasi rekomendasi.

Oleh karena itu penelitian ini bertujuan untuk meningkatkan akurasi dan relevansi rekomendasi film bagi pengguna dengan memanfaatkan data user-end dan *importance sampling* menggunakan *Knowledge Graph Convolutional Network*

(KGCN). Target luaran yang ingin dicapai mencakup pengembangan model rekomendasi, serta evaluasi performa sistem rekomendasi yang diusulkan dibandingkan dengan metode tradisional. Penelitian ini akan mengimplementasikan data user-end dan teknik *importance sampling* menggunakan *Knowledge Graph Convolutional Network* (KGCN) untuk menciptakan sistem rekomendasi film yang lebih responsive terhadap preferensi pengguna.

## **2. Rumusan Masalah**

Penelitian ini bertujuan untuk menyelidiki efektivitas Knowledge Graph Convolutional Network (KGCN) dalam menganalisis preferensi pengguna terhadap film menggunakan data user-end dan importance sampling. Dengan memanfaatkan data interaksi pengguna dan pengambilan informasi yang relevan dengan pengguna, penelitian ini juga akan mengeksplorasi bagaimana teknik importance sampling pada knowledge graph dapat meningkatkan akurasi model pada sistem rekomendasi film.

## **3. Tujuan Penelitian**

- a. Mengembangkan penggunaan metode Knowledge Graph Convolutional Network menggunakan data user-end untuk meningkatkan akurasi dan menangani masalah cold start, sparsity.
- b. Penggunaan metode *importance sampling* pada Knowledge Graph Convolutional Network untuk mengurangi waktu komputasi dan meningkatkan akurasi model.

## **4. Batasan Masalah**

- a. Penelitian ini dibatasi pada penggunaan Knowledge Graph Convolutional Network sebagai metode utama dalam menganalisis dan merekomendasikan film dengan menggabungkan data user-end dan data item-end serta penggunaan teknik *importance sampling*.
- b. Penelitian ini hanya menggunakan dataset dari platform MovieLens

## **5. Luaran dan Manfaat Penelitian**

Luaran dari penelitian ini membuat sistem rekomendasi film menggunakan Knowledge Graph Convolutional Network (KGCN) dengan memanfaatkan data user-end dan importance sampling, sehingga dapat meningkatkan akurasi model dan memberikan rekomendasi yang relevan berdasarkan preferensi pengguna serta dapat mengurangi waktu komputasi model dalam pembelajaran. Manfaat yang dapat diambil dari penelitian ini adalah kemampuan untuk memberikan rekomendasi film yang lebih relevan dan personal kepada pengguna, dengan mempertimbangkan interaksi dan umpan balik dari pengguna.

## 6. Tinjauan Pustaka

### a. Data Interaksi

Pada bentuk sistem rekomendasi, terdapat interaksi yang secara *implicit* diberikan oleh pengguna mengenai preferensi yang dipilih (seperti, melihat, klik, membeli, menilai). Terdapat sekumpulan pengguna di definisikan sebagai  $\mathbf{U}$  dan sekumpulan item di definisikan sebagai  $\mathbf{V}$ . Matrix interaksi user-item didefinisikan sebagai:

$$\mathbf{Y} \in \mathbb{R}^{M \times N}$$

Penjelasan:

$\mathbf{Y}$  = Matrix interaksi antara pengguna dan item

$\mathbb{R}$  = Himpunan bilangan real dalam matrix  $\mathbf{Y}$

$\mathbf{M}$  = Jumlah baris yang mewakili jumlah pengguna dalam matrix

$\mathbf{N}$  = Jumlah kolom item yang mewakili jumlah item dalam matrix

Matrix pengguna-item tersebut disesuaikan dengan umpan balik *implicit* pengguna, di mana  $y_{uv} = 1$  yang mengindikasikan bahwa user  $\mathbf{U}$  tersebut berinteraksi dengan item  $\mathbf{V}$ , dan sebaliknya  $y_{uv} = 0$  [2].

### b. Knowledge Graph

Knowledge Graph (KG) adalah struktur data yang menyimpan informasi dalam bentuk entitas dan hubungan antar entitas. Setiap node dalam KG merujuk

pada entitas atau konsep tertentu, sedangkan edges merujuk pada jenis hubungan antara entitas tersebut. KG digunakan untuk memperkaya model preferensi pengguna dengan informasi tambahan yang tidak hanya bergantung pada interaksi pengguna-item sehingga rekomendasi menjadi lebih relevan dan akurat [1].

Struktur Knowledge Graph:

- Nodes

Dalam konteks sistem rekomendasi film, nodes dapat mencakup berbagai entitas seperti pengguna, film, genre, sutradara dan aktor. Setiap node mewakili sebuah item atau entitas yang relevan dengan domain rekomendasi.

- Edges

Edges merupakan penghubung antar node yang menampilkan koneksi di antara entitas. Misalnya “hasGenre” (memiliki genre), “hasDirector” (memiliki sutradara), atau “rates” [4]

*Knowledge graph* menyimpan struktur informasi dalam bentuk grafik heterogen, pada knowledge graph didefinisikan  $E$  menjadi sekumpulan entitas dan  $R$  menjadi sekumpulan relasi yang mencakup hubungan dari 2 arah, baik hubungan biasa ataupun sebaliknya, seperti relasi ‘director’ dan ‘directed-by’. Pada knowledge Graph dibuat sebuah triplet didefinisikan sebagai:

$$\mathcal{G} = \{(h, r, t) \mid h \in e, t \in e, r \in R\}$$

Penjelasan:

$\mathcal{G}$  = Sekumpulan triple yang menyimpan informasi hubungan antar entitas

$(h, r, t)$  = kepala, relasi dan ekor dari sebuah triple pengetahuan

$e$  = Himpunan entitas dalam knowledge graph

$R$  = Himpunan relasi dalam knowledge graph [5]

*Knowledge graph* dapat diperkaya dengan informasi tentang preferensi pengguna dalam sistem rekomendasi, Sistem rekomendasi bertujuan untuk mempelajari fungsi dalam memprediksi probabilitas interaksi pengguna dengan item. Matriks

interaksi pengguna-item  $Y$  serta *Knowledge Graph*  $\mathcal{G}$ , digunakan untuk memprediksi menggunakan fungsi prediksi [6].

$$\hat{y}_{uv} = \mathcal{F}(\mathbf{u}, \mathbf{v} | \boldsymbol{\theta}, Y, \mathcal{G})$$

$\hat{y}_{uv}$  = Nilai prediksi probabilitas pengguna  $\mathbf{u}$  akan berinteraksi dengan item  $\mathbf{v}$

$\mathcal{F}$  = Fungsi untuk memprediksi nilai  $\hat{y}_{uv}$

Input  $\mathcal{F}$ :

$\boldsymbol{\theta}$  = Parameter model mengoptimalkan fungsi prediksi

$Y$  = Matriks interaksi pengguna-item

$\mathcal{G}$  = *Knowledge Graph* memberikan konteks tambahan tentang hubungan antar entitas

#### c. Graph Convolutional Network

Graph Convolutional Network (GCN) adalah jenis jaringan saraf yang dirancang untuk bekerja dengan data yang terstruktur dalam bentuk graf. Berdasarkan metode ekstraksi fitur GCN terbagi 2, GCN berdasarkan domain spektrum dan domain spasial. GCN menggunakan domain spektrum berfokus pada analisis data dalam domain frekuensi, dimana metode ini melibatkan transformasi sinyal kedalam domain spektrum, sedangkan GCN berdasarkan domain spatial mendefinisikan konvolusi grafik dengan memulai dari simpul pusat yang mengagregasi informasi dari simpul tetangganya untuk menangkap hubungan lokal antar simpul. Konsep utama GCN adalah membuat embedding node dengan memodelkan pengiriman pesan dalam struktur graf, dengan meng-*aggregasi* data dari tetangga, sehingga setiap node memperoleh embeddingnya sendiri [7].

Agar GCN dapat menangkap lebih banyak informasi dari tetangga user dan item, dibuat sebuah layer. Layer ini akan mempertimbangkan pasangan kandidat user  $\mathbf{u}$  dan item  $\mathbf{v}$ , dengan  $N(\mathbf{v})$  menunjukkan jumlah tetangga yang terhubung dengan item  $\mathbf{v}$ . GCN menggunakan fungsi  $\mathbf{g} = \mathbb{R}^d \times \mathbb{R}^d = \mathbb{R}$  (misal, produk dalam) untuk menghitung skor antara user dan relasi dilakukan:



$$\pi_r^u = g(u, r)$$

Penjelasan:

$\pi_r^u$  = Menggambarkan pentingnya hubungan  $r$  dengan pengguna  $u$

$g(u, r)$  = melakukan embedding user  $u$  dan hubungan  $r$  berdasarkan dimensinya masing-masing

Menghitung skor antara user  $u$  dan item  $v$  digunakan kombinasi linier dari tetangga  $v$ :

$$v_{N(v)}^u = \sum_{e \in N(v)} \tilde{\pi}_{r_{v,e}}^u e$$

Dimana  $e$  harus terhubung sebagai atribut sebuah entitas. Dalam himpunan, entitas dan hubungan diurutkan secara ketat sesuai urutan menjelajah, semakin tinggi probabilitas kemunculan entitas, maka semakin tinggi node entitas dalam himpunan.  $\tilde{\pi}_{r_{v,e}}^u$  adalah skor hubungan-pengguna yang telah dinormalisasikan:

$$\tilde{\pi}_{r_{v,e}}^u = \frac{\exp(\pi_{r_{v,e}}^u)}{\sum_{e \in N(v)} \exp(\pi_{r_{v,e}}^u)}$$

Skor hubungan-pengguna ini akan bertindak sebagai filter personalisasi saat menghitung representasi lingkungan entitas.

Pada Knowledge graph, jumlah nodes  $N(e)$  dapat bervariasi secara signifikan, sehingga untuk menjaga komputasi lebih efisien dan membantu model dalam meningkatkan akurasi maka dilakukan pengambilan sample tetangga menggunakan *importance sampling*. *Importance sampling* (IS) merupakan teknik pengambilan sampel node yang memiliki bobot lebih, pada metode ini dilakukan sampling terhadap  $n$  nodes yang dianggap penting berdasarkan distribusi  $q$  sehingga informasi yang terkumpul dapat memenuhi ekspektasi  $E(\mu^q) = \mu$ . Dimana nodes yang memiliki informasi penting diambil berdasarkan edges  $p(v)$  dan informasi yang dipropagasi  $f(v)$ . Sehingga estimasi  $\mu$  dinyatakan sebagai [8]:

$$\hat{\mu}_q = \frac{1}{n} \sum_j \frac{p(\hat{v}_j)}{q(\hat{v}_j)} f(\hat{v}_j)$$

Penjelasan:

$n$  = Jumlah tetangga yang diambil sampelnya

$p(\hat{v}_j)$  = bobot dari node  $\hat{v}_j$

$f(\hat{v}_j)$  = informasi yang terkait dengan node  $\hat{v}_j$

$\hat{v}_j$  = node tetangga yang diambil dari distribusi  $q$

Langkah terakhir pada GCN layer adalah melakukan agregasi representasi entitas  $v$  dan representasi tetangganya  $v_{N(v)}^u$  menjadi satu vektor [9]

$$agg_{sum} = \sigma(W \cdot (v + v_{S(v)}^u) + b)$$

Penjelasan:

$W$  = Transformasi bobot

$b$  = Transformasi bias

$\sigma$  = Nonlinear function

Tujuan fungsi *agregasi* ini dirancang untuk mengumpulkan dan menggabungkan informasi dari simpul tetangga sehingga memungkinkan model untuk memahami pola dan ketergantungan lokal dalam graf [10].

#### d. Hyperparameter Tuning

Hyperparameter tuning adalah proses dimana parameter model dilakukan penyesuaian atau disetel sedemikian rupa untuk meningkatkan kinerja model. Pada KGCN beberapa parameter yang disesuaikan diantaranya:

<i>learning rate</i>	Mengontrol seberapa besar model menyesuaikan bobot selama proses pelatihan
<i>batch size</i>	Jumlah sampel yang diproses dalam satu iterasi pelatihan
<i>number of layer</i>	Jumlah lapisan konvolusional dalam model

<i>optimizer</i>	Algoritma yang digunakan untuk memperbarui bobot model
------------------	--

Metode hyperparameter yang akan digunakan adalah *GridSearchCV* dimana pemilihan parameter akan dipilih secara acak[11].

e. Evaluasi

Untuk mengevaluasi kinerja, metrik Accuracy, Precision, Recall, dan F1 Score digunakan. Untuk memahami metrik ini, terlebih dahulu harus memahami istilah true positive, false positive (FP), false negative (FN), dan true negative (TN). TP menunjukkan data positif yang diharapkan positif, sementara TN menunjukkan data negatif yang diharapkan negatif. Sebaliknya, FN menunjukkan data positif yang diharapkan negatif, dan FP menunjukkan data negatif yang diharapkan positif. Rumus perhitungan untuk Accuracy, Precision, Recall, dan F1 Score tersedia di bawah ini:

a) Accuracy

$$\text{Accuracy} = \left( \frac{TP+TN}{TP+FP+TN+FN} \right) \times 100\%$$

b) Precision

$$\text{Precision} = \left( \frac{TP}{TP+FN} \right) \times 100\%$$

c) Recall

$$\text{Recall} = \left( \frac{TP}{TP+FN} \right) \times 100\%$$

d) F1 Score

$$\text{F1 Score} = \left( \frac{2 \times \text{PRECISION} \times \text{RECALL}}{\text{PRECISION} + \text{RECALL}} \right) \times 100\% [12]$$

f. Mean Absolute Error

Mean absolute error (MAE) merupakan standard metrics yang digunakan untuk evaluasi model, digambarkan

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

Penjelasan:

$y_i$  = jumlah n sample yang di terambil

$\hat{y}_i$  = Jumlah n yang diprediksi sesuai[13]

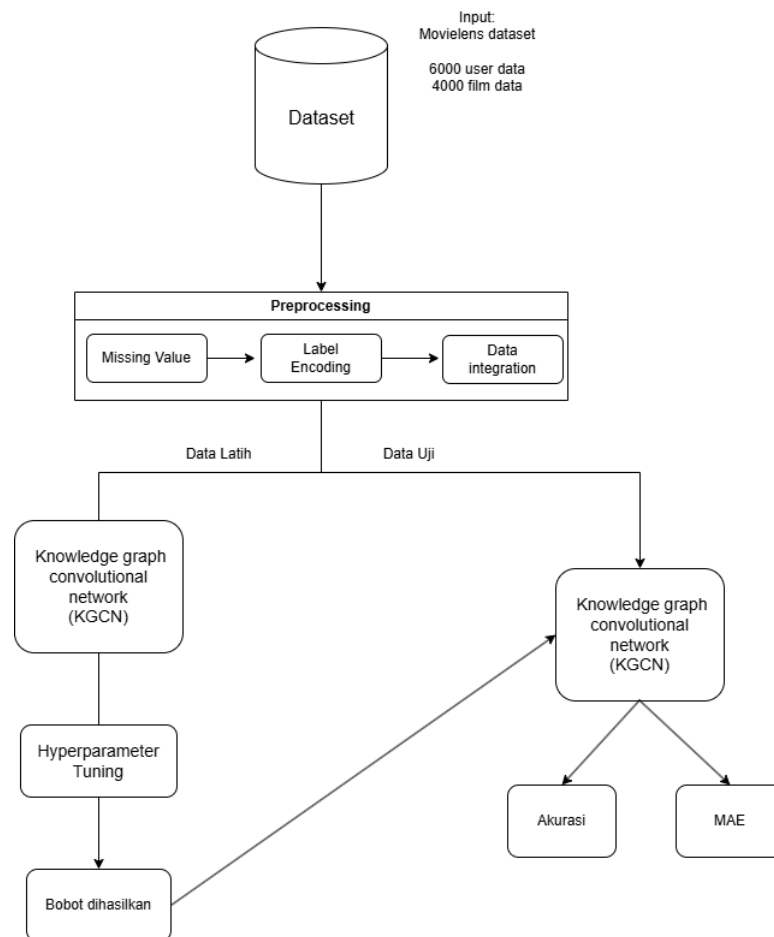
Penelitian terdahulu [2] berfokus pada pengembangan metode sistem rekomendasi yang memanfaatkan *Knowledge Graph* (KG) melalui pendekatan yang disebut *Knowledge Graph Convolutional Networks* (KGCN). Pada penelitian ini mengalami permasalahan seperti dimensionalitas tinggi dan heterogenitas KG yang membuatnya sulit untuk diolah secara langsung dalam model pembelajaran mesin, untuk mengatasi masalah tersebut digunakan metode agregasi penjumlahan informasi dari tetangga dengan cara menjumlahkan informasi dari 2 entitas yang terhubung dan menggabungkannya dengan bias yang relevan sehingga menangkap baik informasi struktural maupun semantik KG. Metode ini menghasilkan AUC 0.978 pada dataset MovieLens.

Pada penelitian [14], pengaruh knowledge graph memberikan hasil AUC 0.982 dijelaskan melalui penerapan metode CUIKG (Combining User-end and Item-end Knowledge Graph learning). Metode ini secara bersamaan mempelajari embedding pengguna dan item dari knowledge graph di kedua sisi, yang memungkinkan model untuk menangkap relevansi antara pengguna dan item dengan lebih baik. Dengan memanfaatkan informasi semantik yang kaya dari knowledge graph, model ini dapat mengatasi masalah sparsity data dengan memperkenalkan lebih banyak hubungan semantik yang mendukung akurasi rekomendasi. Selain itu, model ini memperhitungkan atribut pengguna dan bagaimana atribut tersebut mempengaruhi preferensi mereka terhadap item, sehingga menciptakan representasi yang lebih akurat. Pengelolaan informasi tetangga juga dilakukan untuk menghindari agregasi informasi yang tidak perlu, yang dapat mengurangi noise dan menjaga kualitas representasi pengguna dan item.

Pada penelitian [15] berfokus pada penerapan *importance sampling* sebagai teknik sampling yang diimplementasikan pada model GCN. Metode ini membantu model FastGCN dalam mempercepat proses pelatihan dan juga meningkatkan akurasi dalam prediksi, dalam hasil perbandingan waktu pelatihan antara GCN menunjukkan bahwa FastGCN pada dataset Cora menghasilkan 0.0084 detik, Pudmed menghasilkan 0.0047 detik, dan reddit 0.0129 detik, sedangkan pada model GCN biasa yang menggunakan dataset cora menghasilkan 0.0166 detik.

Pudmed 0.0815 detik, dan pada dataset reddit menghasilkan 2.1731 detik. Dari hasil tersebut menunjukkan bahwa model FastGCN lebih cepat dalam waktu komputasi ketika masa pelatihan dibandingkan dengan model GCN tradisional.

## 7. Metode Penelitian



Gambar 1 Metode Penelitian

### 7.1. Dataset

Pada penelitian ini, terdapat 3 file dataset diperoleh dari website grouplens yang berisi rating film yang telah diberikan oleh user. Dataset ini terdiri dari 6000 users dan 4000 film, dimana pada file rating.dat berisikan UserID, MovieID, Rating, Timestamp kemudian file users.dat berisikan UserID, Gender, Age, Occupation, Zip-Code, sedangkan file movies.dat berisikan MovieID, Title, Genres.

No	Atribut	Deskripsi
1	UserID	ID yang mewakili user
2	Gender	Merupakan atribut user yang menjelaskan jenis kelamin user
3	Age	Merupakan atribut user yang berisikan kategori umur user
4	Occupation	Merupakan atribut user yang berisikan kategori pekerjaan user
5	Zip-Code	Merupakan atribut user yang berisikan lokasi user menonton
6	MovieID	ID yang mewakili film tertentu
7	Title	Merupakan Atribut judul film
8	Genres	Merupakan atribut film yang mengkategorikan genre film
9	Rating	Merupakan atribut yang menggambarkan rating film yang diberikan user
10	Timestamp	Merupakan atribut yang menggambarkan kapan user memberikan rating film tersebut

## 7.2. Preprocessing Data

Preprocessing data merupakan tahap untuk membersihkan data agar data tidak mengurangi akurasi model sehingga data mentah bisa dipakai saat dianalisis, dilakukan beberapa tahap preprocessing:

### 7.1.1. Missing Values

Dataset akan dilakukan pengecekan missing values terlebih dahulu di semua atribut, karena jika ditemukan missing values akan mempengaruhi kemampuan model dalam memahami struktur graph.

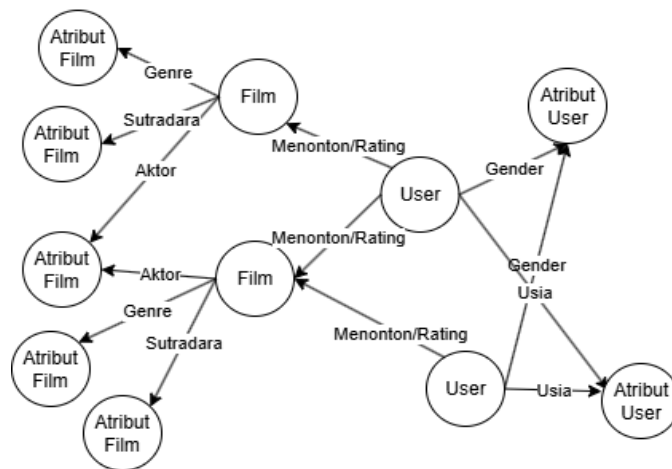
### 7.1.2. Label Encoding

Dataset juga akan dilakukan transformasi pada atribut berbentuk kategorikal menggunakan label encoding agar meningkatkan kinerja model dalam belajar pola

### 7.1.3. Data Integration

Dataset akan digabung menjadi satu berdasarkan nomor MovieID, dan userID agar didapatkan informasi yang lebih jelas mengenai bagaimana user berinteraksi dengan film tersebut.

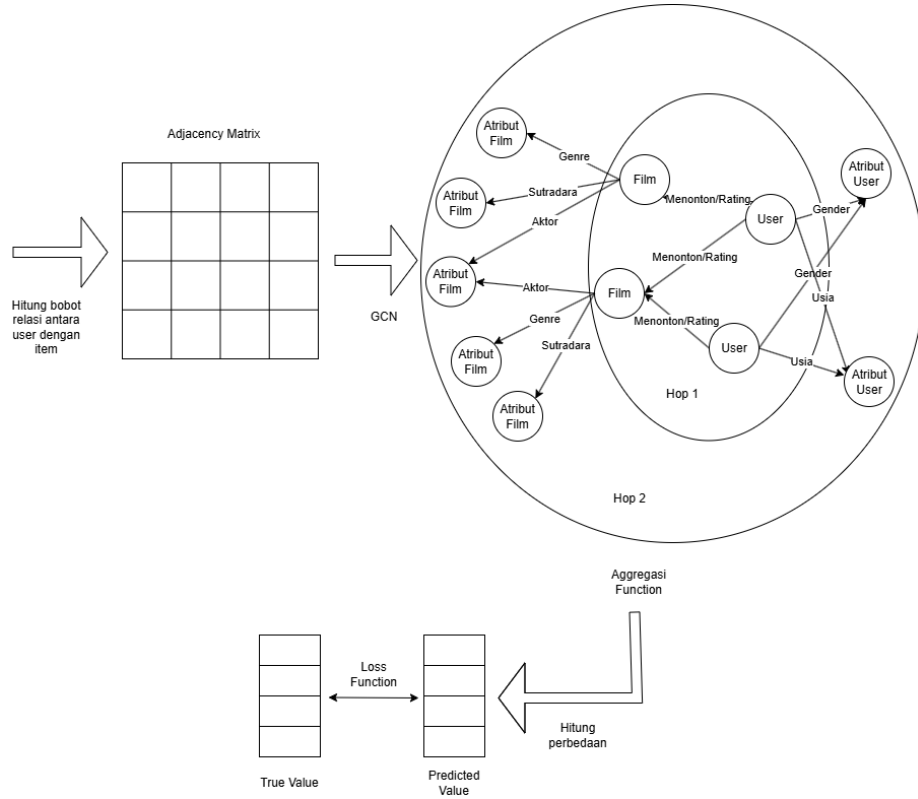
## 7.2. Knowledge Graph



Gambar 2 Arsitektur Knowledge Graph

Dari dataset yang digabung dimiliki kumpulan pengguna  $U$  dan kumpulan film  $V$ , kemudian dibuat matrix hubungan user-film dengan  $Y \in \mathbb{R}^{M \times N}$ , dimana matrix hubungan user-film  $y_{u,v} = 1$  jika user telah menonton film dan  $y_{u,v} = 0$  jika user belum pernah menonton film tersebut. Kemudian dibuat knowledge graph dengan aturan triple entitas-relasi-entitas  $(h, r, t)$  dimana  $h$ ,  $r$ ,  $t$  menunjukkan kepala, relasi, dan ekor.

### 7.3. Graph Convolutional Network



Gambar 3 Arsitektur Graph Convolutional Network

Input yang diterima GCN berupa struktur graph yang telah dibuat sebelumnya, kemudian hitung bobot relasi  $\pi_r^u = g(u, r)$  untuk mendapatkan nilai *matrix adjacency* dimana  $g: \mathbb{R}^u \times \mathbb{R}^r = \mathbb{R}$  digunakan untuk menghitung produk dalam vector  $u$  dan vector  $r$ , dari matrix ini maka akan didapatkan pola vektor pengguna  $u$  sehingga menemukan vektor hubungan dengan tetangga  $r_i$  berdasarkan hubungan yang terhubung dengan pola pada knowledge graph. Kemudian digunakan normalization softmax pada nilai penting  $\pi_r^u$  yang merupakan skor hubungan  $r$  pada pengguna  $u$ .

Pada fungsi softmax terdapat  $N(v)$  yang menunjukkan jumlah entitas yang terhubung langsung ke entitas  $v$ , kemudian digunakan importance sampling untuk memilih subset tetangga yang paling relevan berdasarkan distribusi bobot. Kemudian dilakukan agregasi informasi menggunakan sum agregasi  $v_{N(v)}^u =$



$\sigma(W \cdot (v + v_{S(v)}^u) + b$ . Kemudian dilakukan perhitungan kombinasi linear  $v_{N(v)}^u$  pada tetangga  $v$  untuk mendapatkan nilai prediksi  $\hat{y}_{u,v}$ .

## 8. Jadwal Penelitian

No	Jadwal Kegiatan	Bulan 1				Bulan 2				Bulan 3				Bulan 4			
	Minggu ke-	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
1.	Pengambilan Dataset																
2.	Preprocessing Data																
3.	Pembangunan dan pelatihan model																
4.	Pengujian																
5.	Evaluasi																
6.	Pembuatan Laporan																

## 9. Referensi

- [1] Q. Guo *et al.*, “A Survey on Knowledge Graph-Based Recommender Systems,” *IEEE Trans Knowl Data Eng*, vol. 34, no. 8, pp. 3549–3568, Aug. 2022, doi: 10.1109/TKDE.2020.3028705.
- [2] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, “Knowledge graph convolutional networks for recommender systems,” in *The Web Conference 2019 - Proceedings of the World Wide Web Conference, WWW 2019*,

- Association for Computing Machinery, Inc, May 2019, pp. 3307–3313. doi: 10.1145/3308558.3313417.
- [3] Y. Afoudi, M. Lazaar, and M. Al Achhab, “Hybrid recommendation system combined content-based filtering and collaborative prediction using artificial neural network,” *Simul Model Pract Theory*, vol. 113, Dec. 2021, doi: 10.1016/j.simpat.2021.102375.
  - [4] J. A. P. Sacenti, R. Fileto, and R. Willrich, “Knowledge graph summarization impacts on movie recommendations,” *J Intell Inf Syst*, vol. 58, no. 1, pp. 43–66, Feb. 2022, doi: 10.1007/s10844-021-00650-z.
  - [5] X. Wang *et al.*, “Learning intents behind interactions with knowledge graph for recommendation,” in *The Web Conference 2021 - Proceedings of the World Wide Web Conference, WWW 2021*, Association for Computing Machinery, Inc, Apr. 2021, pp. 878–887. doi: 10.1145/3442381.3450133.
  - [6] C. Li, Y. Cao, Y. Zhu, D. Cheng, C. Li, and Y. Morimoto, “Ripple Knowledge Graph Convolutional Networks for Recommendation Systems,” *Machine Intelligence Research*, vol. 21, no. 3, pp. 481–494, Jun. 2024, doi: 10.1007/s11633-023-1440-x.
  - [7] H. Xia, K. Huang, and Y. Liu, “Unexpected interest recommender system with graph neural network,” *Complex and Intelligent Systems*, vol. 9, no. 4, pp. 3819–3833, Aug. 2023, doi: 10.1007/s40747-022-00849-9.
  - [8] Y. Ji *et al.*, “Accelerating Large-Scale Heterogeneous Interaction Graph Embedding Learning via Importance Sampling,” *ACM Trans Knowl Discov Data*, vol. 15, no. 1, Jan. 2021, doi: 10.1145/3418684.
  - [9] Y. Wang, X. F. Ma, and M. Zhu, “A knowledge graph algorithm enabled deep recommendation system,” *PeerJ Comput Sci*, vol. 10, 2024, doi: 10.7717/PEERJ-CS.2010.
  - [10] U. A. Bhatti, H. Tang, G. Wu, S. Marjan, and A. Hussain, “Deep Learning with Graph Convolutional Networks: An Overview and Latest Applications in Computational Intelligence,” 2023, *Wiley-Hindawi*. doi: 10.1155/2023/8342104.

- [11] S. Nur Himawa and R. Sohiburoyyan, “Hyperparameter Tuning on Graph Neural Network for the Classification of SARS-CoV-2 Inhibitors,” 2023. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/JAIC>
- [12] S. Sathyanarayanan, “Confusion Matrix-Based Performance Evaluation Metrics,” *African Journal of Biomedical Research*, pp. 4023–4031, Nov. 2024, doi: 10.53555/AJBR.v27i4S.4345.
- [13] T. O. Hodson, “Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not,” Jul. 19, 2022, *Copernicus GmbH*. doi: 10.5194/gmd-15-5481-2022.
- [14] T. Gu, H. Liang, C. Bin, and L. Chang, “Combining user-end and item-end knowledge graph learning for personalized recommendation,” *Journal of Intelligent and Fuzzy Systems*, vol. 40, no. 5, pp. 9213–9225, 2021, doi: 10.3233/JIFS-201635.
- [15] J. Chen, T. Ma, and C. Xiao, “FastGCN: Fast Learning with Graph Convolutional Networks via Importance Sampling,” Jan. 2018, [Online]. Available: <http://arxiv.org/abs/1801.10247>