IMPLEMENTASI SISTEM REKOMENDASI TEKS PADA JUDUL ANIME DENGAN METODE BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS

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 2
BAB I Pendahuluan 1
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan Penelitian 4
1.5 Manfaat Penelitian 4
1.6 Tahapan Penelitian 4
1.6.1 Rencana dan Tahapan Penelitian 4
1.6.2 Metode Pengembangan Sistem 5
1.6.3 Pengujian Sistem 6
1.6.4 Pengujian Penelitian 6
1.7 Sistematika Penulisan 6
BAB II Tinjauan Pustaka 7
2.1 Anime 7
2.2 Sistem Rekomendasi 7
2.2.1 Content Based Filtering 8
2.2.2 Collaborative Filtering 9
2.2.3 Hybrid Filtering 10
2.3 Deep Learning 10
2.4 Recurrent Neural Network 11
2.5 Long-Short Term Memory 15
2.6 Attention Mechanism 20
2.7 Bidirectional Long-Short Term Memory 22
2.8 Transformers 22
2.9 Bidirectional Encoder Representations from Transformers 26
2.10 Normalized Discounted Cumulative Gain 27
2.11 Studi Pustaka (*State of the Art)* 29
BAB III Metodologi Penelitian 32
3.1 Metodologi Penelitian 32
3.1.1 Studi Literatur 32
3.1.2 Pengumpulan Data 33
3.1.3 Pre-processing 33
3.2 Pembuatan Model 33
3.2.1 BiLSTM 33
3.2.2 Attention 33
3.2.3 BERT 33
3.3 Pengujian Sistem 33
3.4 Metodologi Pengembangan Sistem 33
3.4.1 Analisis Kebutuhan Sistem 33
3.4.2 Kebutuhan Fungsional 33
3.4.3 Kebutuhan Non-Fungsional 34
3.4.4 Proses Desain 34
3.4.5 Perancangan Sistem 34
3.4.6 Perancangan Pengujian 34
BAB IV HASIL DAN PEMBAHASAN ..... **Error! Bookmark not defined.**
BAB V KESIMPULAN ..... **Error! Bookmark not defined.**
Daftar Pustaka 34
Lampiran 37

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) Industri anime pada masa sekarang berkembang secara terus menerus dari tahun ke tahun meskipun sempat terjadi penurunan, berdasarkan Anime Report 2020 yang dibuat oleh Asosiasi Animasi Jepang ukuran pasar selama sepuluh tahun terakhir terus berkembang dengan penjualan sebesar 2,51 trilyun yen.

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 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. Pada penelitian selanjutnya yang diterapkan oleh Vie, J. J. et al., 2017 yang menerapkan metode baru yang mereka beri nama BALSE (Blended Alternate Least Squares with Explanation) yang merupakan kombinasi dari beberapa metode, pada penelitian tersebut mereka melakukan ekstrasi fitur pada poster anime dan manga dalam merekomendasikan anime. Komponen dari BALSE adalah Illustration2Vec, ALS (Alternate Least Squares), LASSO (Least Absolute Shrinkage and Selection Operator) serta *Steins gate* yang merupakan metode untuk mengkombinasikan hasil dari dua metode ALS dan LASSO. Dari penelitian tersebut dihasilkan nilai RMSE sebesar 1.4954±0.004 menghasilkan kesimpulan bahwa prediksi BALSE lebih baik dibandingkan ALS.

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. Pada penelitian lainnya yang diteliti oleh Lund, J et al., 2018 meneliti Rekomendasi Movie menggunakan Deep Learning dengan metode autoencoder menghasilkan MAE dan RMSE sebesar 0.15 dan 0.35 selain melakukan menggunakan teknik evaluasi tersebut, penelitian tersebut juga menggunakan evaluasi dengan pengguna langsung dengan jumlah 100 orang partisipan menghasilkan survey sebesar 71.67% partisipan lebih memilih menggunakan hasil penelitian tersebut. Kemudian ada juga penelitian yang dilakukan oleh Soni, B et al., 2021 menerapakan sistem rekomendasi anime *hybrid recommendation filtering* yang menggunakan algoritma autoencoder dan clustering spectral menghasilkan RMSE sebesar 0.591 dan 0.349, Penelitian tersebut telah melampaui *state-of-art* metode pada saat itu.

Penelitian lainnya yang menerapkan sistem rekomendasi menggunakan metode BERT telah diteliti oleh Jeong, C et al., 2020 menerapkan *context-aware recommendation* menggunakan metode BERT dan GCN (*Graph Convolutional Network)*, penelitian tersebut menggunakan dataset baru bernama FullTextPeerRead yang berisikan konteks kalimat ke referensi sitasi dan metadata paper yang menunjukkan hasil yang cukup baik dibandingkan dengan metode yang lain dengan menggunakan beberapa evaluasi yaitu MAP sebesar 0.6189, MPR sebesar sebesar 0.6036, Recall@5 sebesar sebesar 0.6736, Recall@10 sebesar sebesar 0.7109 dan Recall@30 sebesar 0.7814. Peneltian lainnya oleh Wang, T., & Fu, Y. (2020) yang menerapkan BERT pada *collaborative item-based* penelitian ini menggunakan dataset *e-commerce* membuktikan bahwa metode BERT berhasil melampaui metode BiLSTM dengan hasil precision@1 sebesar 0.555, precision@10 sebesar 0.079, Recall@10 sebesar 0.791 dan NDCG@10 sebesar 0.669.

Pada penelitian ini juga akan menerapkan metode *attention* pada metode *recurrent neural network* sebelumnya penelitian yang menerapkan metode *attention* seperti pada penelitian Wu, F et al., 2020 yang menggunakan dataset *news recommendation* yang menerapkan metode *attention* yang membuktikan bahwa metode tersebut akan menghasilkan performa yang lebih baik jika dikombinasikan dengan metode *deep learning* lainnya seperti LSTM atau CNN. Salah satu *state-of-art* yang menerapkan *attention* adalah 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.*

Metode lainnya yang pada sistem rekomendasi…..

Sistem Rekomendasi hadir untuk mempermudah dalam pemilihan judul anime yang sesuai. Sistem rekomendasi sendiri umumnya memiliki tiga teknik yaitu *collaborative filtering*, *content-based filtering*, dan *hybrid filtering*. Pada penerapannya teknik *collaborative filtering* menggunakan data user lain dalam menerapkan sistem rekomendasinya yang berakibat pada permasalahan cold-start kemudian untuk *content-based filtering* menggunakan data *meta* dari item itu sendiri

yaitu menggunakan kemiripan antar satu konten dengan yang lainnya pada penerapannya teknik ini juga memiliki kelemahan salah satunya adalah membutuhkan rating yang cukup sebelum sistem rekomendasi dapat memberikan hasil rekomendasi akurat yang sesuai preferensi user tersebut, yang kemungkinan jika hanya terdapat rating ketika user tersebut baru terdaftar di dalam sistem maka sistem belum bisa memberikan hasil rekomendasi yang sesuai (Lops, P et al., 2011) berdasarkan dari salah satu kelemahan tersebut hadirlah teknik *hybrid filtering* yang bertujuan untuk mengurangi kelemahan dari kedua teknik tersebut.

Penelitian ini akan menerapkan teknik *hybrid filtering* dengan kombinasi antara *content-based filtering* dan *collaborative* filtering dengan menggunakan hybrid filtering kelemahan pada salah satu tipe sistem rekomendasi akan teratasi. Salah satunya *Cold-start* bisa diatasi dengan penggunaan *content-based filtering* (Wang, H et al., 2020). Dengan menggunakan deep learning model rekomendasi bisa merepresentasikan relasi antar user dan item dengan mempelajari pada *deep-leavel* jaringan struktur non-linear (Wang, W et al., 2020). Pada penelitian ini akan mengunakan metode *Recurrent Neural Network* telah dibuktikan dari beberapa penelitian yang telah dipaparkan sebelumnya bahwa dengan menggunakan metode *recurrent neural network* sistem rekomendasi yang dihasilkan bisa menghasilkan hasil yang dibutuhkan, pada penelitian ini juga akan menggunakan *attention* yang mana *attention* merupakan metode yang mampu mengolah struktur kalimat *complex* (Kang, W. C., & McAuley, J., 2018). Kemudian penelitian ini akan mencoba menerapkan BERT4REC sebagai perbandingannya untuk membuktikan yang mana yang merupakan metode yang lebih baik diantara keduanya. Alasan menggunakan metode BERT adalah metode tersebut merupakan metode state-of-art yang terbaru yang diteliti oleh Fei Sun et al., (2018).

Data yang akan digunakan bersumber dari dataset yang ada yang berasal dari situs *kaggle* selain itu dataset yang akan digunakan tidak hanya satu tipe dataset. 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 sebagai berikut :

1. Metode sistem rekomendasi mana yang terbaik dari metode yang diperbandingkan yaitu BERT dan BiLSTM dengan *attention* pada pemilihan film anime
2. Bagaimana hasil metode sistem rekomendasi yang diterapkan yaitu BERT dan BiLSTM dengan attention
3. Menerapkan sistem rekomendasi pada text judul anime menggunakan metode Bidirectional Encoder Representations From Transformers

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.
3. Data penelitian ini berformat .csv
4. Sistem rekomendasi pada penelitian ini menggunakan data masukan yang terbatas pada judul anime, genre, type, episode, score, studio, favorites, rating, premiered, sinopsis dan members
5. Pada penelitian data keluaran yang digunakan akan sama dengan seperti data masukan.
6. Data anime yang akan digunakan pada penelitian ini merupakan anime yang tayang sejak tahun 1979 hingga 2018.

1.4 Tujuan Penelitian

Penelitian ini memiliki tujuan, sebagai berikut :

1. Mengetahui pengaruh metode attention based pada BiLSTM dalam memberikan hasil rekomendasi.
2. Mengetahui tingkat akurasi penerapan metode attention based pada BiLSTM dan BERT terhadap hasil rekomendasi yang diberikan
3. Mengetahui perbandingan akurasi dari BiLSTM dengan *attention* dan BERT pada sistem rekomendasi

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah berusaha untuk 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 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).

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

6.1 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.

6.2 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.

6.2.1 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).

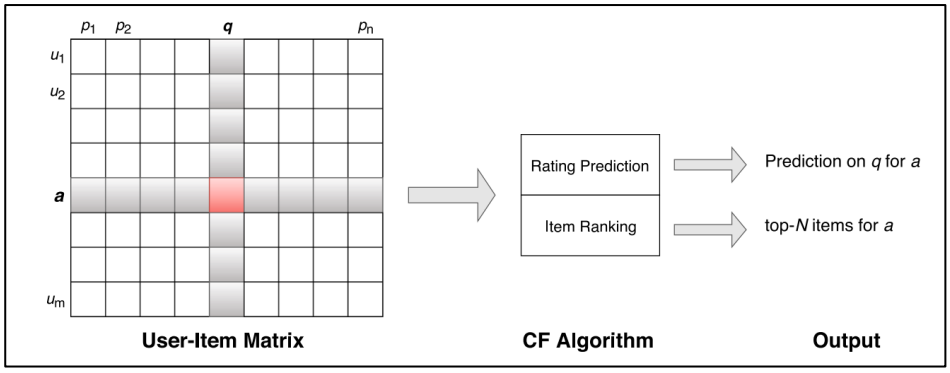
Keuntungan dengan menggunakan teknik content-based yaitu mampu memberikan rekomendasi item baru meskipun user tidak memberikan rating kepada item - item yang dituju oleh user tersebut, jadi meskipun database tidak menyediakan data user preferensi hal itu tidak mempengaruhi tingkat keakurasian sistem 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)

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

A. 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.

2. *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.

B. 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.

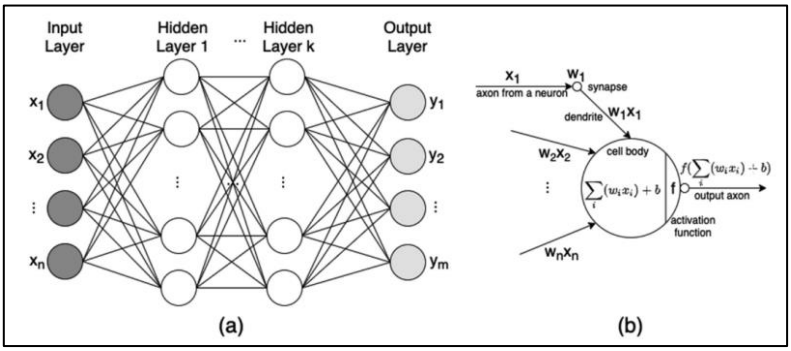
6.2.3 Hybrid Filtering

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.

6.3 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

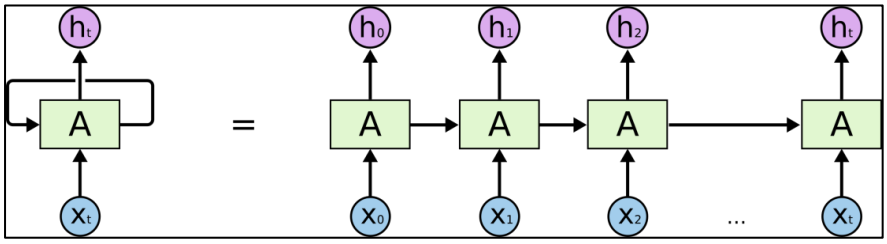
(1) E*nd to End Differentiable* maksudnya adalah umumnya pada metode tradisional penelitian dilaksanakan pada beberapa tahap dengan menggunakan *deep learning* hal tersebut bisa diperpendek dengan menjadikannya hanya dengan satu tahapan yaitu dengan hanya menggunakan sebuah *neural network.*

(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.

6.4 Recurrent Neural Network

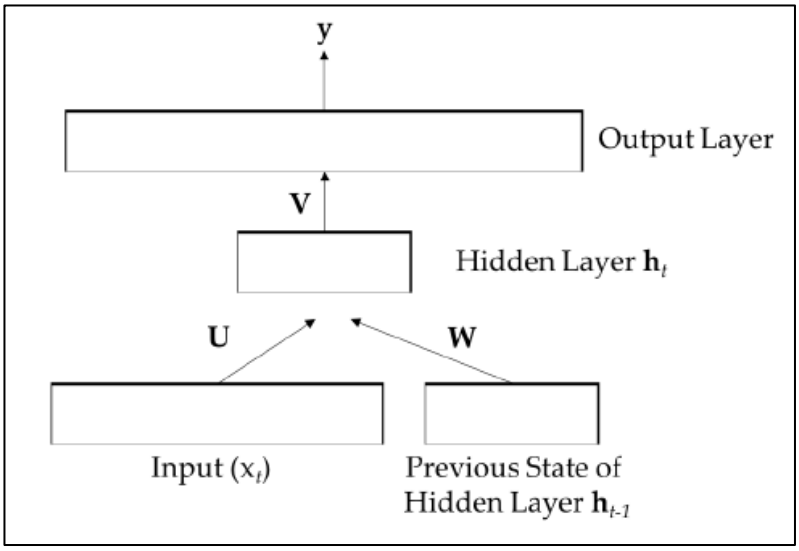
Recurrent Neural Network (RNN) merupakan salah satu jenis arsitektur jaringan syaraf tiruan, RNN sendiri merupakan anggota keluarga dari *neural network* untuk menangani data yang berkelanjutan atau bersambung (*sequential data)* (Goodfellow, Ian et al., 2016). Berbeda dengan CNN yang digunakan secara 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 x = $(x_1, \ldots, x_t)$. Data x_t yang bisa merupakan data vektor, gambar, teks ataupun suara dipengaruhi oleh data sebelumnya yang bisa ditulis sebagai $P(x_t | \{x_1, \ldots, x_{t-1}\})$ (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* :



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 :

$h_t = f(x_t, h_{t-1}, b)$…….(2.1)

Keterangan :

h_t : *hidden state* ke-t

f : Fungsi *activation*

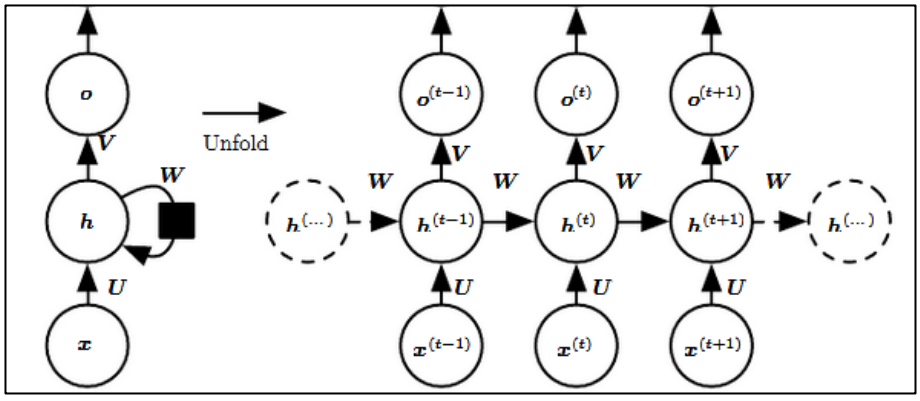
x_t : Nilai *Input* ke-t

h_{t-1} : 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 :

$$h_t = f(x_t, h_{t-1}, b)$$
$$h_t = f(x_t f(x_{t-1}, h_{t-2}, b), b)$$
$$h_t = f(x_t, f(x_{t-1}, f(\{x_t, \ldots, x_{t-2}\}, \{h_1, \ldots, h_{t-3}\}, b), b), b)$$



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.*

- Ou

tput: *o(t)* merupakan nilai keluaran dari proses yang telah dilakukan oleh RNN pada langkah ke t. Pada keluarna ini jika dirumuskan menjadi $o_t = softmax(V_{h_t})$

6.4.1 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 :

$a^t = b + Wh^{t-1} + Ux^t$ ……(2.3)
$h^t = \tanh(a^t)$……(2.4)
$o^t = c + Vh^t$……(2.5)
$y^t = softmax(o^t)$……..(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).

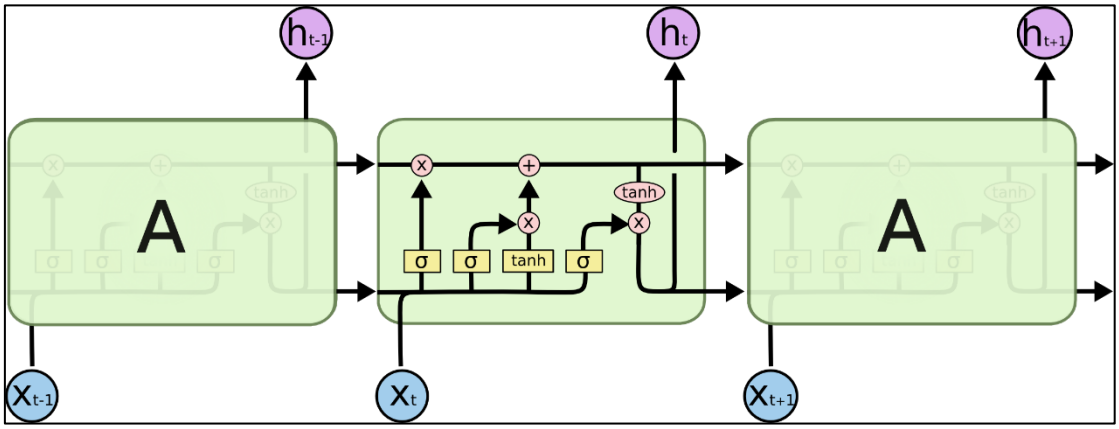
6.4.2 Backpropagation Trough Time

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.

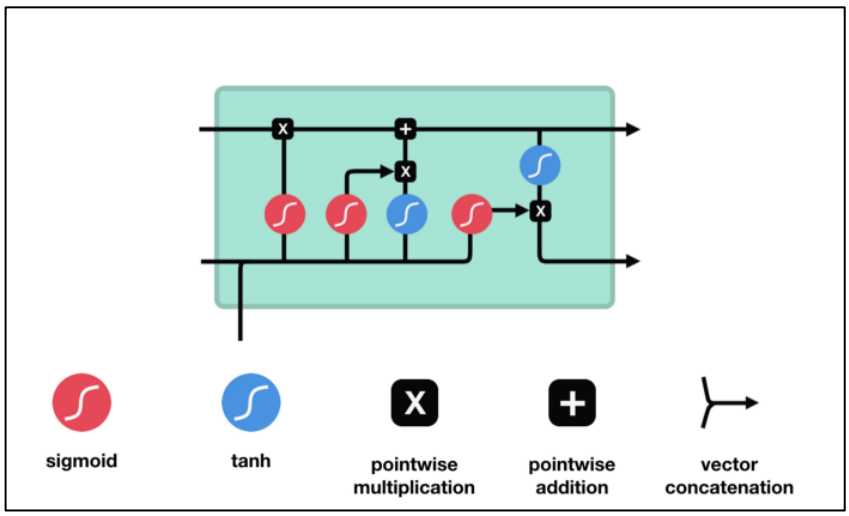
Pada backpropagation through time untuk memperbaharui parameter saat telah mencapai hidden state paling awal berbeda dengan *backpropagation* yang pada umumnya memperbaharui parameter bersamaan dengan mempropagasi error dari hidden state ke hidden state sebelumnya.. RNN merupakan salah satu neural network 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.

6.5 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).



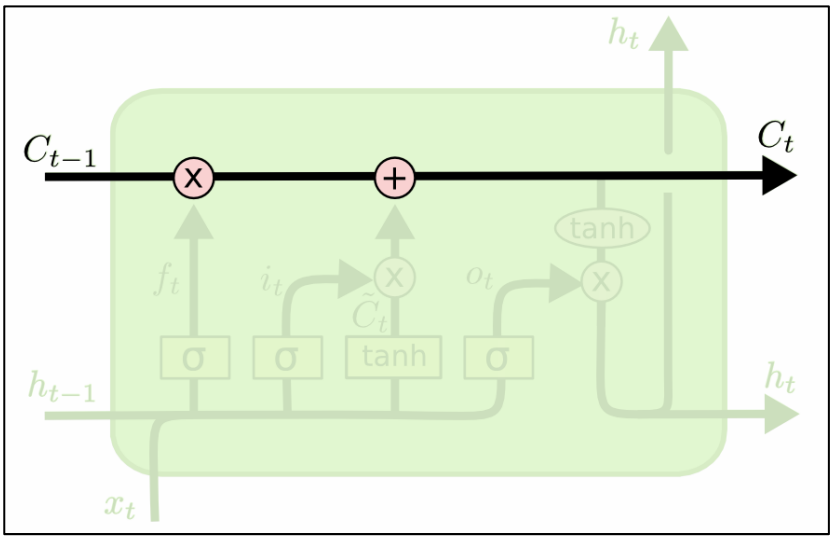
Gambar 2.6 Diagram arsitektur LSTM



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.



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):

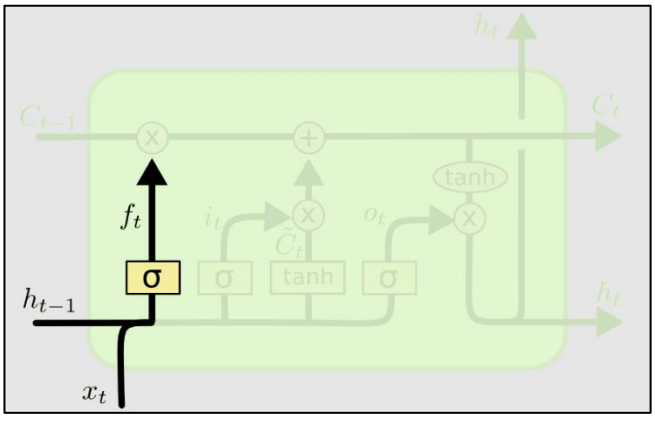
1. Tahap pertama, Proses akan dimulai dari forget gate, pada tahap ini akan ditentukan informasi mana saja yang akan dipertahankan atau dihapus. Pada tahapan ini berdasarkan dari *output* hidden state sebelumnya dan input akan diproses menggunakan fungsi aktivasi sigmoid. Output yang akan dihasilkan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

$$\boldsymbol{ft = \sigma\,(x_t \cdot W_f + h_{t-1} \cdot U_f + b_f)} \quad \text{.....(2.7)}$$

Keterangan :

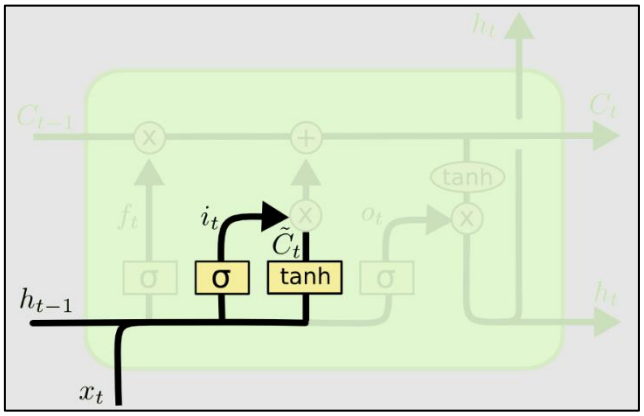
ft : *forget gate*
σ : fungsi sigmoid
x_t : masukan order pada langkah ke-t
h_{t-1} : keluaran pada langkah yang sebelumnya
W_f : bobot pada *forget gate*
U_f : *reccurent weight* pada *forget gate*
b_f : bias *forget gate*

Berikut merupakan alur proses pada *forget gate* :



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 :



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 :

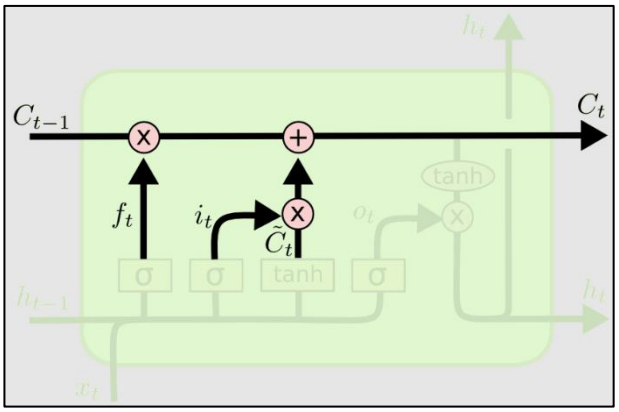
$$\boldsymbol{i_t = \sigma\,(x_t \cdot W_i + h_{t-1} \cdot U_i + b_i)} \text{.....,,,,.....(2.8)}$$

$$\boldsymbol{\check{C}_t = \tanh\,(x_t \cdot W_c + h_{t-1} \cdot U_c + b_c)} \text{.....,,,,.....(2.9)}$$

Keterangan :

i_t : *input gate*
$\check{C}_t$: calon konteks
σ : fungsi *sigmoid*
W_i : *weight* pada *input gate*
U_i : *reccurent weight* pada *input gate*
b_i : nilai pada *input gate*
W_c : *weight* pada calon konteks
U_c : *reccurent weight* pada calon konteks
b_c : nilai bias pada calon konteks
x_t : nilai *input* pada orde ke-t
h_{t-1} : nilai *output* sebelum order ke-t

7. Pada tahap ini dilakukan untuk memperbaharui cell state dari C_{t-1} menjadi C_t.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* (f_t) dengan *cell state* lama (C_{t-1}) dikalikan. Dari kedua hasil perkalian itu dijumlahkan, dari penjumlahan ini menghasilkan *cell state* baru (C_t). Berikut ilustrasi yang menggambarkan proses pada tahap ini serta formula yang menyertakan proses ini dijelaskan pada persamaan 2.10 :



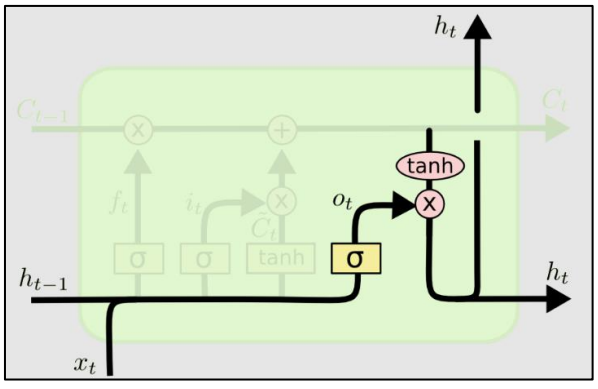
Gambar 2.10 Cell State

$$\boldsymbol{C_t = f_t \times C_{t-1} + i_t \times \breve{C}_t} \quad \text{.....(2.10)}$$

Keterangan :

C_t : *cell state* saat ini
f_t : *forget gate*
C_{t-1} : *cell state* sebelumnya
i_t : nilai *input* pada order ke – t
$\breve{C}_t$: calon konteks

8. Pada tahap akhir ini adalah menghasilkan *output* akhir dari serangkaian proses sebelumnya yaitu *hidden state* selanjutnya (h_t). 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 (h_{t-1}) dan *input* (x_t) untuk mengetahui *ouput gate* (O_t). Hasil *output gate* akan menghasilkan nilai di antara 0 sama dengan 1. Tahapan selanjutnya adalah menentukan hasil *hidden state* atau *output* akhir (h_t), nilai dari *output gate* akan dikalikan dengan nilai *cell state* (C_t) yang diubah menggunakan fungsi *tanh.* Berikut merupakan ilustrasi pada proses tahapan ini yang disertakan dengan pada persamaan 2.11 dan 2.12 :



Gambar 2.11 Output Gate

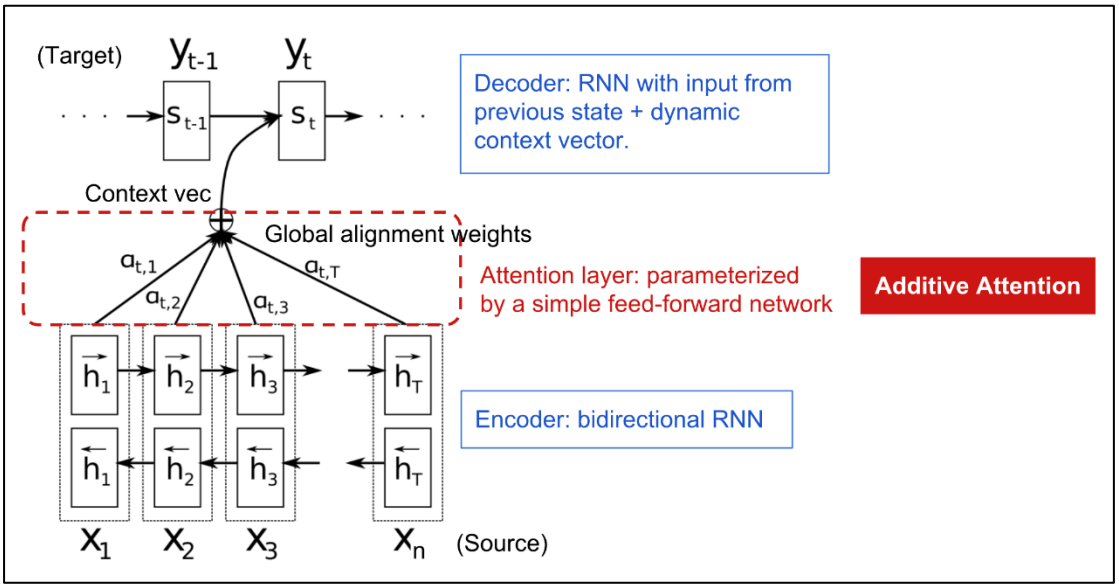
$O_t = \sigma(x_t \times W_o + h_{t-1} \times U_o + b_0)$……(2.11)
$h_t = O_t \times \tanh(C_t)$……(2.12)

Keterangan :
O_t : *output gate*
σ : fungsi sigmoid
x_t : nilai input pada orde ke - t
W_o : nilai *weight* pada *output gate*
U_o : *reccurent weight* pada *output gate*
h_{t-1} : nilai *output* pada order ke – t-1
b_0 : nilai bias pada *output gate*
C_t : *cell state*

2.6 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 keterhubungan 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). Atensi mekanisme memiliki yang konsep 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). Berikut ilustrasi dari attention layer :



Gambar 2.12 Attention Layer pada Seq2Seq model (Bahdanau et al., 2015)

Proses yang terjadi pada *attention layer* sebagai contoh mula – mula terdapat n *sequence* dan keluaran *y* dengan m *sequence* pada sebuah *network.*

x = [x1, x2,…, xn]……...(2.13)
y = [y1, y2,…, ym]……..(2.14)

Pada penelitian ini akan menggunakan jaringan *bidirectional* LSTM yang mana memiliki forward hidden state dan juga backward hidden state dengan BiLSTM yang memiliki forward hidden state $\overrightarrow{h_i}$ dan backward hidden state $\overleftarrow{h_i}$ dengan begitu *encoder state* bisa dicapai dengan menggabungkan forward dan backward states. Berikut contoh bentuknya :

$$\boldsymbol{h}_i = [\overrightarrow{\boldsymbol{h}}_i^{\top}; \overleftarrow{\boldsymbol{h}}_i^{\top}]^{\top}, i = 1, \ldots, n$$

Gambar 2.13 Encoder state

Selanjutnya untuk bagian *decoder network, hidden state*nya adalah sebagai berikut :

$$a_{t,i} = \frac{\exp(score(s_{t-1}, h_i))}{\sum_{i=1}^{n} \exp(score(s_{t-1}, h_i))} \quad \text{……...(2.15)}$$

$$c_t = \sum_{i=1}^{n} a_{t,i} h_i \text{……...(2.16)}$$

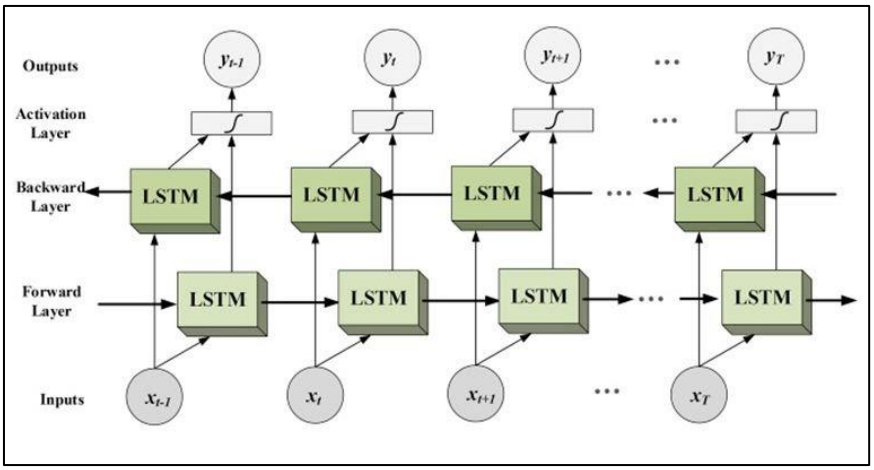
$$s_t = f(s_{t-1}, y_{t-1}, c_t) \text{……..(2.17)}$$

Keterangan :
$a_{t,i}$: *Alignment Score*
c_t : *Context vector* untuk keluaran y_t
s_t : *Hidden State*

Berdasarkan dari formula diatas bisa dikatakan bahwa aligment scores merupakan bobot yang bertanggung jawab dalam mendefiniskan seberapa banyak dari masing - masing sumber hidden state yang bisa dipertimbangkan pada setiap output yang ingin dihasilkan.

2.7 Bidirectional Long-Short Term Memory

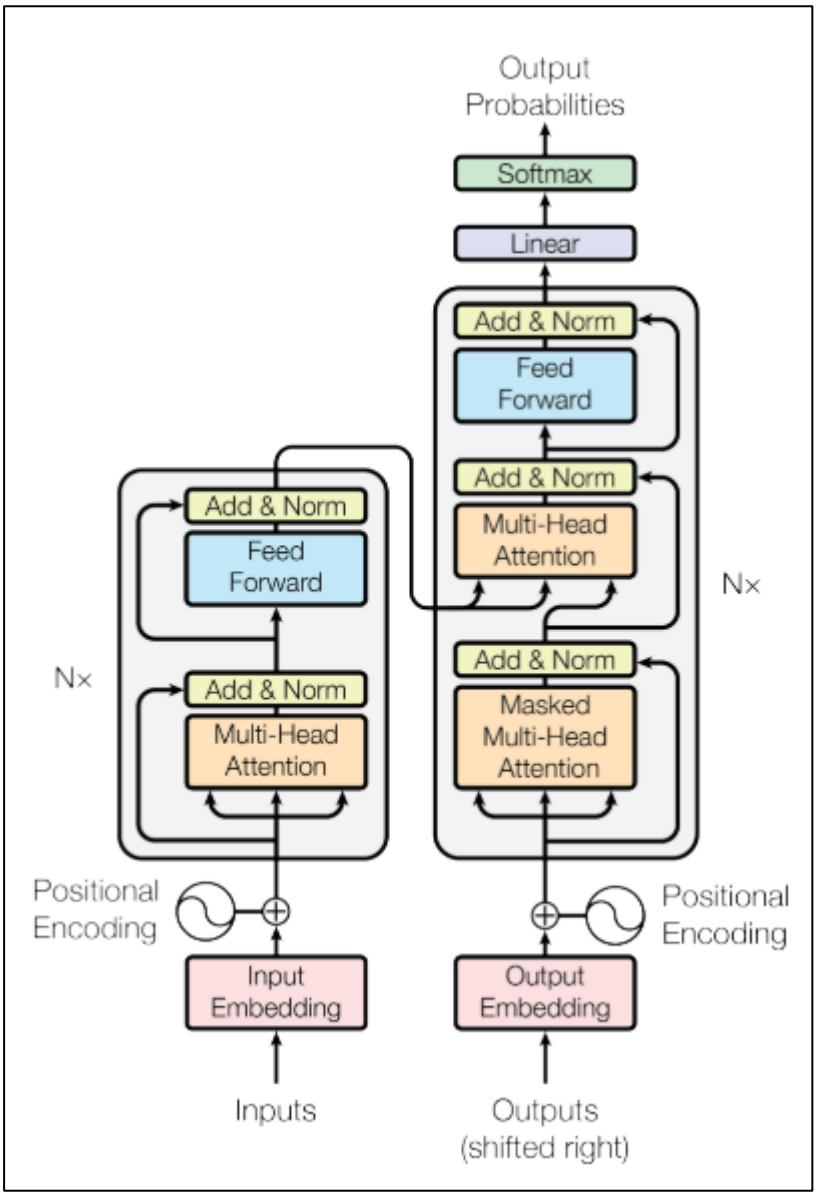
Bidirectional Long Short-Term Memory atau disingkat dengan BiLSTM merupakan model LSTM yang telah dikembangkan yang memiliki masukan dengan proses dengan dua arah yaitu secara *forward* dan *backward.*



Gambar 2.14 Struktur BiLSTM

2.8 Transformers

Transformers adalah model arsitektur yang dibuat berdasarkan atas konsep *attention mechanism*. Transformers merupakan model *tranduction* pertama yang sepenuhnya menggunakan self-attention untuk mengolah representasi *input* dan *output* tanpa menggunakan *sequence-aligned* RNN ataupun *Convolution* (Vaswani, A., et al., 2017). Transformer merupakan model *state-of-art* pada bidang *Natural Languange Processing* yang menggantikan model yang paling umum digunakan pada arsitektur *encoder-decoder* dengan menggunakan *multi-headed self-attention*. Transformer adalah model deep learning yang mana setiap keluaran saling terhubung dengan setiap elemen dan pembobotannya diperhitungkan secara dinamis berdasarkan atas hubungan antar elemennya. Sesuai ilustrasi di bawah Transformers tersusun oleh encoder dan decoder. Dari kedua struktur tersebut secara garis besar Transformers memiliki tiga hal penting yang menyusunnya yaitu *self attention, feed-forward neural network* dan *positional encoding.* Berikut ilustrasi yang menggambarkan arsitektur Transformers :



Gambar 2.15 Transformers Arsitektur (Vaswani, A., et al 2017)

Berdasarkan arsitektur di terdapat beberapa *term* yang penulis jelaskan yaitu sebagai berikut :

2. 8.1 Positional Encoding

Positional Encoding digunakan untuk menggantikan *reccurence* atau *convolutional* yang bertujuan untuk mendapatkan informasi secara relatif atau jelas mengenai posisi dari tokens yang terdapat pada *sequence* (Vaswani, A., et al 2017). Positional encoding terletak sebelum encoder stacks dan setelah decoder stacks. Postional encoding memiliki ukuran dimensi yang sama dengan embedding sehingga embedding dan positional encoding bisa dijumlahkan dengan embedding. Berikut formula Positional Encoding yang digunakan oleh Vaswani, A et al., 2017

PE(pos,2i) = $\sin(pos/10000^{2i/d^{model}})$.....(2.19)

PE(pos,2i+1) = cos(pos/$10000^{2i/d^{model}}$)……..(2.20)

Keterangan :

PE : Positional Encoding
d^{model} : dimensional embedding

Dengan formula diatas setiap input vector dengan indeks ganjil, menggunakan fungsi cos sedangkan item input dengan indeks genap emnggunakan fungsi sin, yang kemudian jumlahkan dengan embedding.

2.8.2 Encoder Stacks

Encoder stack yang berada di posisi kiri dari arsitektur transformer di gambar 2.15 bertujuan untuk memetakan semua masukan *sequences* ke bentuk representasi berkelanjutan *sequences* yang akan digunakan dilanjutkan ke *decoder stacks*. Encode layer tersusun oleh dua sub modules yaitu *multi-head attention* dan *feed forward network* serta terdapat juga *residual connections* di masing - masing sub layer tersebut juga terdapat layer *normalization*. Kenyataannya *encoder stacks* tersusun oleh 6 stack identical layers arsitektur di atas merupakan contoh dari single stack dari transformers (Vaswani, A., et al 2017). Keluaran dari masing - masing sub-layer adalah $LayerNorm(x + SubLayer(x))$, di mana $SubLayer(x)$ merupakan fungsi yang diimplementasikan oleh sublayer itu sendiri. Dalam rangka membantu residual connections, semua sublayer yang ada di model ini termasuk embedding layer menghasilkan keluar dimension d_{model} sebesar 512.

2.8.3 Decoder Stacks

Decoder stacks yang berada di sebelah kanan di arsitektur transformer pada gambar 2.15 bertujuan untuk menerima keluaran dari encoder yang secara berbarengan menghasilkan keluaran yang telah diproses sebelumnya untuk menghasilkan keluaran sequence. Decoder stacks memiliki struktur yang mirip dengan encoder stacks, perbedaan yang dimiliki adalah pada decoder memiliki sub-layer tambahan ketiga di atas positional encodieng sebelum output embedding. Pada decoder ini ditambahkan modifikasi self-attention pada sub-layer untuk mencegah posisi dari memperhatikan ke posisi selanjutnya (Vaswani, A., et al 2017). Sub-layer ini meskipun memiliki kemiripan dengan yang ada di encoder tetapi masing - masing dari layer multi-head attention ini memiliki fungsi yang berbeda - beda.

2.8.4 Multi-head Attention

Mult-head attention merupakan module yang digunakan oleh attention mechanism yang mana menjalankan attention mechanism secara parallel. Pada

Transformers ini Multi-head mechanism menggunakan *self-attention*. Multi-head attention memungkinkan model untuk bersama-sama memperhatikan informasi dari represetansi sub ruang yang berbeda pada posisi yang berbeda (Vaswani, A., et al 2017). Secara intuitive, multiple-atttention memungkinkan untuk memperhatikan ke bagian bagian pada sequence yang berbeda. Berikut formula matematis *multi-head attention* :

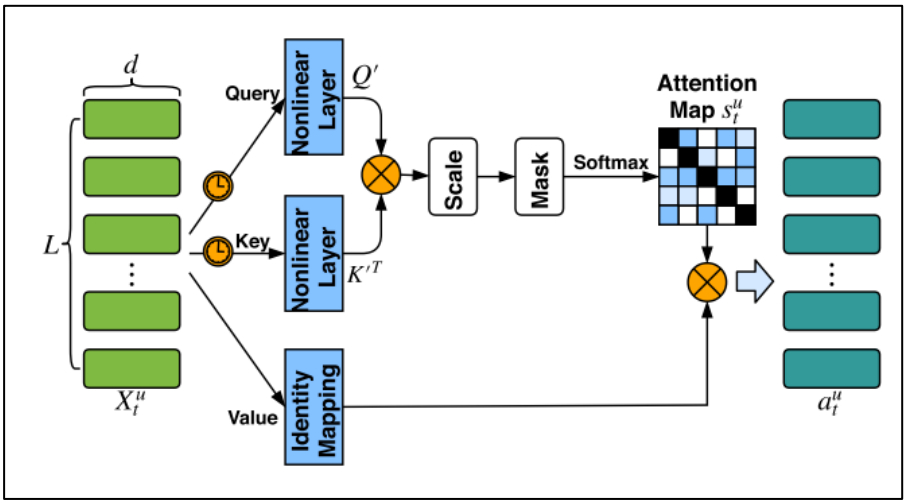
MultiHead(Q,K, V) = Concat($head_1$, ..., $head_h$) W^o ……(2.21)
$head_i$ = Attention(QW_i^Q ,KW_i^k , VW_i^v)……(2.22)

Keterangan :

Q : Query
W : Bobot
K : Key
V : Value

2.8.6 Self Attention

Self-attention merupakan model attention yang juga memiliki sebutan sebagai intra-attention adalah model yang memungkinkan masukan untuk berinteraksi satu sama lain dengan masukan itu sendiri yang bertujuan untuk mencari siapa yang harus diberikan perhatian lebih. Self-attention memilki kemampuan lebih baik dibanding pendahulunya seperti RNN dan CNN dan bahkan bisa menggantikan model - model tersebut pada bidang sequence learning, kelebihannya memiliki akurasi yang lebih baik dengan komputasi kompleksitas yang lebih rendah (Zhang, S et al., 2018).



Gambar 2.16 Ilustrasi Modul Self-Attention (Zhang, S et al., 2018).

Tidak seperti attention yang pada umumnya mempelajari representas dengan batasan informasi pada keseluruhan konteks, self-attention mampu mempertahankan konteks informasi secara sequential dan melihat hubungan antar elemen yang ada di

sequence, terlepas dari panjangnya sequence tersebut. Susunan dari yang menyusun self-attention adalah scaled dot-product attention. Masukan yang diterima oleh self-attention tersusun oleh query, key dan value, sedangkan keluaran dari outputnya adalah jumlah bobot dari value dimana bobot matriks atau penghubung matriks ditentukan oleh query dan key yang sesuai.

Self-attention menerima masukan matriks X $\in R^{T\times D_{in}}$ dengan tersusun dengan sejumlah T token dengan dimensi sebesar D_{in}. Dengan formula self-attention sebgai berikut (Cordonnier, J.-B., et al., 2019) :

Self-Attention$(X)_t$,: := softmax (A_t,:)XW_v……....(2.23)
A := X$W_q W_k^T X^T$ ……...….(2.24)

Keterangan :

X : Matriks input
W_q : Bobot query
W_k^T : Bobot key

2.8.6 Feed Forward Networks

Feed Forward Networks terdiri dari dua fully connected layer. Implementasi Feed-forward network diterapkan pada masing-masing posisi secara terpisah dan identik. Jumlah dari dimensi di hidden layer feed forward network ini secara umum adalah sekitar empat kali lipat dari token embedding dmodel, jadi terkadang feed forward network disebut dengan expand-and-contract network. Sedangkan transformasi linear diterapakan secara sama di berbagai posisi yang berbeda, transformasi linear menggunakan parameter yang berbeda dari layer satu ke layer lain. Cara lain dalam mendeskripsikan hal ini mengganggap dua konvolusi dengan ukuran kernel 1. Dimensi input dan output d_{model} = 512 dan inner layer memiliki dimensi sebesar d_{ff} = 2048

FFN(x) = max(0, $xW_1 + b_1) W_2 + b_2$ ……(2.25)

Keterangan :

Max : fungsi ReLu
W : bobot
b : bias

2.9 Bidirectional Encoder Representations from Transformers

BERT atau Bidirectional Encoder Representations from Transformers merupakan algoritma yang bertujuan untuk mempelajari representasi kata melalui dua tahapan tugas. Tugas pertama adalah memprediksi kata - kata yang hilang yang terdapat pada kalimat, berdasarkan sisa - sisa katanya yang terdapat pada kalimat tersebut. Tugas

Kedua bertugas untuk menentukan apakah kalimat kedua dari dua kalimat adalah kalimat lanjutan atau apakah itu berasal dari teks yang sama sekali tidak berhubungan. BERT merupakan model algoritma yang dibuat oleh Google. BERT digunakan untuk membuat model state-of-art pada berbagai tugas. BERT dipublikasikan pada tahun 2018 oleh Jacob Devlin beserta peneliti lainnnya dari Google kemudian pada tahun 2019 dipublikasikan secara luas.

Umumnya untuk melakukan *pre-trained* representasi bahasa terdiri dari dua cara yaitu pre-training dan fine-tuning (Devlin, J., et al., 2019) Pada kedua cara tersebut terdapat kelemahan seperti hanya mengenalkan tugas spesifik yang minim parameter dan dilatih pada tugas - tugas tahap akhir dengan sederhana menggunakan fine-tuning pada paramter yang telah dilatih. Dengan kelemahan seperti itu akan membatasi kemampuan dari represtasi pre-trained khususnya pada pendekatan fine-tuning, selain itu batasan utamanya adalah pada model bahasa standar hanya berjalan secara searah dan terbatasnya pada pilihan arsitektur selama pre-training, sehingga dari hal itu diciptakanlah BERT dengan cara pendekatan berdasarkan fine-tuning, BERT menggunakan *masked language model* serta *next sentence prediction* yang kemudian berhasil menjadikan BERT sebagai *state-of-art* pada bidang NLP.

Arsitektur BERT terdiri dari multi-layer bidirectional Transformer encoder yang berdasarkan dari penelitian Transformer oleh Vaswani., et al (2017). BERT menggunakan Transformers yang tersusun oleh *bidirectional self-attention.* Terdapat dua ukuran model BERT yaitu BERT Base yang terdiri dari 12 layers (transformer blocks), 12 attention heads, dan 110 juta parameters dan BERT Large yang terdiri dari 24 layers (transformer blocks), 16 attention heads dan 340 juta parameters.

Terdapat dua tipe cara dalam menggunakan BERT yaitu pre-training dan fine-tuning. Pada saat proses pre-training model BERT melakukan pelatihan pada data yang belum dilabeli yang bertujuan untuk menyelesaikan berbagai tugas pre-training yang berbeda. Sedangkan untuk proses fine-tuning, model BERT menggunakan inisialisasi dengan parameter yang sebelumnya sudah dilatih dan semua parameter yang telah di fine-tuned menggunakan label data yang berasal dari permasalahan dari tugas yang ingin diselesaikan, masing-masing dari permasalahan tugas (downstream task) terpisah sesuai model fine-tuned-nya, meskipun mereka menggunakan inisialisasi parameter yang sama.

Secara spesifik pada strategi fine-tuning terdapat dua strategi untuk menggunakan BERT yaitu *masked languange model* merupakan model yang bertugas dalam memprediksi kata yang semestinya terdapat pada kalimat, dan strategi selanjutnya adalah *Next sentence prediction* ialah model pre-training yang bisa memprediksi kalimat lanjutan dari sebuah kalimat.

2.10 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 :

$$CG_{@K} = \sum_{i=1}^{K} G_i \quad \text{.....(2.26)}$$

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 :

$$DCG_{@K} = \sum_{i=1}^{K} \frac{G_i}{\log_2(i+1)} \quad \text{.....(2.27)}$$

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 :

$$nDCG_{@K} = \frac{DCG_{@K}}{IDCG_{@K}} \quad \text{.....(2.28)}$$

$$IDCG_{@K} = \sum_{i=1}^{K_{ideal}} \frac{2^{G_i}-1}{\log_2(i+1)} \quad \text{.....(2.29)}$$

Keterangan :

G : Gradien
K : Posisi rank item
I : posisi indeks G

2.10 Root Mean Square Error

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	Vie, J. J et al. (2018)	CNN, IllustrationN2Vec, ALS^2, LASSO^3 serta Steins gate.	Mangaki dataset	Evaluasi menggunakan RMSE sebesar 1.14954 ± 0.004
3	Nuurshadieq, & Wibowo, A. T. (2020).	Long short-term Memory	MyAnimeList dataset	Evaluasi menggunakan RMSE 1.4475
4	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%
5	Wang, W., et al. (2020).	LSTM dan CNN	MovieLens dataset	Menghasilkan MSE sebesar 0,876 DAN MAE 0,751
6	Khezrian, N et al. (2020).	Bidirectional Encoder Representations from Transformers (BERT)	freecode datasets	F1-Score@10 sebesar 46.5, Precision@5 TagBERT 41.83, Precision@10 sebesar 40.25 dan recall@10 sebesar 64.42
7	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
8	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.
9	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
10	Kaviani, M., & Rahmani, H. (2020).	BERT	Twitter dataset	Menghasilkan Precision 15.18%, Recall 46.12% dan F_measeure 22.34%
11	Pratiwi, R. W et al. (2020)	Attention-Based BiLSTM	Twitter dataset	Accuracy 79.68%, Precision 78.37%, Recall 79.26% dan F1 Score 78%
12	Zhuang, Y., & Kim, J. (2021).	BERT	TripAdvisor dataset	Rekomendasi Multi-criteria menunjukkan hasil evaluasi sebagai berikut HR@5 sebesar 0.333, HR@10 sebesar 0.25, HR@15 sebesar 0.217, NDCG@5 sebesar 0.694, NDCG@10 sebesar 0.606, NDCG@15 sebesar 0.569
13	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.
14	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

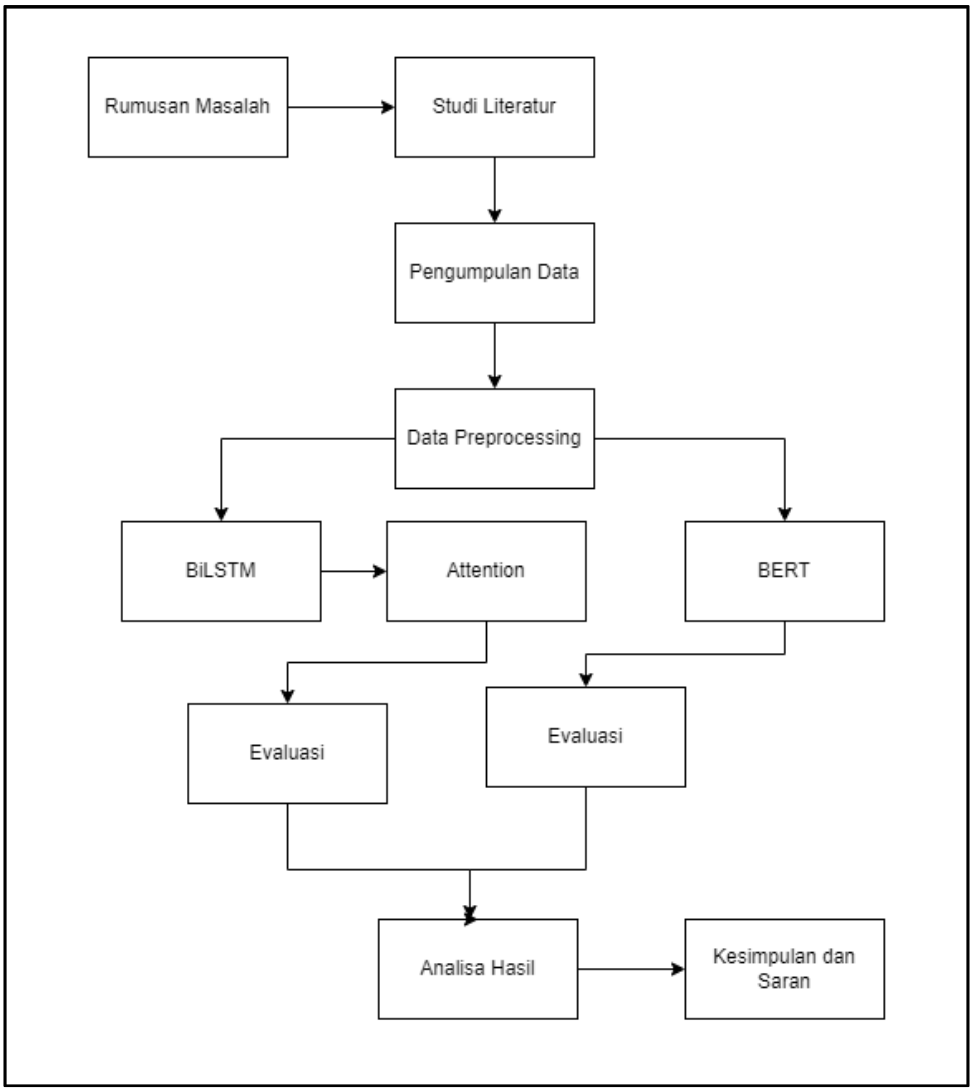
15	Rosa, R. L, et al (2019)	CNN, Bi-LSTM-RNN menggunakan SoftMax	Online Social Network datasets (facebook)	Menghasilkan accuracy sebesar 0.89 dan 0.90 dalam mendeteksi emosi depresi dan stress user, selain itu hasil dari rekomendasi sistem mencapai 94% user yang sangat puas, dibandingkan dengan sistem rekomendasi yang tidak menggunakan sentimen analysis yang hanya mencapai tingkat kepuasan 69%
16	Zhang, X., et al (2019)	Movie4Vec dan Bert4Movie	Hulu movies Series dataset	Menunjukkan hasil leaderboard sebesar 0.64754, hal ini menunjukkan performa melampaui benchmark.

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

3.1 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 :



Gambar 3.1 Tahapan Penelitian

3.1.1 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.

3.1.2 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.

3.1.3 Pre-processing

3.2 Pembuatan Model

B

3.2.1 BiLSTM

3.2.2 Attention

d

3.2.3 BERT

e

3.3 Pengujian Sistem

f

3.4 Metodologi Pengembangan Sistem

f

3.4.1 Analisis Kebutuhan Sistem

g

3.4.2 Kebutuhan Fungsional

a

3.4.3 Kebutuhan Non-Fungsional

3.4.4 Proses Desain

3.4.5 Perancangan Sistem

3.4.6 Perancangan Pengujian

Daftar Pustaka

Ricci, F., Rokach, L., Shapira, B., & Kantor, P. B,. (2011). Recommender Systems Handbook. In Recommender Systems Handbook. https://doi.org/10.1007/978-0-387-85820-3

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. https://doi.org/10.1145/2988450.2988454

Burke, R., Felfernig, A., & Göker, M. H. (2011). Recommender Systems: An Overview. AI Magazine, 32(3), 13. doi:10.1609/aimag.v32i3.2361

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. Egyptian Informatics Journal, 16(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

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. https://doi.org/10.1007/978-3-540-72079-9_12

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. https://doi.org/10.1007/978-981-15-6759-9_1

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). https://doi.org/10.1007/s10462-018-9654-y

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. http://arxiv.org/abs/1806.01261

Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, 404(March), 1–43. https://doi.org/10.1016/j.physd.2019.132306

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. https://doi.org/10.1007/978-981-13-9113-2_16

Lops, P., Gemmis, M. De, & Semeraro, G. (2011). Recommender Systems Handbook. In Recommender Systems Handbook (Issue January). https://doi.org/10.1007/978-0-387-85820-3

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. https://doi.org/10.1109/ISRITI51436.2020.9315363

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. https://www.researchgate.net/publication/323700644

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. https://doi.org/10.1016/j.neucom.2020.10.066

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). https://doi.org/10.3390/app8122426

Katrompas, A., & Metsis, V. (2022). Enhancing LSTM Models with Self-attention and Stateful Training. Lecture Notes in Networks and Systems, 294, 217–235. https://doi.org/10.1007/978-3-030-82193-7_14

Zhang, S., Tay, Y., Yao, L., & Sun, A. (2018). Next Item Recommendation with Self-Attention. August. http://arxiv.org/abs/1808.06414

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. https://doi.org/10.22146/ijccs.60733

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. https://doi.org/10.1109/TII.2018.2867174

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. https://doi.org/10.1145/3343031.3356070

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. http://arxiv.org/abs/2106.12970

Lampiran