**APLIKASI REKOMENDASI PERPUSTAKAAN DIGITAL MENGGUNAKAN DEEP COLLABORATIVE FILTERING BERBASIS MOBILE DAN WEB**

**TUGAS AKHIR**

Oleh:

**EVAN OWEN PASARIBU** (NIM. 171111067)

**IMAN FAJAR LOMBU** (NIM. 171111768)

**MARTUA SINAGA** (NIM. 171112356)



**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**MIKROSKIL**

**MEDAN**

**2021**

**MOBILE AND WEB APPLICATION FOR DIGITAL LIBRARY RECOMMENDATION USING DEEP COLLABORATIVE FILTERING**

**FINAL RESEARCH**

By:

**EVAN OWEN PASARIBU** (NIM. 171111067)

**IMAN FAJAR LOMBU** (NIM. 171111768)

**MARTUA SINAGA** (NIM. 171112356)



**STUDY PROGRAM OF INFORMATICS ENGINEERING**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

**MIKROSKIL**

**MEDAN**

**2021**

# DAFTAR ISI

[DAFTAR ISI i](#_Toc69927279)

[BAB I PENDAHULUAN 1](#_Toc69927280)

[1.1 Latar Belakang 1](#_Toc69927281)

[1.2 Rumusan Masalah 2](#_Toc69927282)

[1.3 Tujuan 2](#_Toc69927283)

[1.4 Manfaat 2](#_Toc69927284)

[1.5 Batasan Masalah 2](#_Toc69927285)

[1.6 Metodologi Pengembangan Sistem 3](#_Toc69927286)

[BAB II TINJAUAN PUSTAKA 4](#_Toc69927287)

[2.1 Perpustakaan Digital 4](#_Toc69927288)

[2.1.1 E-book 4](#_Toc69927289)

[2.1.2 Format E-book 4](#_Toc69927290)

[2.1.3 Klasifikasi Buku Pada Perpustakaan 5](#_Toc69927291)

[2.2 Sistem Rekomendasi 7](#_Toc69927292)

[2.2.1 Content-Based Recommendation 8](#_Toc69927293)

[2.2.2 Collaborative Recommendations 9](#_Toc69927294)

[2.2.3 Hybrid Approaches 10](#_Toc69927295)

[2.3 Collaborative Filtering 10](#_Toc69927296)

[2.4 Matrix Factorization 12](#_Toc69927297)

[2.5 Deep Collaborative Filtering 13](#_Toc69927298)

[2.5.1 Mempelajari Latent Factor dari Rating dan Side Information 14](#_Toc69927299)

[2.5.2 Normalisasi Data pada Deep Collaborative Filtering 15](#_Toc69927300)

[2.5.3 Mendapatkan Latent Factor dari User dan Item 16](#_Toc69927301)

[2.6 Pengujian Perangkat Lunak dan Pengujian Keakuratan Hasil Rekomendasi 17](#_Toc69927302)

# BAB I PENDAHULUAN

## Latar Belakang

Saat ini sistem rekomendasi telah diterapkan di berbagai domain seperti musik, film, buku, dan produk (Yang, 2019). Penggunaan teknik rekomendasi yang akurat dan efisien sangat penting bagi suatu sistem untuk memberikan hasil yang baik dan bermanfaat kepada setiap penggunanya (Isinkaye, et al., 2015). Sistem rekomendasi *collaborative filtering* dapat merekomendasikan *item* berdasarkan kumpulan *user* dengan preferensi yang sama (Rahmawati, et al., 2018). Tetapi, sistem rekomendasi ini mengalami masalah *sparsity* pada matriks *rating*. Banyak *user* hanya menilai sejumlah kecil *item* sehingga rekomendasi yang dihasilkan memiliki akurasi yang sangat rendah (Saeed & Mansoori, 2017). Sementara dalam konteks perpustakaan digital, banyaknya koleksi dan kebutuhan informasi yang beragam membuat *rating* dari *user* sangat diperlukan. Tetapi, *rating* pada perpustakaan digital cenderung jarang diberikan oleh *user* (Wenige & Ruhland, 2018). Oleh karena itu, diperlukan suatu teknik yang dapat memprediksi *rating* *user* terhadap *item* (Bobadilla, et al., 2020).

*Deep learning*, sebagai jenis pendekatan *machine learning* telah berhasil diterapkan di banyak domain penelitian berbeda, seperti *computer vision*, *speech recognition*, *natural language processing* dan lain sebagainya (Liu & Wu, 2017). Menggunakan *deep learning* pada sistem rekomendasi memungkinkan model secara otomatis mempelajari fitur yang digeneralisasi dengan baik bagi *user* dan *item* dari sumber daya yang berbeda. Dengan menggunakan *deep learning* dalam pemodelan berbagai tipe data, sistem rekomendasi akan dapat lebih memahami apa yang dibutuhkan *user* dan hal tersebut akan meningkatkan hasil rekomendasi (Ikasari, et al., 2018).

Model *deep learning* untuk *collaborative filtering* merupakan penggabungan *collaborative filtering* berbasis *matrix factorization* dengan algoritma *deep learning*. *deep collaborative filtering* yang merupakan model *hybrid*, menggunakan matriks *rating* dan *side information* serta menjembatani *matrix factorization* dan fitur *learning*. *Deep collaborative filtering* mengintegrasikan *matrix factorization* dan fitur *deep learning* dengan memodelkan pemetaan antara *latent factors* yang digunakan dalam *collaborative filtering* dan *latent layers* dalam *deep models* (Li, et al., 2015).

Penerapan *deep collaborative filtering* pada perpustakaan digital akan memberikan rekomendasi *item* lebih akurat daripada hanya menggunakan *collaborative filtering*. User pada perpustakaan digital jarang memberikan *rating* pada *item* (Wenige & Ruhland, 2018). Jika hanya menggunakan *collaborative filtering*, hasil rekomendasi tidak akan akurat karena terjadi *sparsity* matriks *rating*. Oleh karena itu, *deep collaborative filtering* diharapkan dapat mengatasi masalah tersebut pada aplikasi perpustakaan digital.

Berdasarkan uraian di atas, maka dikembangkan sebuah aplikasi yang dapat memberikan rekomendasi kepada user tanpa mengalami *sparsity* pada matriks *rating* dan dituangkan dalam tugas akhir dengan judul “APLIKASI REKOMENDASI PERPUSTAKAAN DIGITAL MENGGUNAKAN METODE DEEP COLLABORATIVE FILTERING BERBASIS MOBILE DAN WEB”.

## Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka yang menjadi permasalahan pada penelitian ini adalah *user* jarang memberikan *rating* pada perpustakaan digital sehingga mengalami *sparsity* matriks *rating*. Akibatnya *user* tidak mendapatkan hasil rekomendasi yang baik dan bermanfaat.

## Tujuan

Tujuan tugas akhir ini yaitu sebagai berikut:

1. Menerapkan Deep Collaborative Filtering untuk mengatasi sparsity matriks rating pada perpustakaan digital.
2. Mengembangkan aplikasi rekomendasi perpustakaan digital yang memberikan rekomendasi dalam menentukan item yang sesuai dengan kebutuhan dan keinginan user.

## Manfaat

Manfaat yang diharapkan dari tugas akhir ini adalah sebagai berikut:

1. Aplikasi rekomendasi perpustakaan digital yang bersifat easy to use, sehingga mempermudah dalam melakukan pencarian dan menampilkan hasil sesuai keinginan user.
2. Hasil laporan ini diharapkan menjadi referensi dalam pengembangan sistem rekomendasi perpustakaan digital yang lebih besar.

## Batasan Masalah

Batasan masalah pada penulisan Tugas Akhir ini adalah:

1. Rating didapatkan dengan asumsi ketika user telah selesai membuka item perpustakaan.
2. Member akan mendapatkan rekomendasi setelah mendaftar pada perpustakaan digital.
3. Guest akan mendapatkan rekomendasi item dengan rating tertinggi.
4. Mobile application sebagai produk dari Tugas Akhir ini hanya dapat dijalankan pada smartphone dengan operating system Android versi minimal Lollipop 5.0.
5. Jumlah aktor dalam sistem ini yaitu dari 3 (tiga), terdiri dari admin, member dan guest dengan peran masing-masing aktor yaitu:

Admin:

1. Mengelola data item digital.
2. Melakukan pengujian keakuratan sistem rekomendasi.
3. Melakukan verifikasi member terdaftar.
4. Mengelola laporan member, item digital, pengguna baru.

Member:

1. Mendapat rekomendasi item digital.
2. Mencari, membaca, mengembalikan dan memperpanjang item digital.
3. Memberi rating dan review item digital yang telah dibaca.

Guest:

1. Mendapat rekomendasi item digital.
2. Mencari dan melihat rincian item digital.
3. Melakukan pendaftaran pada sistem.

## Metodologi Pengembangan Sistem

Metodologi yang digunakan pada pengembangan sistem ini yaitu dengan model waterfall dengan tahapan sebagai berikut:

# BAB II TINJAUAN PUSTAKA

## Perpustakaan Digital

Kemajuan teknologi komunikasi informasi, internet, dan digitalisasi, buku, surat kabar, majalah, jurnal, video ceramah, dan sebagian besar buku referensi mengubah konsep perpustakaan konvensional dengan model perpustakaan baru yang disebut *resource center*. Sumber daya digital ini disimpan menggunakan teknologi penyimpanan elektronik atau berlangganan dari penyedia layanan tersebut untuk mengurangi penyimpanan buku, jurnal, majalah berbasis kertas dan kemudahan dalam pencarian database, akses fasilitas, dan menekan biaya pemeliharaan perpustakaan (Aithal, 2016).

Perpustakaan digital adalah bidang penelitian yang terus berkembang. Perpustakaan digital didefinisikan sebagai kumpulan informasi elektronik yang berisi *repository* objek digital yang besar dan beragam, dan dapat diakses oleh sejumlah besar pengguna yang tersebar secara geografis. Objek digital yang dimaksud seperti teks, gambar, peta, suara, video, katalog, dan jurnal ilmiah (Khiste, et al., 2018). Setiap pengguna dapat mengakses sumber daya hingga informasi yang tersedia di perpustakaan dengan menggunakan komputer dan koneksi ke jaringan perpustakaan tanpa harus hadir secara fisik di perpustakaan (Nahak & Padhi, 2019).

### E-book

*E-book* adalah kependekan “*electronic book*” yang merupakan representasi elektronik dari sebuah buku (Sargeant, 2015). Saat ini, penggunaan *e-book* menjadi lebih populer dengan fitur-fitur yang tidak ada dalam buku cetak berbasis kertas seperti kamus internal, video dan audio yang disematkan, fungsionalitas pencarian, dan dapat dibaca menggunakan perangkat lunak atau perangkat *e-reader*. Saat ini komputer desktop, laptop, perangkat genggam, dan ponsel dapat berfungsi sebagai perangkat e-reader karena ketersediaan perangkat lunak membaca yang sesuai (Faniband, et al., 2020). *E-book* dalam perpustakaan membawa keuntungan karena hemat biaya yang memungkinkan satu buku dapat di akses oleh banyak orang dalam waktu yang sama (Manley & Holley, 2012).

### Format E-book

1. PDF

PDF adalah salah satu format *e-book* paling populer. Karena format PDF dikembangkan untuk menyediakan sarana yang tidak bergantung platform untuk menampilkan dan bertukar dokumen dengan tata letak tetap. File PDF didukung oleh hampir semua perangkat *e-reader*, seperti *tablet*, PC, *laptop*, dan *smartphone*. Hal tersebut membuat PDF dapat diterima secara luas sebagai format pengarsipan digital. Namun, beberapa penelitian juga menunjukkan bahwa PDF bukanlah format yang ideal untuk membaca di layar perangkat. Karena format PDF yang dirancang untuk mereproduksi halaman dengan tata letak tetap, sehingga aliran ulang teks agar sesuai dengan perangkat seluler dan layar pembaca *e-book* bisa menjadi masalah. Mallett mencatat bahwa perangkat *Sony Reader* dan *iPod Touch* melakukan pekerjaan yang buruk dalam mendukung PDF, dokumen ditampilkan dengan ukuran yang sangat kecil dan kata-katanya terkadang campur aduk.

1. EPUB

EPUB merupakan standar *e-book* gratis dan terbuka yang dibuat oleh *International Digital Publishing Forum* (IDPF). EPUB dikembangkan dengan menggunakan format teks mengalir secara *real-time* sehingga orang dapat membacanya menggunakan *e-reader* mana pun. Hal tersebut membuat EPUB menjadi format file yang disukai dalam hal keterbacaan informasi secara tekstual dan lebih cocok untuk persyaratan ilmiah daripada format PDF. Meskipun EPUB adalah standar industri, EPUB belum sepenuhnya didukung oleh semua perangkat *e-reader.*

### Klasifikasi Buku Pada Perpustakaan

Klasifikasi merupakan salah satu alat tertua yang dikembangkan oleh ahli pustakawan dan dianggap sebagai dasar dari kepustakawanan karena fungsinya yang serba guna. Tujuan tradisional klasifikasi untuk menempatkan objek fisik di rak telah kehilangan fungsinya dalam kasus sumber daya elektronik jarak jauh. Namun, klasifikasi memiliki fungsi lain dalam kategorisasi, yang lebih berguna dalam kasus sumber daya elektronik, karena berperan dalam pengorganisasian subjek (Kumbhar, 2012). Berikut beberapa sistem klasifikasi yang digunakan sebagai pedoman dalam perpustakaan, yaitu:

1. *Dewey Decimal Classification*

*Dewey Decimal Classification* atau disingkat DDC merupakan sistem yang mengorganisasi pengetahuan secara umum, yang terus menerus direvisi untuk mengikuti perkembangan pengetahuan. DDC diterbitkan pertama kali pada tahun 1873 oleh Melvin Dewey. Edisi termutakhir dari DDC adalah DDC 23 yang diterbitkan pada tahun 2011 oleh Joan S. Mitchell dengan hak cipta pada *Online Computer Library Centre, Inc*. (OCLC) (Mallawa, 2012).

Table 2.1 Kelas Umum Dewey Decimal Classification 23

Sumber (Suresha & Narayanaswamy, 2016)

|  |  |  |
| --- | --- | --- |
| No | Kode | Subject |
| 1 | 000 | *Computers, Information and General Reference* |
| 2 | 100 | *Philosophy and Psychology* |
| 3 | 200 | *Religion* |
| 4 | 300 | *Social Sciences* |
| 5 | 400 | *Language* |
| 6 | 500 | *Science and Mathematics* |
| 7 | 600 | *Technology* |
| 8 | 700 | *Arts and Recreation* |
| 9 | 800 | *Literature* |
| 10 | 900 | *History and Geography* |

1. *Universal Decimal Classification*

*Universal Decimal Classification* atau disingkat UDC merupakan klasifikasi internasional pertama yang dirancang untuk informasi bibliografi dan pekerjaan dokumentasi yang disebut perluasan dari *Dewey Decimal Classification*. *Universal Decimal Classification* mencakup seluruh domain pengetahuan, dengan 10 kelas utama dan kelas utama ke-4 masih kosong sebagai berikut:

Table 2.2 Kelas Utama Universal Decimal Classification

Sumber (Satija, 2008)

|  |  |  |
| --- | --- | --- |
| No | Kode | Subject |
| 1 | 0 | *Generalities* |
| 2 | 1 | *Philosophy; Psychology* |
| 3 | 2 | *Religion, Theology* |
| 4 | 3 | *Social Sciences* |
| 5 | 4 | *[Vacant]* |
| 6 | 5 | *Natural Sciences; Mathematics* |
| 7 | 6 | *Technology* |
| 8 | 7 | *The Arts* |
| 9 | 8 | *Language; Linguistics; Literature* |
| 10 | 9 | *Geography; Biography; History* |

Dari ke-10 kelas utama tersebut dapat diperluas lagi dengan notasi dari table bantu yang terdiri dari dua jenis, yaitu Alat bantu umum yang dapat diterapkan secara universal dan Alat bantu khusus yang dapat diterapkan secara terbatas atau secara local (Satija, 2008).

## Sistem Rekomendasi

Sistem rekomendasi merupakan fitur yang berfungsi untuk membantu *user* untuk menemukan saran tentang *item* apa yang sebaiknya digunakan atau dipilih. Secara umum Sistem Rekomendasi ditujukan untuk individu yang kekurangan pengalaman atau kompetensi yang cukup untuk mengevaluasi banyaknya jumlah alternatif *item* yang ada pada suatu kasus tertentu (Elahi, 2016). Terdapat beragam teori yang dapat digunakan untuk membuat sistem rekomendasi seperti *bayesian algorithm* (Bobadilla, 2017), *decision tree algorithm* (Guabassi, 2016) dan *matrix factorization-based* (Aleksandrova, 2017) dan lain-lain. Contoh *matrix factorization-based* dapat dilihat pada gambar 2.1.

Gambar 2. 1 Contoh Matrix factorization-based

(Sumber : <https://developers.google.com/machine-learning/recommendation/collaborative/matrix>)



Dalam melakukan pengumpulan data user untuk sistem rekomendasi, terdapat dua cara (Zulkarnain, 2013) yaitu dengan cara Eksplisit dan Implisit. Eksplisit merupakan pengumpulan data yang diperoleh berdasarkan feedback dari pengguna secara langsung, data dapat diperoleh dengan meminta pendapat langsung dari pengguna, bisa berupa rating, likes/dislikes, maupun kata pencarian (keyword). Metode Implisit merupakan pengumpulan data berdasarkan pengamatan pola kecenderungan pengguna, seperti kecenderungan user terhadap suatu item dengan kriteria tertentu yang dimiliki item tersebut. Metode pengumpulan data pada teknik implisit tidak melibatkan kontribusi pengguna melainkan dengan cara pendekatan dari perilaku pengguna dan menyimpulkan data-data yang sudah dikumpulkan oleh sistem. Berdasarkan pendekatan yang digunakan untuk menghasilkan rekomendasi, sistem rekomendasi dibagi menjadi enam bagian (Francesco Ricci, 2011) yaitu:

### Content-Based Recommendation

Sistem rekomendasi berbasis konten (Content-based Recommendation) menggunakan ketersediaan konten (sering juga disebut dengan fitur, atribut atau karakteristik) sebuah item sebagai basis dalam pemberian rekomendasi (Francesco Ricci, 2011). Sebagai contoh, sebuah film mempunyai konten seperti genre, author, tahun rilis, dan lain-lain, atau sebuah file dokumen memiliki konten berupa tulisan yang ada di dalamnya.

Sistem rekomendasi berbasis konten mencoba untuk melakukan mencocokkan (matching) antara profil user(user profile) dengan konten item(item content).

Misalnya, seorang user telah memberi rating kepada 5 buah item 𝑖1, 𝑖2, 𝑖3, 𝑖4, 𝑖5 yang memiliki empat buah fitur 𝑗1, 𝑗2, 𝑗3, 𝑗4, 𝑗5. Angka 1 menandakan bahwa item tersebut memiliki atribut yang bersangkutan, sedangkan angka 0 menunjukkan bahwa item tersebut tidak memiliki atribut tersebut:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *Fitur* | *Fitur* | *Fitur* | *Fitur* | *Fitur* |
| *Item* | 1 | 1 | 1 | 1 | 4 |
| *Item* | 1 | 0 | 1 | 1 | 3 |
| *Item* | 1 | 0 | 0 | 0 | 2 |
| *Item* | 1 | 0 | 1 | 1 | 3 |
| *Item* | 1 | 1 | 0 | 1 | ? |

Untuk menghitung bobot dari masing-masing atribut dapat dengan menggunakan rumus berikut (Uluyagmur, et al., 2012).



Dimana:

𝑤 (𝑢, 𝑗𝑘) merupakan bobot yang dimiliki oleh user u terhadap fitur 𝑗𝑘.

𝐼𝑢 merupakan satu set item yang telah di rating oleh user u.

𝑥 (𝑖, 𝑗) merupakan nilai kehadiran (angka 1atau 0) sebuah fitur di dalam sebuah item.

𝑟 (𝑢, 𝑖) merupakan rating yang diberikan user u terhadap item i.

Sedangkan untuk menghitung prediksi rating yang akan diberikan seorang user terhadap sebuah item dapat menggunakan formula berikut ini:



Dimana :

𝑅(𝑢,𝑖)merupakan prediksi rating user u terhadap item i.𝐷𝑖merupakan fitur yang muncul di dalam item i. Sistem rekomendasi berbasis konten memiliki beberapa kelebihan, yaitu:

1. Sistem rekomendasi berbasis konten dapat menjelaskan bagaimana hasil rekomendasi didapatkan.
2. Sistem rekomendasi berbasis konten dapat merekomendasikan item-item yang bahkan belum pernah di-rate oleh siapapun.

Namun, sistem rekomendasi berbasis konten juga memiliki beberapa kelemahan, yaitu:

1. Sistem rekomendasi berbasis konten tidak memiliki kemampuan untuk dapat memberikan hasil rekomendasi yang tidak terduga (Serendipity Problem).
2. Sistem rekomendasi berbasis konten memerlukan sebuah profil user yang berisikan ketertarikan dan minat user. Bagi user baru yang belum pernah melakukan aktivitas apapun dan tidak memiliki profil user yang cukup, sistem rekomendasi tidak dapat memberikan rekomendasi yang handal kepadanya (Cold Start Problem).

### Collaborative Recommendations

Collaborative Recommendation adalah sistem rekomendasi yang memanfaatkan opini user lain yang ada untuk memprediksi item yang mungkin akan disukai atau diminati oleh seorang user (Francesco Ricci, 2011). Kualitas rekomendasi yang diberikan dengan menggunakan metode ini sangat bergantung dari opini user lain (neighbor) terhadap suatu item. Belakangan diketahui bahwa melakukan reduksi neighbor(yaitu dengan memotong neighbor sehingga hanya beberapa user yang memiliki kesamaan / similarity tertinggi sajalah yang akan digunakan dalam perhitungan) mampu meningkatkan kualitas rekomendasi yang diberikan.

Sistem rekomendasi collaborative filtering beroperasi di dalam sebuah ruang dua dimensi User x Item. Rating yang dapat diberikan oleh seorang user terhadap sebuah item dapat direpresentasikan sebagai R (bilangan bulat tidak negatif atau bilangan real dengan jarak tertentu), dan sistem rekomendasi mencoba untuk memprediksi rating yang akan diberikan seorang user untuk sebuah item yang belum pernah ia beri rating sebelumnya.

### Hybrid Approaches

Masing-masing teknik sistem rekomendasi memiliki kelebihan dan kelemahannya tersendiri. Karenanya, Hybrid Recommendation System yang menggabungkan kekuatan dari model dan algoritma yang berbeda untuk mengatasi masalah-masalah yang telah disebutkan di atas menjadi target penelitian yang sedang marak dikembangkan (Lü, et al., 2012).

Terdapat beberapa cara penggabungan yang dapat dilakukan dalam metode hybrid yaitu:

1. Penggabungan Linear(Linear Combination)

Penggabungan ini menggabungkan hasil prediksi (rating) dari metode content-based dan collaborative. Penggabungan ini dilakukan dengan cara pemberian ranking atau rating.

1. Penggabungan secara Sekuensial (Sequential Combination)

Penggabungan ini adalah melakukan perhitungan pada salah satu metode (misalkan content-based) kemudian hasilnya digabungkan dengan metode lainnya (misalkan collaborative).

1. Penggabungan secara Item-based Clustering Hybrid Method (ICHM)

Penggabungan ini mengintegrasikan informasi item dan rating pengguna untuk menghitung kemiripan item-item. Item-based clustering Hybrid Method (ICHM) merupakan sebuah metode yang menerapkan penggabungan hybrid recommender system dengan tujuan untuk meningkatkan akurasi prediksi pendekatan collaborative filtering dan menangani masalah item baru yang belum di rating (cold-star problem).

## Collaborative Filtering

Collaborative Filtering merupakan pemfilteran kolaboratif yang relatif sederhana, efektif, dan telah digunakan secara luas oleh banyak situs web komersial. Collaborative Filtering memanfaatkan data rating item yang ada oleh user untuk membuat prediksi tentang preferensi user. Pendekatan Collaborative Filtering dapat dibagi menjadi dua kelompok: Pendekatan berbasis memori dan model. Pendekatan berbasis memori (berbasis heuristik) membuat prediksi berdasarkan kesamaan antara user dan item. Pendekatan berbasis model berusaha membuat model prediksi melalui machine learning. Secara khusus, model berbasis faktorisasi matriks telah memperoleh popularitas karena memiliki akurasi dan skalabilitas yang relatif tinggi. Meski demikian, Collaborative Filtering masih memiliki tantangan terhadap sparsity matriks rating (Bo Yang, 2016).

Collaborative Filtering bertujuan untuk merekomendasikan item baru atau memperkirakan kegunaan item tertentu terhadap user tertentu berdasar dari kesukaan user di masa lalu dan pandangan dari user lain yang berpikiran sama. Terdapat dua tugas yang dapat dilakukan oleh Collaborative Filtering yakni:

1. Rating Prediction yaitu memprediksi peringkat yang akan dimiliki oleh item yang tak terlihat kepada user target
2. Recommendation Task yaitu menyediakan daftar rekomendasi top-N dari item relevan yang tak terlihat kepada user target

Collaborative Filtering beroperasi di dalam sebuah matrix rating. Rating yang diberikan oleh user terhadap item direpresentasikan sebagai R dan nilai rating tersebut merupakan bilangan bulat tidak negatif atau bilangan real dengan jarak tertentu. Collaborative Filtering mencoba memprediksi rating yang akan diberikan oleh user terhadap suatu item yang belum pernah di beri rating sebelumnya.

Misalnya, terdapat lima user 𝑢1,2,𝑢3,..𝑢5 dan lima item 𝑖1,2,𝑖3,..𝑖5. Sebuah sistem rekomendasi ingin memprediksi berapa rating yang akan diberikan oleh 𝑢1 kepada 𝑖5.

Table 2.3 Matrix User x Item pada Sistem Rekomendasi Collaborative Filtering

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *Item* | *Item* | *Item* | *Item* | *Item* |
| *User* | 5 | 7 | 5 | 7 | 7 |
| *User* | 5 | 7 | 5 | 7 | 9 |
| *User* | 5 | 7 | 5 | 7 | 9 |
| *User* | 6 | 6 | 6 | 6 | 5 |
| *User* | 6 | 6 | 6 | 6 | 5 |

Terdapat sebuah cara yang sering digunakan dalam menghitung atau memprediksi rating user u terhadap item

i dengan mengkalkulasikan rata-rata rating user lain (neighbor) Invalid source specified., yaitu rumus weighted sum:

Dimana:

(u,𝑖) adalah prediksi rating user u terhadap item i.

(𝑢′,) adalah rating user u’ (user selain u) terhadap item i.

(𝑢,′) adalah kemiripan antara user 𝑢 dengan 𝑢′.

N(u) adalah satu set user lain yang mirip dengan user u.

Cosine-based Similarity adalah cara terbaik untuk menghitung kemiripan dua user Invalid source specified.. Cosine-based similarity dapat dikalkulasikan dengan formula sebagai berikut:

Dimana:

𝑠𝑖𝑚 (𝑢,’) adalah kemiripan antara user 𝑢 dengan 𝑢’.

(𝑢,’) adalah satu set item yang telah di-rating oleh kedua user 𝑢 dengan 𝑢’.

Sistem rekomendasi Collaborative Filtering memiliki beberapa kekurangan, diantaranya adalah:

1. Jika terdapat item baru di dalam sistem tersebut, maka sistem tidak dapat merekomendasikan item tersebut sampai user lain berminat pada item tersebut. (First Rater Problem)
2. Hasil rekomendasi tidak selamanya handal. Terdapat kemungkinan adanya sebuah user yang hanya memiliki sedikit neighbor dengan tingkat kesamaan (similarity) yang kecil. (Grey Sheep Problem)
3. Hasil rekomendasi yang tidak handal yang dikarenakan matrix rating yang jarang. (Sparsity Problem)

## Matrix Factorization

Pada penelitian ini, data yang didapatkan akan direpresentasikan berbentuk matriks. Matriks tersebut ditulis dalam bentuk *Rmxn*dimana m merupakan banyaknya baris dan n merupakan banyaknya kolom. Dimana pada matriks tersebut setiap baris merupakan user dan tiap kolom merupakan item dan hubungan antar user dan item (*rij*) adalah rating yang diberikan user terhadap item.

Table 2.4 Contoh matriks rating user terhadap suatu item

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| # | Item-1 | Item-2 | Item-3 | Item-4 | … | Item-n |
| User-1 | r11 | r12 | r13 | r14 | … | r1n |
| User-2 | r21 | r22 | r23 | r24 | … | r2n |
| User-3 | R31 | r32 | r33 | r34 | … | r3n |
| … | … | … | … | … | … | … |
| User-m | rm1 | rm2 | rm3 | rm4 | … | rmn |

Pada tabel 2.4 akan terdapat nilai rating yang kosong. Hal ini dikarenakan user tidak memberikan rating kepada item yang ada. Banyaknya user yang hanya menilai sejumlah item saja menyebabkan terjadinya sparsity matriks rating (Zhang & Chow, 2016). Matrix Factorization adalah pendekatan Collaborative Filtering yang paling efektif. Ini memungkinkan kita untuk menemukan latent factor interaksi user-item dengan memfaktorkan matriks interaksi ke dalam latent space fitur user-item. Salah satu metode Matrix Factorization klasik adalah Probabilistic Matrix Factorization (PMF). Banyak algoritma yang telah dikembangkan untuk meningkatkan kinerja PMF, dengan memasukkan side information seperti hubungan sosial. Tetapi metode Matrix Factorization mengalami masalah cold-start, yaitu rekomendasi apa yang harus dibuat ketika user / item baru tiba di sistem. Masalah lain yang sering muncul di banyak aplikasi dunia nyata adalah ketersebaran data atau cakupan yang berkurang. Memasukkan side information telah menunjukkan kinerja yang menjanjikan dalam Collaborative Filtering. Tetapi akan bermasalah jika side information tidak lengkap. Oleh karena itu, fitur pembelajaran untuk Matrix Factorization sangat penting untuk diterapkan (Li, et al., 2015).

## Deep Collaborative Filtering

Algoritma Deep Collaborative Filtering adalah algoritma yang dibuat untuk mengatasi masalah sparsity matriks rating pada algoritma Collaborative Filtering. Penyelesaian permasalahan tersebut dilakukan dengan mempelajari side information yang diperoleh dari profil user / item seperti demografi user, genre buku, dll. Deep Collaborative Filtering merupakan algoritma yang dikenal sebagai penggabungan matrix factorization pada Based Collaborative Filtering dengan Deep Learning. Deep Collaborative Filtering mengintegrasikan faktorisasi matriks dan pembelajaran deep feature yang memodelkan pemetaan antara faktor laten yang digunakan dalam Collaborative Filtering dan lapisan laten dalam deep models (Li, et al., 2015).

Gambar 2.2 Ilustrasi penerapan kerangka Deep collaborative Filtering

Sumber (Li, et al., 2015)



Table 2.2 – Ringkasan Notasi

|  |  |
| --- | --- |
| Notasi | Deskripsi |
| m | Jumlah *user* |
| n | Jumlah *item* |
| d | Dimensi *latent factor* |
| p | Dimensi *user feature* |
| q | Dimensi *item feature* |
| *R* € | Matriks *Rating* |
| *U* € | *Latent factor user* |
| *V* € | *Latent factor item* |
| *X* € | *Side information of user* |
| *Y* € | *Side information of item* |
| *W1*  € | *Mapping Function* untuk X di *auto-encoder* |
| *P1* € | *Projection matrix* untuk *U* |

### Mempelajari Latent Factor dari Rating dan Side Information

Deep Collaborative Filtering adalah model hybrid, yang menggunakan matriks rating dan side information serta menjembatani faktorisasi matriks dan pembelajaran fitur. Diberikan matriks rating user-item, side information user dan side information item, Deep Collaborative Filtering mendekomposisi matriks rating dan mempelajari latent factor dari rating dan side information dengan rumus berikut:

Ada dua komponen kunci dalam Deep Collaborative Filtering: (i) fungsi l (R, U, V) untuk menguraikan matriks rating menjadi dua matriks laten; (ii) fungsi L (X, U) dan L (Y, V) yang menghubungkan fitur kontekstual user/item dengan faktor laten. Komponen pertama yang diturunkan melalui matrix factorization, mengekstrak pengetahuan laten dari matriks rating. Komponen kedua yang dirancang dengan menggunakan model pembelajaran mendalam membangun koneksi side information dengan faktor laten (Li, et al., 2015).

### Normalisasi Data pada Deep Collaborative Filtering

Normalisasi data dilakukan pada matriks rating yang mengindikasikan penilaian oleh user terhadap item. Normalisasi dilakukan dengan menggunakan zero-mean normalization. Nilai rating dari dataset dalam skala 1 – 10 akan dinormalisasikan menjadi zero-mean. U1-U10 adalah user dan J1-J10 adalah item yang dinilai ataupun yang tidak dinilai oleh user.

Table 2.5 Data Sebelum Normalisasi

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 | J9 | J10 |
| U1 | 8 | 4 | 3 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| U2 | 0 | 1 | 0 | 1 | 2 | 1 | 0 | 2 | 0 | 5 |
| U3 | 5 | 1 | 1 | 0 | 1 | 3 | 0 | 1 | 4 | 0 |
| U4 | 0 | 0 | 3 | 4 | 0 | 7 | 0 | 6 | 0 | 6 |
| U5 | 0 | 0 | 1 | 0 | 0 | 5 | 2 | 1 | 3 | 0 |
| U6 | 0 | 1 | 0 | 5 | 3 | 4 | 0 | 4 | 8 | 2 |
| U7 | 1 | 2 | 0 | 0 | 0 | 4 | 3 | 3 | 4 | 5 |
| U8 | 0 | 4 | 3 | 2 | 3 | 0 | 2 | 0 | 0 | 0 |
| U9 | 0 | 3 | 4 | 2 | 0 | 1 | 2 | 2 | 1 | 6 |
| U10 | 1 | 4 | 0 | 3 | 1 | 3 | 0 | 3 | 5 | 0 |

Table 2.6 Data Setelah Normalisasi

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| User | J1 | J2 | J3 | J4 | J5 | J6 | J7 | J8 | J9 | J10 |
| U1 | 2,989 | 1,042 | 0,555 | -0,905 | -0,905 | -0,419 | -0,905 | -0,905 | -0,905 | -0,905 |
| U2 | -0,905 | -0,419 | -0,905 | -0,419 | 0,068 | -0,419 | -0,905 | 0,068 | -0,905 | 1,528 |
| U3 | 1,528 | -0,419 | -0,419 | -0,905 | -0,419 | 0,555 | -0,905 | -0,419 | 1,042 | -0,905 |
| U4 | -0,905 | -0,905 | 0,555 | 1,042 | -0,905 | 2,502 | -0,905 | 2,015 | -0,905 | 2,015 |
| U5 | -0,905 | -0,905 | -0,419 | -0,905 | -0,905 | 1,528 | 0,068 | -0,419 | 0,555 | -0,905 |
| U6 | -0,905 | -0,419 | -0,905 | 1,528 | 0,555 | 1,042 | -0,905 | 1,042 | 2,989 | 0,068 |
| U7 | -0,419 | 0,068 | -0,905 | -0,905 | -0,905 | 1,042 | 0,555 | 0,555 | 1,042 | 1,528 |
| U8 | -0,905 | 1,042 | 0,555 | 0,068 | 0,555 | -0,905 | 0,068 | -0,905 | -0,905 | -0,905 |
| U9 | -0,905 | 0,555 | 1,042 | 0,068 | -0,905 | -0,419 | 0,068 | 0,068 | -0,419 | 2,015 |
| U10 | -0,419 | 1,042 | -0,905 | 0,555 | -0,419 | 0,555 | -0,905 | 0,555 | 1,528 | -0,905 |

### Mendapatkan Latent Factor dari User dan Item

Latent factor telah menjadi teknik yang sukses digunakan untuk membangun sistem rekomendasi. Meskipun kuncinya adalah minat pengguna secara efektif, sebagian besar penelitian difokuskan untuk mempelajari latent factor dalam situasi cold-start dan data sparsity (Rao, et al., 2017). Berikut Algoritma untuk mendapatkan latent factor user dan Item.

**Algorithm** 1. mDA-CF Algorithm

**Input:** Rating matrix R, user feature X, item feature Y,

Parameters

**Output :** Latent factor U,V

1: Initialize U,V,and

2: **while** validation error decreases, **do**

3: Update using (9);

4: Update using (10);

5: Update using (12);

6: Update using (13);

7: **for** each observed , **do**

8: Update using (14);

9: Update using (14);

10: **end for**

11: **end while**

Dimana rumus di dalam Algoritma:

1. Update W1
2. Update W2
3. Update P1
4. Update P2
5. Update ui and vj

Where is the learning rate, and the detailed derivates are defined as

## Pengujian Perangkat Lunak dan Pengujian Keakuratan Hasil Rekomendasi

Pengujian atau testing sendiri merupakan elemen kritis dari jaminan kualitas perangkat lunak dan merupakan bagian yang tidak terpisah dari siklus hidup pengembangan software seperti halnya analisis, desain, dan pengkodean (Shi, 2010). Pengujian software haruslah dilakukan dalam proses rekayasa perangkat lunak atau software engineering. Ada beberapa jenis pengujian perangkat lunak, antara lain (Khan, 2011):

1. Pengujian Perangkat Lunak dengan Black Box Testing

Pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program. Black Box Testing berfokus pada spesifikasi fungsional dari perangkat lunak. Tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengujian pada spesifikasi fungsional program. Black Box Testing bukanlah solusi alternatif dari White Box Testing tapi lebih merupakan pelengkap untuk menguji hal-hal yang tidak dicakup oleh White Box Testing. Black Box Testing cenderung untuk menemukan hal-hal yaitu sebagai berikut.

1. Fungsi yang tidak benar atau tidak ada
2. Kesalahan antarmuka (interface errors)
3. Kesalahan pada struktur data dan akses basis data
4. Kesalahan performansi (performance errors)
5. Kesalahan inisialisasi dan terminasi
6. Pengujian Keakuratan Hasil Rekomendasi

Pengujian digunakan untuk mengetahui seberapa akurat hasil rekomendasi yang diberikan sistem kepada user. Beberapa cara yang dapat dilakukan untuk mengevaluasi sistem rekomendasi yaitu dengan menghitung tingkat error pada hasil rekomendasi dan juga menghitung nilai presisi dari hasil User Acceptance Test (UAT) yang diberikan user.

Dalam sistem rekomendasi, hasil rekomendasi yang diberikan perlu dilakukan pengukuran untuk mengukur tingkat akurasi hasil rekomendasi dapat dilakukan dengan melihat nilai error pada hasil rekomendasi yang diberikan. Terdapat banyak metode perhitungan error seperti, Mean Square Error (MSE), Mean Absolute Percentage Error (MAPE) dan Mean Absolute Error (MAE).

1. Mean Square Error (MSE)

MSE adalah rata-rata absolut dari kesalahan peramalan yang dikuadratkan. Persamaan dari MSE yaitu sebagai berikut :

Dimana,

𝑓𝑡 = Permintaan aktual periode *t*

𝑓̂ = Ramalan permintaan periode *t*

m = Jumlah periode peramalan

MSE biasa dipakai untuk menghitung hasil dari peramalan namun MSE juga dapat diterapkan pada perhitungan hasil rekomendasi yaitu dengan mengganti *ft* (permintaan aktual periode *t*) menjadi *rating user* terhadap *item t* lalu (ramalan permintaan periode *t*) menjadi prediksi *rating user* terhadap *item* t dan m menjadi jumlah *item* yang dihitung.

1. Mean Absolute Percentage (MAPE)

MAPE adalah rata-rata persentase absolut dari kesalahan peramalan dengan menghitung *error absolute* tiap periode. *Error* ini kemudian dibagi dengan *n*. Persamaan dari MAPE ini yaitu sebagai berikut

1. Mean Absolute Error (MAE)

Dari ketiga jenis perhitungan *error*, MAE merupakan persamaan perhitungan nilai *error* yang paling sering digunakan untuk menghitung nilai *error* dari hasil rekomendasi. *Mean Absolute Error* (MAE) merupakan persamaan yang termasuk jenis *statistical accuracy metrics* dimana MAE akan menghitung nilai rata-rata selisih antara nilai prediksi dengan nilai yang sebenarnya. Persamaan MAE yaitu sebagai berikut:

Dimana*,*

*MAE =* Nilai rata-rata kesalahan hitungan

N = jumlah *item* yang dihitung

*pi* = Nilai prediksi *item* ke i

*qi* = Nilai *rating* sebenarnya *item* ke i

Semakin rendah nilai MAE yang didapat, maka sistem semakin akurat dalam memprediksi.