**IMPLEMENTASI SISTEM REKOMENDASI TEKS PADA JUDUL FILM ANIME DENGAN METODE ATTENTION BASED NEURAL COLLABORATIVE FILTERING**

****

**Disusun oleh :**

**MUHAMMAD RIZAL**

**123170036**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK INDUSTRI**

**UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN” YOGYAKARTA**

**2022**

# DAFTAR ISI

[DAFTAR ISI ii](#_Toc130904285)

[BAB I Pendahuluan 1](#_Toc130904286)

[1.1 Latar Belakang 1](#_Toc130904287)

[1.2 Rumusan Masalah 2](#_Toc130904288)

[1.3 Batasan Masalah 3](#_Toc130904289)

[1.4 Tujuan Penelitian 3](#_Toc130904290)

[1.5 Manfaat Penelitian 3](#_Toc130904291)

[1.6 Tahapan Penelitian 3](#_Toc130904292)

[1.6.1 Rencana dan Tahapan Penelitian 3](#_Toc130904293)

[1.6.2 Metode Pengembangan Sistem 4](#_Toc130904294)

[1.6.3 Pengujian Sistem 5](#_Toc130904295)

[1.6.4 Pengujian Penelitian 5](#_Toc130904296)

[1.7 Sistematika Penulisan 5](#_Toc130904297)

[BAB II Tinjauan Pustaka 7](#_Toc130904298)

[2.1 Anime 7](#_Toc130904299)

[2.2 Sistem Rekomendasi 7](#_Toc130904300)

[2.2.1 Content Based Filtering 7](#_Toc130904301)

[2.2.2 Collaborative Filtering 8](#_Toc130904302)

[2.2.3 Hybrid Filtering 10](#_Toc130904303)

[2.3 Deep Learning 10](#_Toc130904304)

[2.4 Matrix Factorization 11](#_Toc130904305)

[2.5 Neural Collaborative Filtering 11](#_Toc130904306)

[2.6 Recurrent Neural Network 15](#_Toc130904307)

[2.7 Long-Short Term Memory 19](#_Toc130904308)

[2.7 Bidirectional Long-Short Term Memory 23](#_Toc130904309)

[2.8 Pre training 26](#_Toc130904310)

[2.9 Normalized Discounted Cumulative Gain 27](#_Toc130904311)

[2.11 Studi Pustaka (*State of the Art)* 28](#_Toc130904312)

[BAB III Metodologi Penelitian 31](#_Toc130904313)

[3.1 Metodologi Penelitian 31](#_Toc130904314)

[3.2 Studi Literatur 31](#_Toc130904315)

[3.3 Pengumpulan Data 32](#_Toc130904316)

[3.4 Pre-processing 32](#_Toc130904317)

[3.4.1 Memodifikasi kolom dataset 33](#_Toc130904318)

[3.4.2 Data Filtering 33](#_Toc130904319)

[3.4.3 Data Cleaning 34](#_Toc130904320)

[3.4.4 Data Splitting 35](#_Toc130904322)

[3.5 Pembuatan Model 35](#_Toc130904323)

[3.5.1 Matrix Factorization 36](#_Toc130904324)

[3.5.2 Multi-Layer Perceptron 36](#_Toc130904325)

[3.5.3 Neural Collaborative Filtering 36](#_Toc130904326)

[3.6 Pengujian Sistem 36](#_Toc130904327)

[3.7 Metodologi Pengembangan Sistem 36](#_Toc130904328)

[3.7.1 Analisis Kebutuhan Sistem 36](#_Toc130904329)

[3.7.2 Kebutuhan Fungsional 36](#_Toc130904330)

[3.7.3 Kebutuhan Non-Fungsional 36](#_Toc130904331)

[3.7.4 Proses Desain 36](#_Toc130904332)

[3.7.5 Perancangan Sistem 36](#_Toc130904333)

[3.7.8 Perancangan Pengujian 36](#_Toc130904334)

[Daftar Pustaka 36](#_Toc130904335)

[Lampiran 39](#_Toc130904336)

# BAB I Pendahuluan

## 1.1 Latar Belakang

Berkembangnya variasi atas kebutuhan hidup manusia di masa sekarang seakan terus berkembang seiring dengan pertumbuhan teknologi informasi serta telekomunikasi. Termasuk bentuk hiburan yang merupakan salah satu bentuk kebutuhan yang tak lepas dari kehidupan manusia, salah satu bentuk dari hiburan tersebut adalah film (Billah, M et al., 2021). Film yang merupakan kombinasi dari audio serta visual juga terdiri dari berbagai jenis seperti Movie, TV, Dokumentasi dan sebagainya. Animasi merupakan salah satu bentuk bagaimana film ditampilkan yang merupakan kumpulan dari frame yang digambar menggunakan tangan yang kemudian diolah komputer menjadi animasi, animasi atau anime merupakan salah satu teknologi perfilman yang telah berkembang lama di jepang (Soni, B et al., 2021).

Dengan perkembangan yang luar biasa ini pastinya sangat sulit bagi user untuk menentukan preferensi mereka (Nuurshadieq & Wibowo., 2020). Dengan sistem rekomendasi permasalahn tersebut bisa dikurangi, umumnya sistem rekomendasi umumnya memiliki tiga teknik yaitu *collaborative filtering*, *content-based filtering,* dan *hybrid filtering.* Salah dua dari teknik tersebut bisa dikombinasikan sehingga mampu mengatasi kekurangan satu sama lain yaitu *collaborative filtering* dengan *content based filtering*, teknik tersebut disebut dengan *hybrid filtering*. (Lops, P et al., 2011)

Beberapa penelitian sebelumnya yang pernah meneliti sistem rekomendasi anime yang menggunakan *side information* dan informasi user yaitu Nuurshadieq & Wibowo., 2020 menerapkan *collaborative filtering* menggunakan LSTM yang bertujuan mengatasi cold-start yang menghasilkan RMSE sebesar 1.4475 yang menunjukkan penelitian tersebut telah lebih baik dibandingkan metode populer seperti SVD dan KNN, penelitian ini membuktikan bahwa dengan mengekstrasi informasi berbasis teks akan mampu memberikan peningkatan performa.

Penelitian lainnya juga dicoba oleh Billah, M et al., 2021 yang menerapkan sistem rekomendasi anime berbasis *collaborative filtering* menggunakan PCA dan K-Means yang menghasilkan kompleksitas waktu sebesar 2,999602 serta menghasilkan nilai akurasi MMR (Mean Reciprocal Rank) sebesar 0.5619. Penelitian lainnya berhubungan dengan rekomenasi anime yang dilakukan oleh Soni, B et al., 2021 menerapkan sistem rekomendasi *hybrid recommendation filtering* yang menggunakan algoritma *autoencoder* dan *clustering spectral* yang menggunakan dataset *MyAnimeList* yang menghasilkan RMSE sebesar 0.591 dan 0.349,

Pada penelitian yang diteliti oleh Erritali et al.,(2021) menggunakan algoritma KNN dan SVD pada penelitian tersebut menggunakan dataset *movielens* menerapkan pendekatan *hybrid collaborative* filtering, menghasilkan bahwa model SVD berhasil melampaui model KNN, selisih SVD dengan KNN di RMSE sebesar 3.95%, MAE sebesar 3.99%, SVD juga memiliki precision lebih tinggi sebesar 3.94% dan accuracy lebih baik sebesar 5.69%. Kombinasi kedua metode tersebut menghasilkan hanya 0.245% lebih baik dibanding model SVD. sedangkan untuk evaluasi sisanya seperti MAE masih lebih baik SVD sebesar 0.256%, precision sebesar 0.126% dan recall 0.7% lebih baik dibanding kombinasi model SVD dan KNN. Penelitian tersebut menyimpulkan bahwa model KNN tidak begitu berkontribusi besar ketika dikombinasikan dengan SVD bahkan model SVD sendiri sudah cukup lebih baik dibandingkan dengan kombinasi kedua model tersebut.

Penelitian yang berkaitan dengan sistem rekomendasi juga telah diterapkan oleh peneliti lain pada rekomendasi movie menggunakan metode LSTM dan CNN oleh Wentao et al (2020) yang menghasilkan MSE sebesar 0,876 dan MAE 0,751. Penelitian yang dilakukan oleh Haili et al., 2020 juga sama menggunakan metode LSTM-CNN dengan menggunakan dataset movieLens menerapkan personalisasi movie rekomendasi yang menghasilkan MAE sebesar 0,7224 dan MSE 0,691739 dari penelitian tersebut menghasilkan nilai yang lebih baik dibandingkan dengan hanya menggunakan metode CNN.

Penelitian yang menerapkan metode *attention* adalah penelitian SASRec (*Self-Attentive Sequential Recommendation*) yang diteliti oleh Kang, W. C., & McAuley, J., 2018. Penelitian tersebut menggunakan berbagai dataset berbasis teks yaitu dataset Amazon, MovieLens, dan Steam dari evaluasi ketiga dataset penelitian tersebut berhasil memberikan performa yang lebih baik dibanding metode lain dengan berhasil melakukan peningkatan rata - rata sebesar 6.9% Hit Rate dan NDCG mencapai 9.6% , dengan menggunakan *attention* berhasil menangani *long sequence*.

Berdasarkan penjabaran diatas maka penelitian ini akan menggunakan pendekatan *hybrid filtering* karenadengan menggunakan hybrid filtering kelemahan pada sistem rekomendasi akan teratasi. Seperti pada permasalahan *cold-start* bisa diatasi dengan penggunaan *content-based filtering* (Wang, H et al., 2020). Penelitian ini juga akan menambah eksplorasi penelitian *deep learning*, dengan menggunakan *deep learning* model rekomendasi bisa merepresentasikan relasi antar user dan item dengan mempelajari pada *deep-level* jaringan struktur non-linear (Wang, W et al., 2020). Penggunaan *Attention* *Mechanism* juga membantu model dalam meningkatkan performa (Wu, F., et al., 2018). Metode yang akan digunakan pada penelitian ini adalah Neural Collaborative Filtering.

Data yang akan digunakan bersumber dari dataset yang ada yang berasal dari situs *kaggle*. Dengan diterapkan penelitian ini diharapkan dapat menghasilkan sistem rekomendasi yang mampu menghasilkan rekomendasi anime yang lebih baik dari sistem yang telah ada. Sistem yang mampu merekomendasikan sesuai prefensi dari user tersebut serta mampu memperluas pengalaman user dalam menggunakan sistem rekomendasi ini.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dijelaskan sebelumnya maka didapatkan rumusan masalah adalah menerapkan sistem rekomendasi pada text judul anime menggunakan metode Neural Collbaorative Filtering dengan Attention Mechanism

## 1.3 Batasan Masalah

Penelitian ini memiliki batasan dalam penyelesaiannya, sebagai berikut :

1. Menggunakan data penelitian yang bersumber dari *myanimelist* (kaggle).
2. Data penelitian berbahasa inggris dan berformat .csv
3. Sistem rekomendasi pada penelitian ini menggunakan data masukan yang terbatas pada anime\_id, user\_id, genre, episode dan rating user
4. Pada penelitian ini terbatas pada tipe anime yang disiarkan melalui media TV
5. Pada penelitian data keluaran yang dihasilkan hanya judul anime dan score

## 1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan, sebagai berikut :

1. Mengetahui tingkat akurasi penerapan metode Neural Collaborative Filtering terhadap hasil rekomendasi yang diberikan
2. Mengetahui cara kerja pada penerapan metode Neural Collbaorative Filtering dengan attention mechanism pada sistem rekomendasi

## 1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah memberikan hasil rekomendasi terbaik kepada orang - orang penonton aktif film anime maupun orang - orang yang baru mengenal film anime, selain itu penelitian juga bertujuan untuk meningkatkan eksplorasi metode *deep learning* khususnya pada penerapan metode *Neural Collaborative Filtering* dengan *attention mechanism* pada sistem rekomendasi.

## 1.6 Tahapan Penelitian

Tahapan penelitian yang diterapkan pada penelitian ini adalah sebagai berikut :

### 1.6.1 Rencana dan Tahapan Penelitian

a. Studi Literatur

Pada penelitian ini permasalahan dan penyelesaian yang diselesaikan dihimpun dari berbagai referensi sumber literatur yang relevan dan sesuai dengan penelitian ini.

b. Pengumpulan dan Pengolahan Data

Pada tahapan ini data dikumpulkan berasal dari situs *kaggle* yang kemudian diolah dan dipilah yang mana saja yang akan dijadikan data penelitian.

c. Analisis Sistem

Analisis sistem ini dilakukan untuk menganalisa berbagai keperluan dalam proses perancangan sistem sehingga memudahkan dalam proses selanjutnya.

d. Perancangan Sistem

Pada tahapan ini merupakan penerapan dari analisis sistem sebelumnya yaitu menerapkan pemodelan terhadap sistem yang akan dibuat pada penelitian ini.

e. Implementasi Perangkat Lunak

Tahapan ini merupakan tahapan mengimplementasikan sistem yang sebelumnya sudah dirancang.

f. Pengujian dan Analisis

Pada tahapan ini dilakukan pengujian terhadap sistem yang telah diimplemtasikan sebelumnya, yang kemudian akan dilakukan analisis terhadap hasil pengujian yang telah dilakukan.

g. Kesimpulan dan Saran

Pada tahapan ini penelitian akan diberikan kesimpulan dari hasil pengujian yang telah dilakukan dan kemudian akan disertakan saran yang selanjutnya dapat dikembangkan untuk penelitian selanjutnya sehingga mendapatkan hasil penelitian yang lebih baik.

### 1.6.2 Metode Pengembangan Sistem

Metode pengembangan sistem yang akan digunakan pada penelitian ini adalah *prototyping.* *Prototyping* atau *prototype* digunakan dengan alasan karena dengan menggunakan metode ini kedekatan antara perancang dan pengguna. Dengan proses yang dilakukan secara terstruktur pada setiap tahapan pembuatannya membuat sistem lebih cepat dan lebih hemat dibandingkan metode pengembangan sistem lainnya. Tahapan pada proses *prototyping* (Pressman, 2015) adalah sebagai berikut :

1. *Communication*

Pada tahapan awal pengembangan sistem melakukan komunikasi dan kolaborasi antara pengguna atau pemangku kepentingan dengan maksud memahami tujuan dan kebutuhan sistem sehingga bisa mempermudah dalam menentukan fitur dan fungsi sistem yang akan dibangun.

2. *Planning*

Pada tahapan ini dilakukan perencanaan yang berfungsi untuk mempermudah pengembang saat proses pembuatan sistem. Tahap planning mendeskripsikan tugas teknis, resiko yang mungkin akan terjadi, kebutuhan sumber daya, hasil produk, dan jadwal pengerjaan sistem.

3. *Model*

Pada tahapan ini pengembang membuat model dari sistem yang akan dibuat sehingga pengembang dapat memahami kebutuhan sistem dan desain yang sesuai untuk menunjang kebutuhan tersebut.

4. *Construction*

Pada tahap construction, pengembang memulai pembuatan sistem tahap sebelumnya, selain itu tahap ini juga melakukan pengujian atau testing untuk menemukan kesalahan pada pembuatan sistem.

5. *Deployment*

Pada tahap ini sistem yang telah dibuat dikirimkan kepada pengguna baik semua fitur selesai maupun sebagian untuk mendapatkan evaluasi produk dan memberikan feed back berdasarkan evaluasi.

### 1.6.3 Pengujian Sistem

Pengujian sistem yang akan digunakan pada penelitian ini adalah metode black box testing, yaitu metode yang pengujian sistem yang menekankan fungsionalitas tanpa mengetahui coding dari sistem tersebut. Black box testing bertujuan untuk mengukur kinerja dari sistem yang telah dibangun.

### 1.6.4 Pengujian Penelitian

Pengujian penelitian biasa digunakan pada penelitian rekomendasi sistem adalah NDCG (Normalized Discounted Cumulative Gain) dan RMSE (Root Mean Square Error). Kedua metode tersebut merupakan metode yang paling sering digunakan pada penerapan sistem rekomendasi. NDCG mampu memperhitungkan relevansi dan posisi dari sebuah item sehingga mampu memperhitungkan keakuratan lebih tepat (Wang, Y., et al., 2013), sedangkan RMSE merupakan metode yang mampu memperhitungkan seberapa jauh nilai prediksi dengan nilai kebenarannya sehingga peneliti mampu mempertimbangkan kinerja model yang dibuat pada model yang dibuat sehingga bisa diidentifikasi yang kemudian bisa dilakukan perbaikan pada model tersebut sehingga bisa membantu meningkatkan kinerja model tersebut.

## 1.7 Sistematika Penulisan

Sistematika penulisan yang digunakan dalam menyusun laporan penelitian ini adalah sebagai

berikut:

**Bab I Pendahuluan**

Pada bagian ini membahas tentang latar belakang masalah, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, metodologi penelitian, dan sistematika penulisan.

**Bab II Tinjauan Literatur**

Tinjauan literatur memuat tentang dasar teori yang sudah ada sebagai bahan referensi terkini dan pondasi untuk memperkuat argumentasi dalam penelitian ini sekaligus mendasari pemecahan masalah dalam penelitian ini.

**Bab III Metodologi Penelitian dan Pengembangan Sistem**

Bab ini membahas tentang tahap perancangan kebutuhan, tahap analisis, dan tahap perancangan serta meberikan gambaran garis besar penyusunan program.

**Bab IV Hasil, Pengujian dan Pembahasan**

Pada bab ini akan menyajikan hasil penelitian berisi hasil implementasi dari perancangan yang telah dibuat pada bab sebelumnya dan berisi pengujian terhadap hasil penelitian beserta pembahasannya

**Bab V Kesimpulan dan Saran**

Pada bab ini berisi kesimpulan dari hasil penelitian dan saran yang diajukan untuk pengembangan penelitian selanjutnya.

# BAB II Tinjauan Pustaka

## 2.1 Anime

Anime merupakan kartun yang aslinya berasal dari jepang, selain itu istilah Anime dalam bahasa inggris adalah *animation* (B. Soni et al., 2021). Anime atau kartun pada dasarnya memiliki definisi yang sama yaitu kumpulan gambar / frame yang disusun sehingga membentuk animasi. Perbedaan yang membuat anime berbeda pada kartun yang sejenis adalah desain dari karakter anime yang memiliki ciri khas tersendiri seperti memiliki mata yang besar. Anime juga memiliki format penanyangan yang cukup bervariasi yaitu TV, Movie, ONA, OVA, dan OAD (Billah, M et al., 2021). Selain itu anime juga memiliki banyak kategori tidak hanya genre yang bervariasi terdapat musim tayang, sumber cerita (*manga*, *original, light novel)*, dan studio.

## 2.2 Sistem Rekomendasi

Sistem rekomendasi merupakan sistem yang bertujuan untuk menghasilkan suatu *item* kepada user sehingga user mampu meningkatkan pengalaman penggunaan terhadap aplikasi yang menggunakan sistem rekomendasi tersebut, sehingga user bisa membuat keputusan terhadap sesuatu hal yang diinginkanny,. biasanya sistem rekomendasi menghasilkan item secara spesifik seperti merekomendasikan musik ataupun berita (Ricci, F et al., 2011). Sistem rekomendasi juga bisa dilihat sebagai sistem pencarian ranking dimana query masukannya adalah kumpulan dari user dan konteks data informasi yang berhubungan (Cheng, H. T et al., 2016)

Sistem rekomendasi hadir untuk mengurangi derasnya arus informasi yang hadir di internet dengan adanya sistem rekomendasi user mampu memilih serta menentukan arus informasi yang seperti apa yang sesuai kebutuhan user tersebut. Sistem rekomendasi merupakan sistem yang terpersonalisasi yang ditujukan secara khusus kepada setiap user tergantung dari kebutuhan user itu sendiri sehingga tujuan dari sistem rekomendasi adalah meningkatkan pengalaman ditujukan kepada satu user bukan merepresentasikan suatu grup secara keseluruhan (Burke, R et al., 2011)

Dalam pendekatannya sistem rekomendasi biasanya menggunakan dua pendekatan yaitu *collaborative filtering, content based filtering* dan *knowledged based.* Dengan perkembangan informasi saat ini yang lebih kompleks maka hadirlah metode *hybrid* *filtering* yang mengkombinasikan kedua kemampuan dari metode tersebut sehingga bisa saling menutupi kekurangan satu sama lain.

### Content Based Filtering

*Content Based Filtering* merupakan salah satu pendekatan yang terdapat pada sistem rekomendasi yang lebih berfokus terhadap attribut atau *features* yang terdapat pada *item* yang akan direkomendasikan kepada user. Content based filtering merupakan metode yang dimana orang - orang yang menyukai sebuah item dengan beberapa attribut di aktifitas sebelumnya yang dimana di masa depan akan memiliki item yang rekomendasi yang mirip dengan item tersebut (Çano, E., & Morisio, M., 2017).

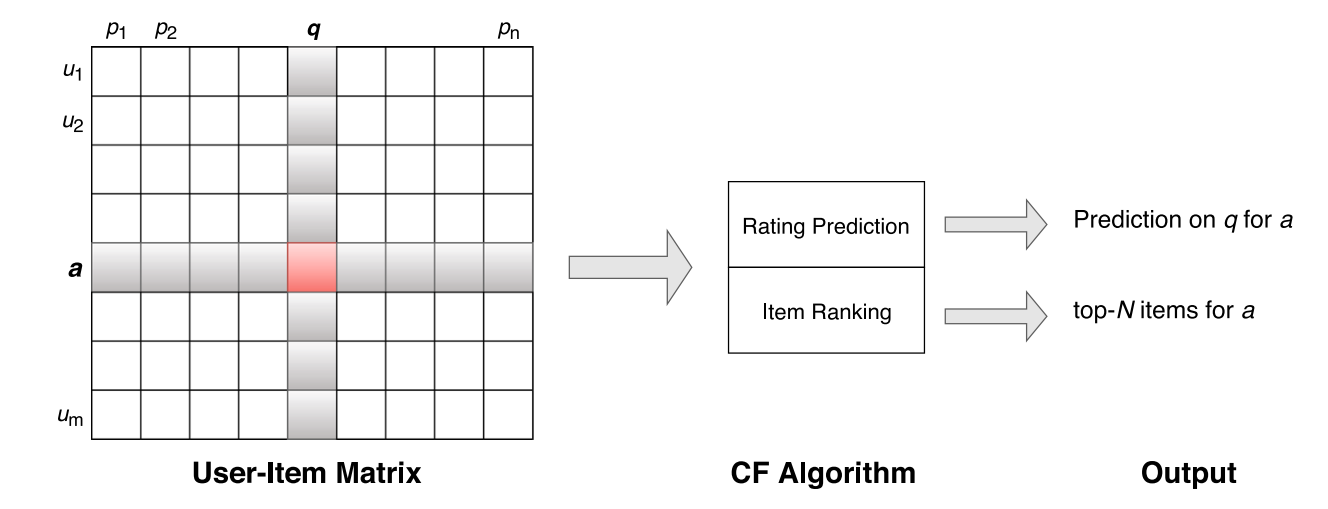
Satu kunci permasalahan dari rekomendasi content-based adalah kualitas dari feature. Item yang direkomendasikan membutuhkan deskripsi attribut yang jelas dan berarti sehingga rekomendasi item tersebut kepada user bisa dilakukan (Burke, R et al., 2011). Teknik *content-based filtering* tidak membutuhkan data *profile* orang lain karena hal tersebut tidak mempengaruhi hasil rekomendasi, selain itu jika data user profle berubah teknik content-based masih mempunyai potensi untuk menyesuaikan hasil rekomendasi dalam waktu yang ingkat (Isinkaye, F et al., 2015).

Keuntungan dengan menggunakan teknik content-based yaitu mampu memberikan rekomendasi item baru meskipun user tidak memberikan rating kepada item - item yang dituju oleh user tersebut, jadi meskipun database tidak menyediakan data user preferensi hal itu tidak mempengaruhi tingkat keakurasian sistem rekomendasi (Isinkaye, F et al., 2015). User tidak perlu untuk berbagi preferensi satu sama lain sehingga memastikan keamanan *privacy* user tersebut. *Content-based filtering* juga memberikan rekomendasi yang *transparency* artinya sistem rekomendasi yang diberikan sangat jelas dan masuk akal karena memiliki keterhubungan dengan item yang user sukai dari aktifitas user itu sendiri berbeda dengan *collaborative filtering* yang mengambil kecocokan item berdasarkan data dari user lain yang artinya *item* rekomendasi yang diberikan belum tentu memiliki kesamaan *features* dengan *item* yang sesuai prefensi user tersebut (Ricci, F et al., 2011).

Kekurangan dari menggunakan pendekatan CBF ini adalah Keterbatasan hasil rekomendasi karena terlalu bergantung kepada features yang terdapat pada item preferensi user tersebut sehingga hasil rekomendasi yang diberikan oleh sistem akan terbatas pada konten dengan tema yang sama yang kadang bisa memberikan hasil rekomendasi yang sama tanpa ada hasil rekomendasi yang baru. (Ricci, F et al., 2011). Pada teknik CBF juga sulit mendapatkan *feedback* dari user karena user terbiasa untuk tidak memberikan rating kepada hasil rekomendasi yang diberikan mengakibatkan sulit untuk mengetahui apakah rekomendasi yang diberikan merupakan hasil yang benar atau tidak. (Modallal, S., 2015)

### Collaborative Filtering

Collaborative Filtering adalah metode sistem rekomendasi yang menghasilkan hasil rekomendasi secara spesifik yang dilakukan dengan cara melakukan pendekatan berdasarkan kemiripan selera dengan pengguna lain. Collaborative filtering adalah sebuah sistem rekomendasi yang mempunyai cara kerja yang cukup berbeda dengan content-based pada teknik ini prediksi sistem rekomendasi bukan dihasilkan dari deskripsi metadata seperti deskripsi yang terdapat pada movie dan musik, pada teknik pendekatan yang digunakan adalah menggunakan preferensi item oleh user jadi user yang memiliki kemiripan preferensi akan menghasilkan prediksi sesuai kemiripan preferensi antar user. (Isinkaye, F et al., 2015) Aktor utama dari sistem rekomendasi CF adalah user yang secara aktif mencari hasil rekomendasi dari ranking items atau prediksi rating. Dengan memanfaatkan prefensi sebelumnya sebagai patokan untuk menentukan korelasi antar user, sebuah pendekatan CF mengandalkan prefensi pengguna yang sesuai. (Batmaz, Z et al., 2019) .



**Gambar 2.1** Gambaran sederhana Colllaborative Filtering (Batmaz, Z et al., 2019)

Umumnya pada collaborative filtering terdapat dua pendekatan yaitu *memoery-based* dan *model-based.*

1. Memory-based filtering

*Memory based filtering* dikenal sebagai *neighborhood fltering*, metode collaborative filtering memanfaatkan kombinasi dari *ratings user-items* yang terprediksi melalui basis kedekatan mereka (Aggarwal, C. C., 2016). Terdapat dua pendekatan yang digunakan pada *memory-based filtering* ini yaitu :

1. *User-based collaborative filtering*

Pada pendekatan *user-based* ini memanfaatkan kemiripan preferensi user yang kemudian memberikan rekomendasi item berdasarkan kemiripan preferensi user. Sebagai contoh misal user A dan User B menyukai item P maka hasil rekomendasi jika user B menyukai item Q berarti ada kemungkinan user A akan menyukai item Q juga.

1. *Item-based collaborative filtering*

Pada pendekatan *item-based* berfokus pada kemiripan selera item jadi hasil rekomendasi yang diberikan akan berdasarkan kemiripan selera item sebagai contoh misal banyak user yang jika menyukai item A maka menyukai item B maka user yang menyukai item A akan diberikan rekomendasi item B.

1. Model-based filtering

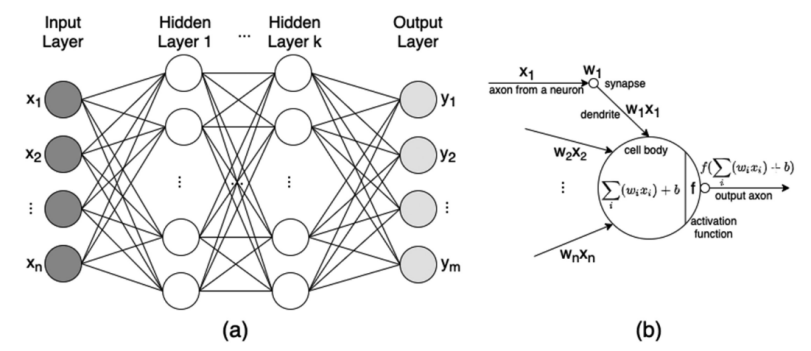
*Model-based filtering* merupakan teknik collaborative filtering yang menggunakan model machine learning atau teknik data mining. Dengan menggunakan pendekatan ini dibandingkan dengan *memory-based* atau *neighborhood-based systems* pada *model-based* pendekatan yang digunakan adalah dengan menggunakan rating untuk belajar pada model prediksi yang akan diterapkan (Ricci, F et al., 2011). Ide dasar dari model-based adalah memodelkan hubungan antara user-item dengan fitur antara pengguna dan item dalam sistem, seperti preferensi dari user dan kategori dari item. Model ini dilatih dengan menggunakan data yang tersedia yang kemudian digunakan untuk memprediksi rating user untuk item baru.

### Hybrid Filtering

Hybrid Filtering yaitu teknik sistem rekomendasi yang menggabungkan atau mengkombinasikan dua atau lebih teknik sistem rekomendasi. Variasi teknik rekomendasi yang telah ada hingga saat ini adalah *content-based, collaborative-filtering, knowledge-based* dan *demographic technique* dari tiap teknik tersebut masing - masing memiliki kekurangan sehingga untuk menanggulangi kekurangan tersebut adalah dengan menggunakan teknik *hybrid filtering*. Salah satu kelemahan yang paling umum dari teknik *content-based* dan *collaborative filtering* adalah *cold-start* (Burke, 2007) yang dimana kelemahan tersebut adalah singkatnya adalah apa hal yang direkomendasikan kepada user yang memiliki hanya sedikit rating.

### 2.3 Deep Learning

Deep learning merupakan bidang yang masih sangat digandrungi oleh komunitas *machine learning* dan *data mining*, *deep learning* itu sendiri termasuk cabang dari machine learning (Alfarhood, M., & Cheng, J. 2021). Model dari *deep learning* ini sendiri bisa dilatih dengan *supervised learning* ataupun unsupervised learning. Definisi sederhana dari *deep learning* adalah setiap jaringan syaraf dengan lebih dari dua lapisan (Suyanto et al., 2019). Berikut gambaran dasar dari deep learning :



**Gambar 2.2** **(a) Arsitektur umum dari jaringan syaraf deep learning, (b) Jaringan syaraf tiruan: Diagram Perhitungan dasar untuk jaringan syaraf. (Selvaraj, S., 2021)**

Alasan mengapa *deep learning* cocok digunakan pada bidang sistem rekomendasi menurut Alfarhood, M., & Cheng, J. 2021 adalah

1. E*nd to End Differentiable* maksudnya adalah umumnya pada metode tradisional penelitian dilaksanakan pada beberapa tahap dengan menggunakan *deep learning* hal tersebut bisa diperpendek dengan menjadikannya hanya dengan satu tahapan yaitu dengan hanya menggunakan sebuah *neural network.*
2. Memberikan *inductive bias* yang sesuai kepada tipe data masukan. *Inductive bias* memungkinkan sebuah algoritma pembelajaran untuk memprioritaskan satu solusi dibandingkan yang lain, *inductive bias* mampu memberikan asumsi tentang proses pembuatan data atau kemungkinan ruang solusi (Battaglia, P. W et al., 2018). Karena hal tersebut jika terdapat struktur yang inheren yang dapat dieksploitasi oleh model, maka dengan begitu deep neural networks bisa berguna.

### Matrix Factorization

Matrix Factorization merupakan tknik yang digunakan untuk memformulasikan data matriks menjadi dua matriks dua matriks yang lebih kecil. Pada konteks sistem rekomendasi matriks factorization adalah salah satu teknik yang terdapat pada collaborative filtering yang digunakan dalam sistem rekomendasi untuk menghasilkan rekomendasi item kepada pengguna dengan cara memprediksi nilai yang hilang pada matriks yang merepresentasikan riwayat yang terdapat pada rating pengguna terhadap item (Koren, Y et al., 2009).

Matrix Factorization melakukan pembagian matriks rating menjadi dua matriks yang lebih kecil yaitu matriks user dan matriks item, kemudian kedua matriks tersebut diberi ukuran dimensi tertentu, yang diberi nama rank atau faktor. Rank ini merupakan representasi fitur yang terdapat pada matriks user dan item tersebut. Dengan tujuan untuk memprediksi rating yang belum diketahui pengguna, bisa dengan melakukan pengalian dua matriks tersebut. Secara sederhana bentuk matriks factorization berdasarkan *paper* (He, X et al.,2017) adalah sebagai berikut :

Text

Description automatically generated

**Gambar 2.3** Formula Matriks Factorization (He, X et al.,2017)

Keterangan :

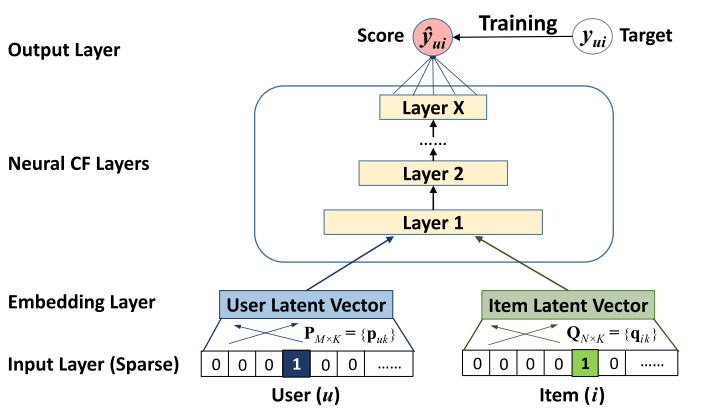
: matriks rating dimensi u x i

p : matriks user preferensi user dimensi u x k dengan k fitur

q : matriks yang merepresentasikan fitur pada item dengan dimensi i x k

### Neural Collaborative Filtering

Neural Collaborative Filtering merupakan metode sistem rekomendasi yang mengkombinasikan dua pendekatan yaitu collaborative filtering dan neural network (He, X et al.,2017). Metode NCF (Neural Collaborative Filtering) merupakan metode yang terdiri dua komponen utama yaitu embedding layer dan multi-layer perceptron. Embedding layer menggunakan data user dan item sebagai masukan yang kemudian diubah menjadi vektor yang digunakan untuk mengambil features dari data user dan item tersebut, sedangkan layer MLP mengambil gabungan dari embedding tersebut sebagai masukan yang kemudian untuk dipelajari hubungan secara non-linear antara user dan item. Penelitian Neural Collaborative Filtering (NCF) pertama kali diperkenalkan oleh He, X et al pada tahun 2017. Model NCF disusun menggunakan kombinasi dari kelebihan matrix factorization dan neural networks. Model menggunakan kombinasi interaksi user-item dan informasi tambahan seperti dari fitur item. Berikut merupakan Arsitektur Neural Collaborative Filtering :



**Gambar 2.4** Arsitektur Neural Collaborative Filtering (He, X et al.,2017).

Salah satu kelebihan dari Neural Collaborative Filteirng adalah mampu mengolah data secara explicit dan implicit. Data Explicit merupakan data yang secara langsung diberikan oleh user seperti data rating, sedangkan data implicit merupakan data yang secara tidak langsung yang terdapat pada interaksi user dengan item seperti aktifitas clicks, views dan memesan sesuatu. Berikut merupakan formula Neural Collaborative Filtering :

Text

Description automatically generated with medium confidence

**Gambar 2.5** Formula Neural Collaborative Filtering (He, X et al.,2017).

Keterangan :

P : Matrix vector user ukuran m x k

Q : Matrix vector item ukuran n x k

: Theta yang merupakan model parameter

Fungsi *f* merupakan Multi-Layer Perceptron yang jika dijabarkan akan berbentuk seperti berikut ini :



**Gambar 2.6** Formula Neural Collaborative Filtering (He, X et al.,2017).

Keterangan :

: Fungsi Mapping untuk layer output

: Fungsi Mapping untuk layer ke x neural collaborative filtering

#### 2.5.1 General Matrix Factorization

General Matrix Factorization (GMF) merupakan salah satu dari komponen dari arsitektur Neural Collaborative Filtering (NCF). GMF diimplementasikan sebagai representasi data interaksi user dengan item dalam bentuk matriks. Setelah dijadikan dalam bentuk matriks, matriks tersebut difaktorisasikan untuk menemukan representasi laten pada setiap item dan pengguna. GMF merupakan vektor numerik yang mana setiap elemen vektor merepresentasikan fitur atau atribut dari user atau item data tersebut. Dari kedua kedua representasi matriks tersebut dikalikan dan diambil dari hasil dot product yang kemudian akan menghasilkan interaksi item dan pengguna yang sesuai. Berikut merupakan bentuk matematis General Matrix Factorization :

A picture containing text

Description automatically generated

**Gambar 2.6** Formula General Matrix Factorization (He, X et al.,2017).

Keterangan :

: Fungsi Aktivasi

: Bobot tepian dari output layer

#### 2.5.2 Multi-Layer Perceptron

Multi-Layer Perceptron merupakan salah satu jenis arsitektur neural network. MLP pada Neural Collaborative Filtering digunakan untuk mengkombinasikan fitur dari kedua tipe data interaksi user dan item. MLP terdiri dari beberapa lapisan yang terhubung secara sekuesial yaitu lapisan input, lapisan tersembunyi dan lapisan output. Setiap lapisan terdiri dari sejumlah neuron yang saling terhubung dengan bobot tertentu. MLP digunakan untuk menggabungkan collaborative filtering dan content-based filtering sehingga sistem rekomendasi menjadi sistem rekomendasi hybrid yang mengambil kelebihan dari kedua jenis tipe sistem rekomendasi tersebut sehingga mampu menghasilkan rekomendasi yang lebih baik. Berikut merupakan formula Multi-layer perceptron :

Text, letter

Description automatically generated

**Gambar 2.6** Formula Multi-Layer Perceptron (He, X et al.,2017).

Keterangan :

: Bobot Matrix

: Bias

: fungsi aktivasi untuk layer ke-x perceptron

P : latent vector untuk user

q : latent vector untuk item

#### 2.5.3 Neural Matrix Factorization

Gabungan antara General Matrix Factorization dengan Multi-Layer Perceptron disebut dengan NeuMF. GMF dan MLP dipisahkan dalam mempelajari proses embeddings, dan penggabungan diantara keduanya dilakukan pada hidden layer terakhir (He, X et al.,2017). Keuntungan penggunaan GMF dan MLP pada NCF adalah GMF menerapkan kernel linear untuk model interaksi user-item sedangkan MLP menggunakan multiple neural layers untuk layer interaksi nonlinear. Pemisahan GMF dan MLP dalam penerapan user-item embedding karena untuk memastikan keduanya secara optimal mempelajari embedding secara independen. Berikut merupakan formula fusion GMF dan MLP :

Text, letter

Description automatically generated

**Gambar 2.7** Formula Fusion GMF dan MLP (He, X et al.,2017).

Model NCF mengkombinasikan linearity dari Matrix Factorization dengan Non-linearity dari Deep Neural Networks untuk memodelkan user-item struktur latent melalui layer NeuMF. Berikut merupakan arsitektur sebelum masuk ke layer NeuMF :

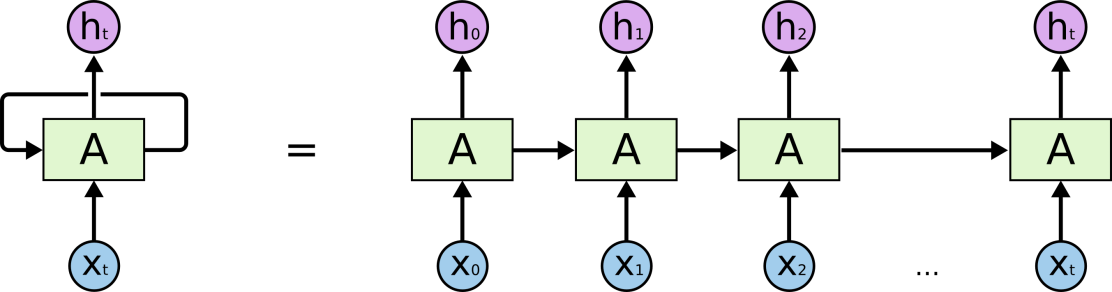
Diagram

Description automatically generated

**Gambar 2.6** Fusion GMF dan MLP (He, X et al.,2017).

### Recurrent Neural Network

*Recurrent Neural Network* (RNN) merupakan salah satu jenis arsitektur jaringan syaraf tiruan, RNN sendiri merupakan anggota keluarga dari *neural network* untuk menangani data yang berkelanjutan atau bersambung (*sequential data)* (Goodfellow, Ian et al., 2016). Berbeda dengan CNN yang digunakan secara efektif pada pengolahan data *spatial* Dengan menggunakan RNN yang didesain untuk menangani data sequential lebih baik (Gao, Z., & Wang, X., 2019). RNN disebut *reccurent* karena dalam prosesnya RNN melakukan proses yang sama pada setiap elemen dalam urutan, dengan keluaran yang bergantung pada perhitungan sebelumnya. Pada RNN konsepnya adalah bagaimana menangani data yang saling berhubungan satu sama lain atau data yang berututan atau bersambung (*sequential).* RNN memperkenalkan variabel status yang menyimpan informasi masa lalu, bersama dengan *input* saat ini serta untuk menentukan *output* saat ini. RNN merupakan metode yang berfokus pada sifat data dimana instance pada waktu sebelumnya yaitu (t-1) memberi pengaruh pada instance waktu selanjutnya (t), karena hal tersebut RNN mampu mengingat masa lalu (Jan Wira Gotama Putra, et al., 2020).



**Gambar 2.3 diagram recurrent neural network.**

Dalam bentuk matematisnya jika diberikan sebuah sekuens input . Data yang bisa merupakan data vektor, gambar, teks ataupun suara dipengaruhi oleh data sebelumnya yang bisa ditulis sebagai (Jan Wira Gotama Putra, et al., 2020). Meskipun RNN digunakan dengan tujuan untuk mengingat kejadian secara keseluruhan, tetapi secara praktikal hal itu sulit dilakukan dalam urutan kejadian yang panjang, hal ini ini dikenal dengan *vanishing* atau *exploding gradient problem*. Berikut konsep dasar dari *Reccurent Neural Network* :

Diagram

Description automatically generated

**Gambar 2.4 Konsep Neural Network (Jan Wira Gotama Putra, et al., 2020)**

Berdasarkan konsep sederhana pada diagram diatas, konsep tersebut sudah sessuai dengan tujuan dari konsep RNN yaitu mengingat kejadian sebelumnya (Jan Wira Gotama Putra, et al., 2020). Konsep RNN di atas bisa ditulis kembali dalam persamaan sebagai berikut :

……………………………………………….(2.1)

Keterangan :

: *hidden state* ke-t

: Fungsi *activation*

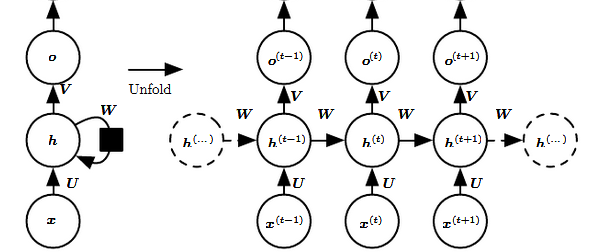
: Nilai *Input* ke-t

: hidden state ke t-1

b : bias

Dari persamaan diatas didapat *f* merupakan fungsi aktivasi yang adalah non-linear dan dapat diturunkan (Jan Wira Gotama Putra, et al., 2020). *b* adalah *bias* yang merupakan parameter yang selalu diikut sertakan kedalam *artficial neural network.* Konsep pada persamaan di atas punya analogi dengan *full markov chain* artinya, *hidden state* pada saat ke-t bergantung pada semua *hidden state* dan input sebelumnya. Fungsi *f* bisa diganti dengan fungsi variasi RNN lainnya misal nya bisa menggunakan *long short-term memory* (LSTM).fungsi diatas jika dijabarkan adalah sebagai berikut :

*,*



**Gambar 2.5 Diagram arsitektur RNN (Goodfellow, Ian et al., 2017)**

Pada diagram gambar 2.5 diatas merupakan detail dari gambar 2.3 dari diagram di atas terlhat proses RNN dengan simbol matematisnya, bagian kiri merupakan arsitektur dasar dari RNN sedangkan bagian kanan merupakan penjabaran dari bagian kiri. Sebagai contoh pada permasalahan kalimat dengan tiga kalimat, maka jaringan RNN yang tergambar seperti pada gambar 2.5 satu layer untuk masing - masing kata. Penjelasan dari diagram di atas adalah sebagai berikut :

**-** **Input**: *x(t)* merupakan *input* yang masuk kedalam jaringan pada langkah ke - t.

**-** **Hidden state:** *h(t)* merepresentasikan sebuah *hidden state* pada langkah ke - t yang bertindak sebagai *memory* pada jaringan. Seperti yang sudah dijelaskan sebelumnya fungsi *f* ini bisa digantikan dengan persamaan *non-linear* seperti *tanh, ReLU.*

**- Weights:** *U* merupakan parameter bobot yang terdapat pada hubungan *input* menuju *hidden state* sedangkan *W* merupakan nilai parameter bobot yang menghubungkan antara *hidden state* dengan *hidden state,* dan *V* merupakan parameter bobot yang menghubungkan *hidden* dengan *output.*

**- Output:** *o(t)* merupakan nilai keluaran dari proses yang telah dilakukan oleh RNN pada langkah ke t. Pada keluarna ini jika dirumuskan menjadi

* + 1. **Forward Propagation**

*Forward propagation* merupakan proses yang dilewati setiap *neural network.* Pada proses *forward propagation* ini *neural network* menerima *input* yang nantinya akan diproses oleh *hidden state*. Proses *forward propagation* pada proses *hidden state* diproses melalui *activation function.* Dengan proses *forward propagation* ini akan didapatkan hasil keluaran. RNN merupakan metode yang mengambil prinsip parameter sharing yang mana neuron yang sama akan diulang-ulang pada saat proses feed forward (Jan Wira Gotama Putra, et al., 2020). Pada setiap *neuoran* di *hidden state* atau *output layer* terdapat dua proses yang dijalani yaitu :

1. **Preactivation :** Pada proses ini dilakukan penjumlahan bobot dari *input.*
2. **Activation :** Berdasarkan proses *preactivation* sebelumnya yang merupakan proses penjumlahan bobot dari input, proses *activation* merupakan proses yang menggunakan fungsi matematika yang mana menambah persamaan non-linear ke jaringan, umumnya terdapat fungsi aktivasi popular yang digunakan yaitu *sigmoid, hyperbolic tangent(tanh), ReLU* dan *Softmax.*

Berikut merupakan formula persaman yang merepresentasikan *forward propagation* pada RNN :

…………………………………………………………(2.3)

…………………………………………………………………….(2.4)

………………………………………………………………………(2.5)

………………………………………………………………..(2.6)

Pada formula diatas ***b*** dan ***c*** merupakan vector bias dengan bobot matrik yaitu ***U, V*** dan ***W*** secara berurut-urut yaitu *input-ke-hidden, hidden-ke-output* dan *hidden-ke-hidden.* Contoh formula di atas merupakan RNN yang memiliki masukan *sequence*  dan keluaran *sequence* dengan Panjang yang sama (Goodfellow, Ian et al., 2017).

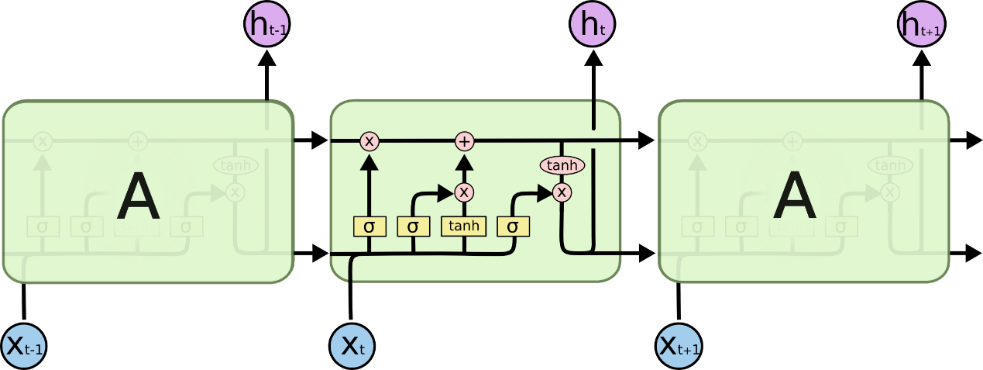
* + 1. **Backpropagation Trough Time**

*Backpropagation Through Time* merupakan *backpropagation* yang dilakukan pada RNN. *Backpropagation Through Time* atau disingkat dengan BPTT merupakan Teknik yang dilakukan untuk melatih RNN (Jan Wira Gotama Putra, et al., 2020). RNN tidak menggunakan *backprogation* untuk melakukan *training* dikarenakan metode tersebut kurang intuitif dalam menangani permasalahan data yang bersifat sekuensial (Jan Wira Gotama Putra, et al., 2020). Backpropagation melakukan pembaharuan parameter pada bobot dan bias sesuai error yang didapatkan pada saat foorward propagation dengan tujuan untuk meminimalkan error yang didapat.

Pada backpropagation through time untuk memperbaharui parameter saat telah mencapai hidden state paling awal berbeda dengan *backpropagation* yang pada umumnya memperbaharui parameter bersamaan dengan mempropagasi error dari hidden state ke hidden state sebelumnya.. RNN merupakan salah satu neural network yang menggunakan teknik *stochastic gradient decent* yang merupakan variasi dari *gradient decent*. *Stohastic gradient decent* digunakan karena *gradient decent* tidak cocok dalam penggunaan dataset yang besar karena kemampuannya yang memakan waktu sangat lama. Dengan RNN yang menggunakan pencarian global mnimal error tersebut menghasilkan permasalahan yaitu *vanishing gradient* atau *exploiding gradient*. Kedua permasalahan tersebut terjadi karena pencarian parameter yang bisa mencapai nol (*vanisihing gradient)* atau melebihi batas seharusnya (*exploiding gradient)* sehingga dibutuhkan solusi lain dari hal tersebut salah satunya menggunakan metode baru yaitu LSTM.

### Long-Short Term Memory

Long Short Term Memory atau biasa disingkat dengan LSTM merupakan pengembangan dari metode RNN. LSTM didesain untuk menangani permasalahan pada saat pembelajaran *long-term dependencies.* Permasalahan tersebut bisa disebut juga sebagai *vanishing gradient* yaitu ketidakmampuan RNN dalam menangani menyimpan informasi pada jangka waktu yang panjang (long-term dependencies). Dengan LSTM hal tersebut bisa teratasi dengan *memory cell* dan *gate units.* Dengan *memory cell* menjadikan LSTM bisa memiliki kemampuan menyimpan dan menghapus informasi yang diregulasi oleh *gate units* (Sherstinsky, A., 2020).



**Gambar 2.6 Diagram arsitektur LSTM**

Diagram

Description automatically generated

**Gambar 2.7 Komponen pada LSTM**

Sesuai pada gambar 2.7 terdapat komponen – komponen pada LSTM yang membuat LSTM bekerja seperti fungsi *sigmoid* yang berfungsi sebagai memperbaharui atau mengubah nilai antara -1 dan 1 menjadi nilai yang diantara 0 dan 1, hal ini bertujuan untuk memperbaharui atau melupakan data karena angka yang dikalikan dengan angka 0 akan menghasilkan nilai 0 sehingga menyebabkan nilai menghilang dilain sisi setiap angka yang dikalikan dengan 1 akan tetap bernilai sama sehingga nilai tersebut akan tetap disimpan, sedangkan untuk fungsi *tanh* digunakan sebagai cara untuk mengontrol nilai yang melewati jaringan agar selalu berada diantara nilai -1 dan 1, operator pointwise serta *vector concatenation* sebagai jalannya proses LSTM. Pada gambar 2.6 terdapat simbol kotak yang berwarna kuning menunjukkan bahwa simbol tersebut adalah layer *neural network.*

Pada struktur jaringan LSTM tersusun oleh blok memori yang disebut dengan sel. Pada LSTM terdapat dua jalur yang menghubungkan informasi ke proses - proses selanjutnya yaitu *cell state* dan *hidden state*. Konsep inti pada LSTM adalah *cell state*. *Cell state* pada dasarnya merupakan "memory" pada jaringan LSTM. Informasi yang dibawa oleh cell state diatur oleh *gates* yang melakukan proses penambahan atau penghapusan informasi. Berikut pada gambar 2.8 merupakan struktur *cell state* pada LSTM.

Timeline

Description automatically generated

**Gambar 2.8 Struktur cell state LSTM**

Pada gambar 2.8 diatas garis hitam lurus merupakan *cell state*. Sedangkan untuk *hidden state* yang sudah menjadi bagian dari model RNN yang digunakan untuk menge*ncode* informasi yang diterima dari nilai masukan *hidden state* sebelumnya berbeda dengan *cell state* yang berfungsi sebagai *long memory* sedangkan *hidden state* merupakan *memory* yang bekerja pada jangka pendek *hidden state* juga mengatur proses *gates*. Pada LSTM terdapat tiga jenis *gates* yaitu *input gate*, *forget gate* dan *output gate*. *Forget gate* berfungsi untuk menentukan informasi atau masukan apa yang akan dipertahankan atau dihapus dari *cell state* ataupun ke proses selanjutnya, semakin nilainya mendekati nilai nol maka artinya nilai tersebut akan dilupan dan nilai yang mendekati nilai 1 akan dipertahankan. *Input gate* berfungsi sebagai untuk menentukan nilai pada cell state memory dari masukan apakah akan diperbaharui atau tidak. *Output gate* berfungsi untuk menentukan seperti apa hidden state selanjutnya yang akan ditentukan oleh masukan sebelumnya dan *cell state* baru.

Berikut merupakan proses LSTM bekerja (Hochreiter dan Schmidhber., 1997):

1. Tahap pertama, Proses akan dimulai dari forget gate, pada tahap ini akan ditentukan informasi mana saja yang akan dipertahankan atau dihapus. Pada tahapan ini berdasarkan dari *output* hidden state sebelumnya dan input akan diproses menggunakan fungsi aktivasi sigmoid. Output yang akan dihasilkan dari proses ini adalah nilai antara 0 sampai 1 pada cell state. Berikut formula *forget gate* berdasarkan Hochreiter dan Schmidhuber (1997) :

………………………………………(2.7)

Keterangan :

: *forget gate*

: fungsi sigmoid

: masukan order pada langkah ke-t

: keluaran pada langkah yang sebelumnya

: bobot pada *forget gate*

: *reccurent weight* pada *forget gate*

: bias *forget gate*

Berikut merupakan alur proses pada *forget gate* :

A picture containing diagram

Description automatically generated

**Gambar 2.9 Forget Gate**

2. Langkah selanjutnya bertujuan untuk menentukan informasi baru apa yang akan ditambahkan ke *cell state* berdasarkan dari *hidden state* sebelumnya dan data *input* baru*.* Langkah ini terdiri dari dua bagian, pada bagian pertama yatiu *input gate* melakukan penentuan nilai yang akan diperbaharui menggunakan fungsi aktivasi *sigmoid*, selanjutnya untuk bagian kedua adalah sebuah layer *tanh* membuat sebuah vektor untuk calon nilai baru (*cell state).* Berikut ilustrasi pada langkah ini :

Diagram

Description automatically generated

**Gambar 2.10 Input Gate**

Pada proses ini terdapat dua formula yaitu perhitungan untuk *input gate* dan penentuan calon konteks yang akan ditambahkan ke *cell state*  yang ditunjukkan pada formula berikut :

……………………………………,,,,….…(2.8)

…………………………………,,,,…...(2.9)

Keterangan :

: *input gate*

: calon konteks

: fungsi *sigmoid*

: *weight* pada *input gate*

: *reccurent weight* pada *input gate*

: nilai pada *input gate*

: *weight* pada calon konteks

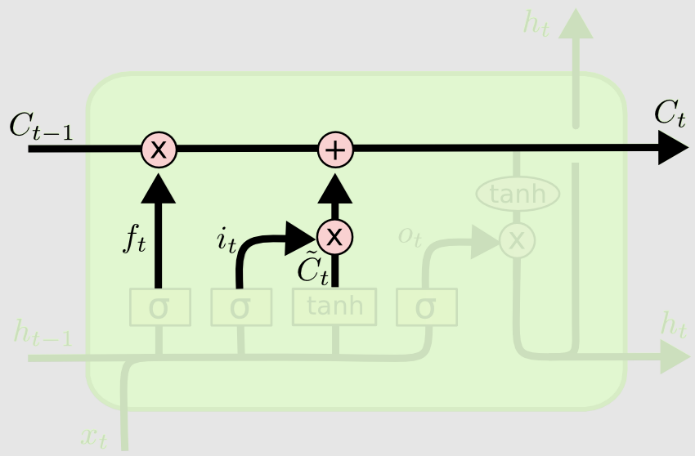
: *reccurent weight* pada calon konteks

: nilai bias pada calon konteks

: nilai *input* pada orde ke-t

: nilai *output* sebelum order ke-t

1. Pada tahap ini dilakukan untuk memperbaharui cell state dari menjadi .Berdasarkan dari proses sebelumnya yang menghasilkan nilai *input gate* dan calon konteks untuk cell state, pada tahapan ini akan dilakukan proses perkalian dari kedua nilai keluaran tersebut untuk memperbaharui nilai *cell state*. Sebelumnya juga hasil *output* dari *forget gate* () dengan *cell state* lama () dikalikan. Dari kedua hasil perkalian itu dijumlahkan, dari penjumlahan ini menghasilkan *cell state* baru (). Berikut ilustrasi yang menggambarkan proses pada tahap ini serta formula yang menyertakan proses ini dijelaskan pada persamaan 2.10 :



**Gambar 2.10 Cell State**

………………………………………………………(2.10)

Keterangan :

: *cell state* saat ini

: *forget gate*

: *cell state* sebelumnya

: nilai *input*  pada order ke – t

: calon konteks

1. Pada tahap akhir ini adalah menghasilkan *output* akhir dari serangkaian proses sebelumnya yaitu *hidden state* selanjutnya (). Pada tahap akhir ini memiliki dua tahapan, pertama proses ini akan menentukan *output gate* yang diproses melalui layer *sigmoid* pada layer ini menggunakan *output* sebelumnya () dan *input* () untuk mengetahui *ouput gate* (). Hasil *output gate* akan menghasilkan nilai di antara 0 sama dengan 1. Tahapan selanjutnya adalah menentukan hasil *hidden state* atau *output* akhir (), nilai dari *output gate* akan dikalikan dengan nilai *cell state* )yang diubah menggunakan fungsi *tanh.* Berikut merupakan ilustrasi pada proses tahapan ini yang disertakan dengan pada persamaan 2.11 dan 2.12 :

A picture containing text, clock

Description automatically generated

**Gambar 2.11 Output Gate**

………………………………………….(2.11)

…………………………………………………………….(2.12)

Keterangan :

: *output gate*

: fungsi sigmoid

: nilai input pada orde ke - t

: nilai *weight* pada *output gate*

: *reccurent weight* pada *output gate*

: nilai *output* pada order ke – t-1

: nilai bias pada *output gate*

: *cell state*

### 2.7 Bidirectional Long-Short Term Memory

Bidirectional Long Short-Term Memory atau disingkat dengan BiLSTM merupakan model LSTM yang telah dikembangkan yang memiliki masukan dengan proses dengan dua arah yaitu secara *forward* dan *backward.* BiLSTM memiliki kelebihan dibandingkan pendahulunya LSTM yaitu kemampuan memahami konteks kaliamt secara dua arah yaitu mampu membaca teks dari awal sampai akhir dan dari akhir ke awal sehingga membuat proses pemahaman konteks yang terdapat pada kalimat lebih mudah (Pasaribu et al., 2020). Penerapan BiLSTM dengan *attention mechanism* pernah diteliti oleh (Pratiwi, R. et al 2020) pada penelitian tersebut terbukti bahwa BiLSTM dengan Attention mampu berhasil mencapai akurasi yang lebih baik dibandingkan penelitian dengan metode lain. penerapan BiLSTM dengan menggunakan attention memberikan pengaruh yang cukup berarti karena dengan menggunakan attention model mampu untuk fokus pada kata yang yang penting untuk mencapai hasil yang maksimal, kekurangan dari penggunaan attention terdapat pada pengolahan kalimat negasi, dikarenakan pada layer attention tidak bisa menentukan kata tertentu untuk diperhatikan. Berikut arsitektur BiLSTM tertera pada **gambar 2.14 :**

Diagram

Description automatically generated

**Gambar 2.14 Struktur BiLSTM**

**Sumber:** (Yildirim, Ö. (2018))

Pada **gambar 2.14** terlihat bahwa arsitektur struktur BiLSTM terdiri dari dua lapisan LSTM yang berlawanan arah, secara maju dan mundur, berikut merupakan persamaan dari BiLSTM :

= + + …………………………………………………(2.13)

Keterangan :

: *output* BiLSTM

: *weight* pada *output gate* LSTM maju

: nilai keluaran LSTM maju

: *weight* pada *output gate* LSTM mundur

: nilai keluaran LSTM mundur

: bias *output* BiLSTM

#### 2.8 Attention Mechanism

Mekanisme Atensi (Attention Mechanism) merupakan teknik yang umumnya dikenal sering digunakan pada bidang *natural languange processing* (NLP) yang menggunakan model *Seq2Seq encoder-decoder* umumnya *standard-encoder* melakukan operasi dengan cara mengencoder proses masukan *sequence* dan kemudian mengkompres atau menyederhanakan informasi tersebut menjadi konteks vektor yang mempunyai panjang yang tetap untuk kemudian diteruskan ke *decoder*. Kekurangan dari panjang vektor yang tetap ini mengakibatkan ketidakmampuan sistem dalam mengingat *sequences* yang panjang serta dalam menentukan kepentingan informasi terbaru terlepas dari relevansi yang sebenarnya, dengan menggunakan permasalahan tersebut berhasil diatasi (Katrompas, A., & Metsis, V., 2022).

Attention Mechanism merupakan algoritma yang bertujuan untuk membantu model *neural network* meningkatkan performanya dengan cara fokus pada *local feature* yang mempunyai hubungan lebih kuat dibanding yang lainnya pada saat model dilatih. Teknik mekanisme memberikan memberikan kemampuan pada model untuk fokus pada bagian terpenting pada target dengan pembobotan yang berbeda (Xu, C., et al 2021). Berikut ilustrasi dari attention layer

Diagram

Description automatically generated

**Gambar 2.4 Attention Layer pada Seq2Seq model (Bahdanau et al., 2015)**

Atensi mekanisme memiliki konsep yang pada dasarnya yaitu dengan cara memberikan *weighted access* pada tiap timestep yang bertujuan untuk meningkatkan kemampuan model dalam memproses *sequential data*. *Attention mechanism* telah membuktikan mampu memberikan bobot pada item tertentu sesuai kepentingan item tersebut dan mekanisme perhatian juga berguna untuk membentuk karakteristik pengguna. (Huang, R., et al 2018). Umumnya Attention Mechanism terdiri dari tiga komponen utama yaitu *Q* (*queries*)*, K* (*keys*) *dan V* (*values*). Sebuah fungsi attention merupakan fungsi yang memetakan query dan satu set pasangan key dan value yang bertugas sebagai output, dari komponen - komponen tersebut semuanya merupakan vector.

Diagram

Description automatically generated

**Gambar 2.5 Scaled Dot-Product Attention** (Vaswani, A., et al 2017)

*Scaled Dot-Product Attention* merupakan komponen dari *Multi-head Attention*, *Scaled Dot-Product* merupakan salah satu bentuk fungsi *attention* yang lebih cepat, dan memiliki ruang effisiensi yang lebih luas dalam praktiknya karena pada penerapannya menggunakan matriks multiplication yang telah teroptimmasi (Vaswani, A., et al 2017). Bentuk formula yang digunakan fungsi *attention* adalah sebagai berikut :

Text, letter

Description automatically generated

**Gambar 2.6 Formula Scaled Dot-Product Attention** (Vaswani, A., et al 2017)

Keterangan :

Q : Query

K : Key

V : Value  
 : Dimension key

### 2.8 Pre training

Pre training merupakan salah satu bagian proses penting yang terdapat pada BERT. Pre-trained model merupakan model yang telah latih oleh orang lain yang nantinya akan digunakan kembali pada penelitian lain sehingga akan memudahkan proses penelitian serta membantu pada tugas model dengan tujuan tertentu. Pada BERT terdapat dua proses yang terdapat pada pre-trained yaitu *Masked Language Model* dan *Next Sentence Prediction* (Devlin, J et al., 2019).

2.9 Word Embedding

2.9.1 Word2Vec

2.9.2 Glove

### 2.9 Normalized Discounted Cumulative Gain

NDCG (Normalized Discounted Cumulative Gain) merupakan metode metrik yang digunakan untuk memperhitungkan kualitas dari perankingan suatu data seperti pada perankingan hasil sistem rekomendasi. Sebelum menjelaskan lebih lanjut mengenai NDCG perlu penulis deskripsikan terlebih dahulu dari Gain, Cumulative Gain dan Discounted Cumulative Gain. Gain pada konteks NDCG merupakan score seberapa relevannya setiap item yang direkomendasikan sedangkan Cumulative Gain adalah jumlah nilai gain pada item K dari item K pertama yang direkomendasikan. Berikut bentuk formula Cumulative Gain :

………………………………………………………………….(2.26)

Discounted Cumulative Gain (DCG) merupakan jumlah bobot dari tingkat relevansi item - item yang diperingkat (Wang, Y., et al 2013). Bobot yang terdapat pada DCG merupakan fungsi penurunan dari posisi objek, karena hal tersebut diberi nama discount. Alasan utama dari *discount* tersebut adalah probabilitas pengguna melihat item tersebut akan menurun sehubungan dengan posisi peringkat item tersebut. jadi item pada rekomendasi teratas memiliki bobot item lebih besar di sisi lain rekomendasi yang berada di tingkat bawah mendapatkan bobot lebih rendah. Berikut bentuk formula Discounted Cumulative Gain :

…………………………………………………………(2.27)

NDCG seperti namanya merupakan bentuk DCG yang telah dinormalisasikan pada penyebutnya. Dengan menggunakan NDCG terdapat dua kelebihan dibandingkan metrik lainnya yaitu pertama, NDCG memungkinkan setiap dokumen yang diambil memiliki relevansi berjenjang dibandingkan metrik pengukuran tradisional lainnnya yang hanya memungkinkan relevansi biner. Dengan kelebihan tersebut berarti dokumen dipandang secara relevan atau tidak relevan berdasarkan pengukuran perangkingan sebelumnya sementara dengan NDCG terdapat derajat relevansi pada dokumen. Kedua, pada NDCG terdapat fungsi diskon atas peringkat yang mana pada metode metrik lainnya pembobotan dilakukan secara seragam pada semua posisi. Berikut merupakan formula NDCG :

……………………………………………………………...(2.28)

……………………………………………………..(2.29)

Keterangan :

G : Gradien

K : Posisi rank item

I : posisi indeks G

### 2.11 Studi Pustaka (*State of the Art)*

Penelitian ini dilakukan karena...

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Penulis** | **Metode** | **Dataset** | **Hasil** |
| 1 | Billah, M et al. (2021) | Collaborative Filtering, PCA dan K-Means | Film Anime dataset | kompleksitas waktu sebesar 2.999602 dengan menggunakan akurasi nilai MMR dengan rata-rata sebesar 0.5619 |
| 2 | Nuurshadieq, & Wibowo, A. T. (2020). | Long short-term Memory | MyAnimeList dataset | Evaluasi menggunakan RMSE 1.4475 |
| 3 | Juarto, B., & Suganda Girsang, A. (2021). | Neural Collaborative dan Sentence BERT | Microsoft news dataset | Ratio@10 sebesar 74% pada epoch 50, serta Accuracy 95.83%, Precision 92.69%, Recall 98.61%, F1 95.56% dan ROC 98% |
| 4 | Wu, F et al. (2018) | LSTM dan Self Attention | Microsoft  News Dataset (MIND) | Menghasilkan AUC sebesar 66.91, MRR sebesar 32.48, nDCG@5 sebesar 35.12 dan nDCG@10 sebesar 40.85 |
| 5 | Wang, H., Lou, N., & Chao, Z. (2020) | LSTM dan CNN | MovieLens dataset | MSE dan MAE dari LSTM-CNN adalah 0.7724 dan 0.691739 masing - masing untuk MSE dan MAE serta dengan running time sebesar 64.2548. |
| 6 | Wang, T., & Fu, Y. (2020) | BiLSTM dan BERT | e-commerce dataset | Hasil terbaik mengunakan BERT Prec@1 0.555, Prec@10 0.079, Recall@10 0.791 dan NDCG@10 0.669 |
| 7 | Pratiwi, R. W et al. (2020) | Attention-Based BiLSTM | Twitter dataset | Accuracy 79.68%, Precision 78.37%, Recall 79.26% dan F1 Score 78% |
| 8 | Xu, C et al. (2021) | LSTM dan self-attention | Gowalla, ML-10M dan Foursquare dataset | Dengan evaluasi metrics Hit Ratio, NDCG dan MAP menghasilkan hasil evaluasi yang lebih baik disbanding penelitian lainnya. |
| 9 | Lund, J & Ng, Yiu-Kai. (2018). | Autoencoder | MovieLens dataset | Menghasilkan evaluasi yang lebih baik disbanding dengan KNN, hasil evaluasi dengan RMSE mendapat nilai sebesar 0.42 |
| 10 | Rosa, R. L, et al (2019) | CNN, Bi-LSTM-RNN menggunakan SoftMax | Online Social Network datasets (facebook) | Menghasilkan accuracy sebesar 0.89 dan 0.90 dalam mendeteksi emosi depresi dan stress user, selain itu hasil dari rekomendasi sistem mencapai 94% user yang sangat puas, dibandingkan dengan sistem rekomendasi yang tidak menggunakan sentimen analysis yang hanya mencapai tingkat kepuasan 69% |

Berdasarkan beberapa penelitian yang telah penulis telusuri sebelumnya terdapat beberapa metode yang merupakan metode yang terbaik yang bisa diterapkan serta objek yang bisa penulis teliti di penelitian ini yaitu Penelitian ini akan membandingkan dua metode yaitu metode BERT dan BiLSTM dengan attention dengan menggunakan dataset myanimelist.

# BAB III Metodologi Penelitian

## Metodologi Penelitian

Pada bagian ini akan dibahas tentang metodologi penelitian yang berisi tahapan - tahapan yang akan diterapakan pada penelitian ini. Dengan kerangka seperti ini tahapan penelitian akan lebih mudah dipahami dan diikuti oleh pihak yang tertarik akan penelitian ini. Tahapan Penelitian ini bisa di lihat pada gambar 3.1 di bawah :

Diagram

Description automatically generated

**Gambar 3.1 Tahapan Penelitian**

## Studi Literatur

Studi Literatur merupakan suatu tahapan atau kegiatan yang melakukan pengumpulan data yang bertujuan untuk mendapatkan informasi penting yang berhubungan dengan masalah penelitian, metode serta berbagai sumber tulisan dari penelitian - penelitian sebelumnya. Sumber pengumpulan data pada studi literatur harus dari sumber yang bisa dipercaya seperti buku, artikel, jurnal dan lain sebagainya. Studi literatur pada penelitian ini bertujuan untuk mencari berbagai informasi yang berkaitan dengan penelitian sistem rekomendasi serta hal yang relevan dengannya.

## Pengumpulan Data

Pengumpulan data adalah tahapan yang diterapkan untuk mendapatkan berbagai informasi yang akan digunakan pada penelitian ini. Data yang akan digunakan pada penelitian ini merupakan data sekunder yang berasal dari website MyAnimeList, data tersebut didapatkan secara tidak langsung atau sudah ada pihak ketiga yang sudah mengumpulkan. Data MyAnimelist sudah tersedia di Kaggle, terdapat beberapa dataset yang telah dikumpulkan oleh pihak ketiga yang cukup lengkap dengan update terakhir pada tahun 2019. Dataset tersebut bisa didapatkan dengan dua cara yaitu *Web Scrapping* atau melalui API. Sebelum bisa digunakan dataset tersebut perlu dilakukan proses *cleaning* dan *tokenizing* sehingga memudahkan dalam pengolahan model nantinya.

## Pre-processing

Preprocessing merupakan tahapan yang perlu dilakukan untuk mempermudah model dalam melakukan memproses data, tahapan praprocessing juga dilakukan unutk memastikan data merupakan data yang berkualitas sehingga bisa membantu meningkatkan keakuratan model dan efisien termasuk mengurangi kemungkinan kesalahan yang akan terjadi pada model yang akan dibuat. Pada tahapan *praprocessing* ini terdiri dari tahapan seperti berikut ini *flowchart* proses *preprocessing* dapat dilihat pada gambar 3.2 berikut :

A picture containing diagram

Description automatically generated

**Gambar 3.2 Flowchart Preprocessing**

Berdasarkan flowchart propressing data di atas, sebelum siap digunakan data - data tersebut memiliki beberapa kolom yang tidak diperlukan pada proses pemodelan nanti sehingga perlu dilakukan pengolahan terlebih dahulu yaitu dengan memodifikasi kolom pada dataset tersebut pada dataset tersebut juga dilakukan penggabungan dataset setelah proses memodifikasi proses selanjutnya adalah melakukan data cleaning, data yang memiliki karakter - karakter yang tidak diperlukan seperti tanda baca ataupun angka - angka yang tidak dibersikan dari dataset setelah dilakukan data cleaning kemudian dilakukan pembagian dataset yang akan digunakan sebagai data latih dan data uji rasio yang digunakan adalah 7 dibanding 3.

### Memodifikasi kolom dataset

Pada penelitian ini hanya menggunakan beberapa kolom dataset, beberapa kolom dataset akan dihapus sehingga mengurangi beban pengolahan model nantinya pada saat dilatih. Seperti yang terlihat pada **gambar 3.3** flowchart di bawah terdapat tiga proses yaitu penghapusan kolom data yang tidak diperlukan, penggabungan kolom antara dataset dan mengganti nama kolom. Ketiga proses diatas dilakukan juga untuk mempermudah pengenalan dalam proses pengimplementasian sistem rekomendasi. Kolom – kolom yang akan digunakan adalah kolom *anime\_id*, *user\_id*,, *episode* *user\_rating*, dan *genre*

A picture containing application

Description automatically generated

**Gambar 3.3 Flowchart Memodifikasi kolom**

### Data Filtering

Pada proses *data filtering* ini tahapan proses yang dijalankan ada dua tahapan yaitu pertama menghapus data rating yang bukan angka artinya terdapat data rating yang masih bukan bertipe data angka atau user tersebut belum memberi nilai pada judul anime yang user tersebut ratingkan, proses ini dilakukan untuk menghindari *error* pada saat mengkonversi nilai menjadi nilai matriks. Tahapan selanjutnya adalah menghapus data rating yang tidak tersedia pada dataset anime, tahapan ini dilakukan untuk menghindari data *error*. Proses *data filtering* bisa dilihat pada *flowchart* **gambar 3.3** berikut :

A picture containing timeline

Description automatically generated

**Gambar 3.4 Flowchart Data Filtering**

### Data Cleaning

### A picture containing polygon Description automatically generated

**Gambar 3.4 Flowchart Data Filtering**

Pada tahapan ini beberapa proses dilakukan untuk membersihkan atau memperbaiki dataset semakin lebih baik lagi dari proses sebelumnya. Proses pertama yang dilakukan adalah mengubah beberapa tipe data kolom yang ada di dataset menjadi integer. Proses selanjutnya adalah menromalisasikan nilai rating menjadi skala 0 - 1 sehingga mempermudah pemrosesan model. Tahapan selanjutnya kemudian menghapus tanda baca atau karakter spesial yang terdapat pada teks untuk mempermudah pengubahan teks pada saat dilakukan tokenisasi nanti, proses terakhir adalah tokenisasi yaitu proses yang memisahkan kalimat menjadi per kata, pemisahan berdasarkan spasi antar kata.

### Data Splitting

Data yang telah melalui beberapa tahapan sebelumnya kemudian akan dipisahkan menjadi dua bagian yaitu data latih dan data uji yang mana data tersebut akan dibagiakan dalam rasio 7 dibanding 3, yang artinya 70% untuk data latih dan sisanya 20% dijadikan data uji.

## Pembuatan Model

Proses yang dilakukan setelah melakukan data *pre-processing* adalah proses pembuatan model, model yang dibuat merupakan *Neural Collaborative Filtering*, yang merupakan kombinasi dari *Matrix Factorization* dengan *Multi-Layer Percepetron*, metode tersebut merupakan metode *deep learning* yang pada penelitian ini menerapakan sistem rekomendasi *hybrid filtering* yang mengolah data secara *collaborative filtering* dan *content based filtering*.

### General Matrix Factorization

### Multi-Layer Perceptron

### Neural Collaborative Filtering

## Pengujian Sistem

## Metodologi Pengembangan Sistem

## Analisis Kebutuhan Sistem

## Kebutuhan Fungsional

## Kebutuhan Non-Fungsional

## Proses Desain

## Perancangan Sistem

## Perancangan Pengujian

# Daftar Pustaka

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B,. (2011). Recommender Systems Handbook. In Recommender Systems Handbook.

Cheng, H. T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., Anderson, G., Corrado, G., Chai, W., Ispir, M., Anil, R., Haque, Z., Hong, L., Jain, V., Liu, X., & Shah, H. (2016). Wide & deep learning for recommender systems. ACM International Conference Proceeding Series, 15-Septemb, 7–10.

Burke, R., Felfernig, A., & Göker, M. H. (2011). Recommender Systems: An Overview. AI Magazine, 32(3), 13.

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, 16(3), 261–273.

Modallal, S. (n.d.). Survey on Collaborative Filtering , Content-based Filtering and Hybrid Recommendation System Filtering and Hybrid Recommendation System (Modallal, S ., 2011).

Burke, R. (2007). Hybrid web recommender systems. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4321 LNCS, 377–408.

Suyanto., Nur Ramadhan, Kurniawan., Mandala, Satria., (2019) Penerbit Informatika

Alfarhood, M., & Cheng, J. (2021). Deep learning-based recommender systems. Advances in Intelligent Systems and Computing, 1232(1), 1–23.

Ng, Andrew. (n.d.). Structuring Machine Learning Projects [MOOC]. Coursera.

Batmaz, Z., Yurekli, A., Bilge, A., & Kaleli, C. (2019). A review on deep learning for recommender systems: challenges and remedies. Artificial Intelligence Review, 52(1).

Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., Gulcehre, C., Song, F., Ballard, A., Gilmer, J., Dahl, G., Vaswani, A., Allen, K., Nash, C., Langston, V., … Pascanu, R. (2018). Relational inductive biases, deep learning, and graph networks. 1–40.

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, 404(March), 1–43.

Goodfellow, Ian., Bengio, Yoshua., Courville, Aaron., (2016). Deep Learning. Nature, 26(7553), 436

Gao, Z., & Wang, X. (2019). Deep learning. EEG Signal Processing and Feature Extraction, 325–333.

Lops, P., Gemmis, M. De, & Semeraro, G. (2011). Recommender Systems Handbook. In Recommender Systems Handbook (Issue January).

Nuurshadieq, & Wibowo, A. T. (2020). Leveraging Side Information to Anime Recommender System using Deep learning. 2020 3rd International Seminar on Research of Information Technology and Intelligent Systems, ISRITI 2020, 62–67.

Jindal, R., & Jain, K. (2019). A review on recommendation systems using deep learning. International Journal of Scientific and Technology Research, 8(10), 2978–2985.

Jan Wira Gotama Putra, et al. (2020). Pengenalan konsep pembelajaran mesin dan deep learning. Computational Linguistics and Natural Language Processing Laboratory, 4, 1–235.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. Advances in Neural Information Processing Systems, 2017-Decem(Nips), 5999–6009.

Xu, C., Feng, J., Zhao, P., Zhuang, F., Wang, D., Liu, Y., & S. Sheng, V. (2021). Long- and short-term self-attention network for sequential recommendation. Neurocomputing, 423, 580–589.

Huang, R., McIntyre, S., Song, M., Haihong, E., & Ou, Z. (2018). An attention-based recommender system to predict contextual intent based on choice histories across and within sessions. Applied Sciences (Switzerland), 8(12).

Katrompas, A., & Metsis, V. (2022). Enhancing LSTM Models with Self-attention and Stateful Training. Lecture Notes in Networks and Systems, 294, 217–235.

Zhang, S., Tay, Y., Yao, L., & Sun, A. (2018). Next Item Recommendation with Self-Attention. August.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference, 1(Mlm), 4171–4186.

Kaviani, M., & Rahmani, H. (2020). EmHash:基于BERT嵌入的神经网络Hashtag推荐. *2020 6th International Conference on Web Research, ICWR 2020*, 113–118.

Pratiwi, R. W., Sari, Y., & Suyanto, Y. (2020). Attention-Based BiLSTM for Negation Handling in Sentimen Analysis. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *14*(4), 397.

Zhuang, Y., & Kim, J. (2021). A bert-based multi-criteria recommender system for hotel promotion management.

Lund, J & Ng, Yiu-Kai. (2018). Movie Recommendations Using the Deep Learning Approach.

Rosa, R. L., Schwartz, G. M., Ruggiero, W. V., & Rodriguez, D. Z. (2019). A Knowledge-Based Recommendation System That Includes Sentiment Analysis and Deep Learning. IEEE Transactions on Industrial Informatics, 15(4), 2124–2135.

Zhang, X., Yuan, X., Li, Y., & Zhang, Y. (2019). Cold-start representation learning: A recommendation approach with BeRt4Movie and Movie2vec. MM 2019 - Proceedings of the 27th ACM International Conference on Multimedia, 2612–2616.

Wang, Y., Wang, L., Li, Y., He, D., Chen, W., & Liu, T. Y. (2013). A theoretical analysis of NDCG ranking measures. Journal of Machine Learning Research, 30, 25–54.

Soni, B., Thakuria, D., Nath, N., Das, N., & Boro, B. (2021). RikoNet: A Novel Anime Recommendation Engine.

Islek, I., & Oguducu, S. G. (2020). A Hybrid Recommendation System Based on Bidirectional Encoder Representations. Communications in Computer and Information Science, 1323, 225–236.

Cordonnier, J.-B., Loukas, A., & Jaggi, M. (2019). On the Relationship between Self-Attention and Convolutional Layers.

Erritali, M., Hssina, B., & Grota, A. (2021). Building Recommendation Systems Using the Algorithms KNN and SVD. International Journal of Recent Contributions from Engineering, Science & IT (IJES), 9(1), 71.

Pratiwi, R. W., Sari, Y., & Suyanto, Y. (2020). Attention-Based BiLSTM for Negation Handling in Sentimen Analysis. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *14*(4), 397.

Yildirim, Ö. (2018). A novel wavelet sequences based on deep bidirectional LSTM network model for ECG signal classification. *Computers in Biology and Medicine*, *96*, 189–202. https://doi.org/10.1016/j.compbiomed.2018.03.016

# Lampiran