**APLIKASI REKOMENDASI PERPUSTAKAANDIGITAL 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](#_Toc69581959)

[DAFTAR GAMBAR iii](#_Toc69581960)

[DAFTAR TABEL iv](#_Toc69581961)

[BAB I PENDAHULUAN 1](#_Toc69581962)

[1.1 Latar Belakang 1](#_Toc69581963)

[1.2 Rumusan Masalah 2](#_Toc69581964)

[1.3 Tujuan 2](#_Toc69581965)

[1.4 Manfaat 2](#_Toc69581966)

[1.5 Batasan Masalah 2](#_Toc69581967)

[1.6 Metodologi Pengembangan Sistem 4](#_Toc69581968)

[BAB II TINJAUAN PUSTAKA 5](#_Toc69581969)

[2.1 Perpustakaan Digital 5](#_Toc69581970)

[2.1.1 *E-book* 5](#_Toc69581971)

[2.1.2 Format E-book 5](#_Toc69581972)

[2.1.3 Klasifikasi Buku pada Perpustakaan 6](#_Toc69581973)

[2.2 Sistem Rekomendasi 8](#_Toc69581974)

[2.2.1 Content-based recommendation 9](#_Toc69581975)

[2.2.2 Collaborative Recommendations 10](#_Toc69581976)

[2.2.3 Hybrid approaches 11](#_Toc69581977)

[2.3 Collaborative Filtering 12](#_Toc69581978)

[2.4 Matrix Factorization 14](#_Toc69581979)

[2.5 Deep Collaborative Filtering 15](#_Toc69581980)

[2.5.1 Mempelajari latent factor dari rating dan side information 16](#_Toc69581981)

[2.5.2 Normalisasi Data Pada Deep Collaborative Filtering 16](#_Toc69581982)

[2.5.3 Mendapatkan Latent Factor dari User dan Item 17](#_Toc69581983)

[2.6 Pengujian Perangkat Lunak dan Pengujian Keakuratan Hasil Rekomendasi 19](#_Toc69581984)

[2.6.1 Pengujian Perangkat Lunak dengan Black Box Testing 19](#_Toc69581985)

[2.6.2 Pengujian Keakuratan Hasil Rekomendasi 19](#_Toc69581986)

[2.6.1. Mean Absolute Error (MAE) 20](#_Toc69581987)

[2.6.2 Root Mean Square Error (RMSE) 20](#_Toc69581988)

[DAFTAR PUSTAKA 22](#_Toc69581989)

# DAFTAR GAMBAR

[Gambar 2. 1 Contoh *Matrix factorization-based* 2](#_Toc69357132)

# DAFTAR TABEL

[Table 2.1 Kelas Umum Dewey Decimal Classification 23 2](#_Toc69357013)

[Table 2.2 Kelas Utama Universal Decimal Classification 2](#_Toc69357014)

[Table 2. 3 Matrix *User* x *Item* pada Sistem Rekomendasi *Collaborative Filtering* 2](#_Toc69357015)

[Table 2.4 Contoh matriks *rating user* terhadap suatu *item* 2](#_Toc69357016)

[Table 2.5 Data Sebelum Normalisasi 2](#_Toc69357017)

[Table 2.6 Data Setelah Normalisasi 2](#_Toc69357018)

# BAB I PENDAHULUAN

## Latar Belakang

Saat ini sistem rekomendasi telah diterapkan di berbagai domain seperti musik, film, buku, dan produk (Zhang & 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, penerapan pemodelan berbagai tipe data, sistem rekomendasi akandapat 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* merupakan model *hybrid* yang menggunakan matriks *rating* dan *side information* serta menjembatani *matrix factorization* dengan 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* dan mengurangi *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.
4. Dataset yang diperoleh dari kaggle.com yang berisi metadata produk buku *pada Book-Crossing: User review ratings* pada agustus 2020 yang dikumpulkan oleh Ruchi Bhatia [(https://www.kaggle.com/ruchi798/bookcrossing-dataset)](https://www.kaggle.com/ruchi798/bookcrossing-dataset). Dataset terdiri dari 3 *file* yaitu *file* *BX\_Books* (.csv) dengan attribut *ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL,* *file BX-Book-Ratings* (.csv) dengan attribut *User-ID, ISBN, Book-Rating,* dan *file BX-Users* (.csv) dengan attribut *User-ID, Location, Age* dengan jumlah 1,149,780 *metadata.*
5. Format dokumen yang digunakan pada aplikasi perpustakaan digital berupa Portable Document Format (PDF) dan *file Electronic Publication* (EPUB) dengan ukuran maksimal 1 (satu) *file* maksimal 50 MB (Megabyte).
6. Kriteria aturan dalam aplikasi perpustakaan digital tentang penggunaan materi data digital pada aplikasi yang akan dikembangkan pada Tugas Akhir ini yaitu batas akhir penggunaan buku yaitu 1 minggu (7 hari) menggunakan model bisnis meminjam (rental) dengan ketentuan maksimal jumlah peminjaman aktif setiap user adalah 5 (lima) buku.

## Metodologi Pengembangan Sistem

Metodologi yang digunakan dalam penyusunan tugas akhir ini adalah sebagai berikut, dimana dalam membangun aplikasi menggunakan model *waterfall*.

1. Analisis Proses

Pada tahapan ini dilakukan perhitungan untuk algoritma yang digunakan*, a*nalisis yang dilakukan adalah perhitungan dengan contoh masalah yang sederhana dan menggunakan metode yang dipilih kedalam contoh permasalahan.

1. Analisis Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan fungsional, menggunakan *use case diagram*, kebutuhan *non-*fungsional memanfaatkan *PIECES* (*Performance*, *Information, Economy, Control, Efficiency, Service*).

1. Perancangan

Pada tahap ini adalah tahap lanjutan dari tahap analisis dengan melakukan perancangan tampilan *(interface)* menggunakan Figma, pemodelan basis data menggunakan Microsoft Visio, perancangan basis data menggunakan ERD (*Entity Relationship Diagrams*).

1. Perancangan

Pada tahap ini dilakukan penulisan kode program, aplikasi *wesbsite* dengan menggunakan Pyhton, *Javascript* dan *Node.Js,* aplikasi *mobile* menggunakan Kotlin*,* dan *Python..*

1. Pengujian Program

Pada tahap ini dilakukan proses pengujian dilakukan untuk memastikan perangkat lunak yang dibuat dapat berjalan sesuai dengan fungsionalitasnya.

1. Pengujian sistem rekomendasi deep collaborative filtering menggunakan Mean Absolute Error (MAE) dan Root Mean Square Error (RMSE). Pengujian dengan Mean Absolute Error (MAE) digunakan untuk mengukur keakuratan nilai rekomendasi dimana nilai dari hasil pengujian pada sistem rekomendasi akan menjadi parameter akurat atau tidaknya rekomendasi yang diberikan dan Root Mean Square Error (RMSE) mendapatkan hasil dengan mengkuadratkan error.
2. Pengujian fungsionalitas dari aplikasi perpustakaan, menggunakan metode *Black Box Testing* untuk menguji perangkat lunak apakah hasilnya sesuai dengan yang diharapkan.
3. Kesimpulan hasil pengujian

Penarikan kesimpulan diambil berdasarkan hasil pengujian yang telah dilakukan*.*

# 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, dan majalah berbasis kertas. Konsep perpustakaan digital ini mengurangi ruang penyimpanan buku, kemudahan pencarian database, kemudahan 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).

Sejak 1990, sistem rekomendasi telah menjadi bidang yang dipelajari secara ekstensif, karena telah menjadi teknologi kunci dalam aplikasi *e-commerce*. Sistem rekomendasi dalam perpustakaan digital merupakan permintaan yang spesifik, karena luasnya koleksi digital dan kebutuhan informasi yang beragam membuat umpan balik yang efektif dari *user* diperlukan. Berbagai sistem rekomendasi telah dikembangkan seperti *content-based*, *collaborative filtering*, dan *hybrid systems*. Sistem ini bekerja berdasarkan histori *user*, seperti catatan pinjaman, *session* data atau umpan balik dari *user*. Namun, informasi umpan balik jarang dilakukan oleh *user* sehingga mempengaruhi kualitas rekomendasi (Wenige & Ruhland, 2018).

### *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 keras *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 **Invalid source specified.**.

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***Invalid source specified.**.

### 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 (Rikowski, 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) (Suharyanto, 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). Saat ini sistem rekomendasi telah diterapkan di berbagai domain seperti musik, film, buku, dan produk. Fungsi tersebut tidak hanya dapat merekomendasikan produk yang disukai *user* tetapi juga dapat secara aktif mempromosikan pemasaran produk. Dasar dari sistem rekomendasi dibangun dengan asumsi bahwa riwayat user Internet dapat sepenuhnya mewakili preferensi *user*, dan asumsi ini berarti bahwa preferensi *user* dapat diperoleh dengan *data mining* dan analisis riwayat *user* (Zhang & Yang, 2019).

Penggunaan teknik rekomendasi yang akurat dan efisien sangat penting bagi suatu sistem untuk memberikan hasil yang baik dan bermanfaat kepada setiap penggunaanya. Salah satu masalah yang menyebabkan sistem rekomendasi tidak akurat adalah karena terjadinya *sparsity*. *Sparsity* merupakan masalah yang terjadi akibat kurangnya informasi yang cukup, yaitu hanya sedikit dari jumlah *item* yang tersedia dalam database yang dinilai oleh *user.* Akibatnya, *user* tidak mendapatkan hasil rekomendasi yang baik dan bermanfaat (Isinkaye, et al., 2015).

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 *user* secara langsung, data dapat diperoleh dengan meminta pendapat langsung dari *user*, bisa berupa rating, *likes/dislikes*, maupun kata pencarian (*keyword*). Metode Implisit merupakan pengumpulan data berdasarkan pengamatan pola kecenderungan *user*, seperti kecenderungan *user* terhadap suatu *item* dengan kriteria tertentu yang dimiliki *item* tersebut. Metode pengumpulan data pada teknik implisit tidak melibatkan kontribusi *user* melainkan dengan cara pendekatan dari perilaku *user* dan menyimpulkan data-data yang sudah dikumpulkan oleh sistem. Berdasarkan pendekatan yang umum dalam sistem rekomendasi dibagi menjadi tiga klasifikasi yaitu(Lü, et al., 2012):

### 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 :

(𝑢,) merupakan prediksi rating *user u* terhadap *item i.*

𝐷𝑖 merupakan fitur yang muncul di dalam *item i*.

Sistem rekomendasi content-based memiliki beberapa kelebihan, yaitu:

1. Sistem rekomendasi ini dapat menjelaskan bagaimana hasil rekomendasi didapatkan.
2. Sistem rekomendasi ini dapat merekomendasikan *item-item* yang bahkan belum pernah di 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 Filtering Recommendation

Sistem rekomendasi collaborative filtering memberikan rekomendasi item berdasarkan kumpulan user dengan preferensi yang sama atau similarity dari setiap pengguna. Proses collaborative filtering dilakukan dengan melihat persamaan rating pada pekerjaan dengan metode user to user. Similarity antara pengguna dapat dihitung berdasarkan dari rating yang diberikan pada oleh *user*. Metode collaborative filtering juga sering disebut sebagai “people-to-people correlation” (Rahmawati, et al., 2018). Pendekatan *collaborative filtering* 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 berdasarkan kesukaan *user* di masa lalu dan pandangan dari *user* lain yang berpikiran sama. Terdapat dua tugas yang dapat dilakukan oleh *collaborative filtering* yakni (Nilashi, et al., 2013) :

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

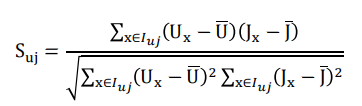
*Collaborative filtering* beroperasi di dalam sebuah *matrix rating*. *Rating* yang diberikan oleh *user* terhadap *item* direpresentasikan sebagai R dan nilai *rating* tersebut merupakan bilangan bulat tidak negatif atau bilangan real dengan jarak tertentu. *Collaborative filtering* mencoba memprediksi *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 |

Proses pada collaborative filtering dimulai dengan menghitung similarity antar pengguna yang memiliki ketertarikan yang sama dengan menggunakan persamaan (Rahmawati, et al., 2018):



Keterangan:

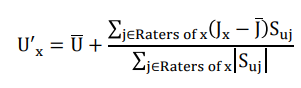
- S adalah similarity antara *user* u dan *user* j

- U adalah interaksi *user* u pada pekerjaan x

- U adalah rata-rata interaksi dari *user* u

- J adalah interaksi dari *user* j pada pekerjaan x - J adalah rata rata interaksi dari pengguna j

Kemudian dihitung prediksi interaksi menggunakan persamaan (Rahmawati, et al., 2018):



Keterangan:

- U’x adalah prediksi interaksi pengguna u pada pekerjaan x

- U adalah rata rata interaksi dari pengguna u

- J adalah interaksi dari pengguna j pada pekerjaan x

- S adalah similarity antara pengguna u dan pengguna j

Pada praktiknya sistem rekomendasi *collaborative filtering* mengalami berbagai macam masalah. Salah satu masalahnya yaitu *sparsity* matriks *rating*. *Sparsity* data mengacu pada masalah data yang tidak mencukupi pada matriks *rating*. Di Sebagian besar sistem rekomendasi, banyak *user* hanya menilai sejumlah kecil *item* sehingga membuat ukuran kesamaan memiliki akurasi yang sangat rendah (Saeed & Mansoori, 2017). Sistem rekomendasi *collaborative filtering* memiliki beberapa kekurangan, diantaranya adalah (Lucas, et al., 2013):

1. Hasil rekomendasi yang tidak andal yang dikarenakan matrix *rating* yang jarang. (*Sparsity Problem*).
2. 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*)
3. Hasil rekomendasi tidak selamanya andal. Terdapat kemungkinan adanya sebuah *user* yang hanya memiliki sedikit neighbor dengan tingkat kesamaan (similarity) yang kecil. (*Grey Sheep Problem*)

### Hybrid approaches

Masing-masing teknik sistem rekomendasi memiliki kelebihan dan kelemahannya tersendiri. Sehingga Algoritma *Hybrid Recommendation* *System* yang menggabungkan kekuatan dari model dan algoritma yang berbeda untuk mengatasi dan menyelesaikan masalah-masalah yang telah disebutkan di atas dan 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 *Voting* (Li & Kim , 2003).

1. Penggabungan secara Sekuensial (*Sequential Combination*)

Penggabungan ini adalah melakukan perhitungan pada salah satu Pendekatan filtering (misalkan *content-based*) kemudian hasilnya digabungkan dengan pendekatan lainnya (misalkan *collaborative*) (Li & Kim , 2003).

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

Penggabungan ini menggabungkan informasi item dan rating pengguna untuk menghitung kemiripan *item-item*. Metode *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*) (Li & Kim , 2003).

## 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. (Li, et al., 2015).

Gambar 2. 1 Contoh *Matrix factorization-based*

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



## Deep Learning

*Deep learning* adalah bentuk *machine learning* yang memungkinkan komputer untuk belajar dari pengalaman dan memahami dunia dalam hal hierarki konsep. Karena komputer mengumpulkan pengetahuan dari pengalaman, sehingga operator komputer manusia tidak diperlukan untuk menentukan semua pengetahuan yang dibutuhkan oleh komputer. Hierarki konsep memungkinkan komputer untuk mempelajari konsep yang rumit dengan membangunnya menjadi konsep yang lebih sederhana. grafik hierarki ini akan memiliki banyak lapisan (Kwang Gi Kim, 2016).

*Deep learning* telah berhasil diterapkan di banyak domain penelitian berbeda, seperti *computer vision*, *speech recognition*, *natural language processing* dan lain sebagainya.Pada bidang ini, dibandingkan dengan pendekatan tradisional, pendekatan *deep learning* meningkatkan kinerja secara luar biasa. Karena keberhasilan besar deep learning, beberapa peneliti mencoba menggunakannya dalam sistem rekomendasi dan berharap dapat meningkatkan kinerja sistem rekomendasi, seperti halnya penerapan model deep learning di bidang penelitian lain (Liu & Wu, 2017). Dalam satu dekade terakhir, banyak sekali penelitian yang mencoba untuk memperkenalkan metode deep learning ke dalam sistem recommendasi untuk meningkatkan kinerja, hasilnya metode ini dapat memperoleh hasil rekomendasi yang memuaskan berbeda dengan sistem rekomendasi tradisional. Misalnya, *Restricted Boltzmann Machines* (RBM) yang terdiri dari *hidden layer* dan *visible layer* (Zhou, et al., 2018)*.*

Salah satu teknik yang digunakan pada deep learning adalah *auto-encoder. Auto-encoder* merupakan metode *unsupervised learning* yang mencobamerekonstruksi masukan data di lapisankeluaran secara umum, lapisan *bottleneck* (lapisan paling tengah) digunakan sebagairepresentasi fitur menonjol dari datamasukan. Ada banyak varian dari *auto-encoders* seperti *denoising auto-encoder, marginalized denoising auto-encoder, sparse auto-encoder, kontraktif auto-encoder* dan *variational auto-encode*r (VAE). Teknik *deep learning* memungkinkan model untuk secara otomatis mempelajari fitur bagi user dan *item* dari sumber daya yang berbeda. Fitur ini digeneralisasi dengan baik dan dapat digunakan secara efektif untuk meningkatkan kualitas rekomendasi. Dengan menggunakan *deep learning* dalam pemodelan berbagai tipe data, sistem rekomendasi akandapat lebih memahami apa yang dibutuhkan *user* dan hal tersebut akan meningkatkan hasil rekomendasi (Ikasari, et al., 2018).

## 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 *item*, dll. *Deep collaborative filtering* yang *menjembatani matrix factorization* dan *deep feature learning* merupakan *hybrid collaborative filtering model*. *Deep collaborative Filtering* menggabungkan *probabilistic matrix factorization* dan *marginalized denoising auto-encoders* (mDA). *Probabilistic matrix factorization* adalah pendekatan *collaborative filtering* yang diterapkan secara luas dengan kinerja yang sangat baik, dan *marginalized denoising auto-encoders* adalah alat yang ampuh dalam mengekstraksi fitur tingkat tinggi dari input. Kombinasi keduanya memanfaatkan fungsinya masing-masing untuk mempelajari model yang lebih kompleks (Li, et al., 2015).

|  |  |
| --- | --- |
| 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* |

Table 2.2 – Ringkasan Notasi

Sumber (Li, et al., 2015)



Gambar 2.2 Ilustrasi penerapan deep collaborative filtering

### 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. Side information merupakan data yang diperoleh dari profil user untuk mengatasi masalah sparsity dalam rekomendasi, adapun data dari side information seperti demografik *user*, *item rivews* dan deskripsi dsb (Han, et al., 2019). 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 l*atent 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***

## Pengujian Perangkat Lunak dan Pengujian Keakuratan Hasil Rekomendasi

Perangkat lunak memiliki bug dan hal itu tidak bisa untuk dihindari. Kode ditulis oleh manusia, dan manusia dapat membuat kesalahan. Kebutuhan dapat ambigu atau salah, Kebutuhan dapat disalahpahami, komponen perangkat lunak dapat disalahgunakan, pengembang dapat membuat kesalahan saat menulis kode, dan bahkan kode yang pernah berfungsi mungkin tidak lagi benar ketika asumsi yang sebelumnya valid menjadi tidak lagi berlaku setelah perubahan. Pengujian perangkat lunak adalah respons intuitif untuk masalah ini. Setelah membangun dan menjalankan sistem, dilakukan pemeriksaan apakah sistem berfungsi seperti yang diharapkan (Fraser, et al., 2019). Terdapat 2 pengujian yang akan dilakukan antara lain:

### Pengujian Perangkat Lunak dengan Black Box Testing

*Black Box Testing* memainkan peran penting dalam pengujian perangkat lunak, ini membantu dalam validasi fungsionalitas keseluruhan sistem. *Black Box Testing* dilakukan berdasarkan kebutuhan pelanggan sehingga setiap persyaratan yang tidak lengkap atau tidak terduga dapat dengan mudah diidentifikasi dan diatasi. *Black Box Testing* dilakukan berdasarkan perspektif *user* akhir. Pentingnya *Black Box Testing* adalah menangani masukan yang valid dan tidak valid dari sudut pandang pelanggan (Nidhra, et al., 2012).

Keuntungan utama dari *black box testing* adalah, penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu, tidak hanya bahasa pemrograman tetapi juga pengetahuan tentang implementasi. Dalam *black box testing*, baik pemrogram maupun penguji tidak bergantung satu sama lain. Keuntungan lainnya adalah pengujian dilakukan dari sudut pandang *user*. Keuntungan signifikan dari *black box testing* adalah membantu mengungkap ambiguitas atau ketidakkonsistenan dalam spesifikasi persyaratan (Nidhra, et al., 2012).

### 2.6.2 Pengujian Keakuratan Hasil Rekomendasi

Sistem rekomendasi diimplementasikan untuk membantu *user* dalam mengenali informasi yang diinginkan. Akurasi adalah salah satu metrik kinerja yang banyak digunakan untuk rekomendasi untuk mengukur tingkat *error* antara peringkat aktual dan prediksi. Ini juga merupakan kriteria paling umum yang digunakan untuk mengevaluasi keberhasilan sistem pemberi rekomendasi baik yang terkait dengan prediksi rating maupun rekomendasi. 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. Metode perhitungan *error* yang digunakan yaitu *Mean Absolute Error (MAE)* dan *Root Mean Square Error* (RMSE) yang disebut Metrik Akurasi Prediksi untuk mengevaluasi kesesuaian dengan *rating* *user* sebenarnya dari prediksi sistem pemberi rekomendasi (Nilashi, et al., 2013).

* + - 1. **Mean Absolute Error (MAE)**

MAE adalah metrik yang umumnya digunakan untuk pengukuran akurasi prediksi dan jauh lebih banyak digunakan daripada metrik lainnya. MAE ditentukan sebagai deviasi absolut rata-rata antara peringkat yang diprediksi dan peringkat sebenarnya. Persamaan MAE yaitu sebagai berikut (Nilashi, et al., 2013):

Dimana*,*

*pi* = Nilai prediksi *item* ke i

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

N = jumlah *item* yang dihitung

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

* + - 1. Root Mean Square Error (RMSE)

Root Mean Squared Error (RMSE) adalah metrik lain dalam akurasi prediksi. yaitu menghitung statistik akurasi. Secara konsep ini mirip dengan MAE, tetapi RMSE mendapatkan hasil dengan mengkuadratkan error. Persamaan RMSE yaitu sebagai berikut (Nilashi, et al., 2013).

Dimana,

𝑓𝑡 = Permintaan aktual periode *t*

𝑓̂ = Ramalan permintaan periode *t*

m = Jumlah periode peramalan

# DAFTAR PUSTAKA

Aithal, D. P. S., 2016. SMART LIBRARY MODELS FOR FUTURE GENERATIONS.

Aleksandrova, M. e. a., 2017. Identifying representative users in matrix factorization-based recommender systems: application to solving the content-less new item cold-start problem. *J Intell Inf Syst.*

Bo Yang, Y. L. J. L. a. W. L., 2016. Social Collaborative Filtering by Trust. p. 1.

Bobadilla, J., Alonso, S. & Hernando, A., 2020. Deep Learning Architecture for Collaborative Filtering Recommender Systems. p. 1.

Bobadilla, J. e. a., 2017. Recommender Systems Clustering Using BayesianNon Negative Matrix Factorization. *Digital Object Identifier 10.1109/ACCESS.2017.2788138.*

Candillier, L., Jack, K., Fessant, F. & Meyer, F., 2009. State of the Art Recommender Systems. *Research Gate.*

Chen, J. et al., 2020. Deep attention user-based collaborative filtering for recommendation. p. 2.

Elahi, M., 2016. A survey of active learning in collaborative filtering recommender systems. *C O M P U T E R S C I E N C E R E V I E W2 0 ( 2 0 1 6 ).*

Faniband, Y. P., Ishak, I., Sidi, F. & Jabar, M. A., 2020. A Cloud-Based Distributed Platform for Secured EPUB EBOOK Contents.

Francesco Ricci, L. R. B. S., 2011. Recommender System Handbook. *Kantor, P. B..*

Guabassi, I. E., 2016. Recommender system for ubiquitous learning based on decision tree. *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt).*

Ikasari, D., Suhendra, A. & Farida, N., 2018. Metode Deep Learning Pada Sistem Rekomendasi : Review Paper. Volume 2, p. 47.

Isinkaye, F., Folajimi, Y. & Ojokoh, B., 2015. Recommendation systems: Principles, methods and. p. 4.

Khan, M. E., 2011. DIFFERENT APPROACHES TO BLACK BOX TESTING TECHNIQUE FOR FINDING ERRORS.

Khiste, G. P., Deshmukh, R. K. & Awate, A. P., 2018. Literature Audit of 'Digital Library': an Overview.

Laurent Candillier, K. J. F. F. F. M., 2009. State of the Art Recommender Systems. *Research Gate.*

Li, S., Kawale, J. & Fu, Y., 2015. *Deep Collaborative Filtering via Marginalized DenoisingAuto-encoder.* s.l.:s.n.

Li, S., Kawale, J. & Fu, Y., 2015. Deep Collaborative Filtering via Marginalized Denoising Auto-encoder. pp. 812, 814.

Liu, J. & Wu, C., 2017. Deep Learning Based Recommendation: A Survey. Volume 424, p. 451.

Lü, L. et al., 2012. Recommendersystems. *PhysicsReports519,* p. 39.

Manley, L. & Holley, R. P., 2012. History of the Ebook: The Changing Face of Books.

Mishra, N. et al., 2017. Solving Sparsity Problem in Rating-BasedMovie Recommendation System. *Computational Intelligencein Data Mining,* p. 113.

Moh. Irfan, A. D. C. F. H. R., 2014. Sistem Rekomendasi: Buku Online dengan Metode Collaborative Filtering. *Jurnal Teknologi Technoscienta,* pp. 77-78.

Nahak, B. & Padhi, S., 2019. The Role of Smart Library and Smart Librarian for E- Library Services.

Rahmawati, S., Nurjanah, D. & Rismala, R., 2018. Analisis dan Implementasi Pendekatan Hybrid untuk Sistem Rekomendasi dengan Metode Knowledge Based Recommender System dan Collaborative Filtering. *Ind. Journal on Computing,* 3(2), p. 12.

Rao, V., V, R. K. & Padmanabhan, V., 2017. Divide and Transfer: Understanding Latent Factors for Recommendation Tasks. p. 1.

Rikowski, R., 2012. *Library Classification Trends in the 21st Century.* s.l.:s.n.

Saeed, M. & Mansoori, E. G., 2017. A NOVEL FUZZY-BASED SIMILARITY MEASURE FOR COLLABORATIVE FILTERING TO ALLEVIATE THE SPARSITY PROBLEM. *Journal of Fuzzy Systems,* Volume 14, p. 2.

Saranya, K. G., Sadasivam , . G. S. & Chandralekha, M., 2016. Performance Comparison of Different Similarity Measures for Collaborative Filtering Technique. *Indian Journal of Science and Technology,* Volume 9, p. 2.

Sargeant, B., 2015. What is an ebook What is a Book App And Why Should We Care An Analysis of Contemporary Digital Picture Books.

Satija, M., 2008. *Universal Decimal Classification: Past and Present.* s.l.:s.n.

Satyaputra, M. &. E. M. A. S. K., 2016. *Lets Build Your Android Apps with Android Studio.* s.l.:Elex Media Komputindo.

Shi, Q., 2010. An Empirical Study Of Thinking Aloud Usability Testing From a Cultural Perspective.

Steven Hendrawan, e. a., 2020. Perancangan Sistem Informasi Permohonan Perizinan Penelitian dengan Metode Agile dan Framework Laravel Berbasis W. *Journal of Information Systems and Informatics.*

Suharyanto, 2012. *Dewey Decimal Classification Edisi Ke-23 : Perubahan dan Perluasan Notasi Tentang Indonesia.* s.l.:s.n.

Suresha, G. & Narayanaswamy, B., 2016. *The Scheme of Library Classifications: Concerning the Structural changes of 23rd Dewey Decimal Classification (DDC).* s.l.:s.n.

Uluyagmur, M., Cataltepe , Z. & Tayfur, . E., 2012. Content-Based Movie Recommendation Using Different Feature Sets. *Proceedings of the World Congress on Engineering and Computer Science.*

Wenige, L. & Ruhland, J., 2018. Retrieval by recommendation: using LOD technologies to improve. *digital library search,* p. 2.

Yudhanto, Y. a. A. W., 2018. *Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio.* s.l.:Elex Media Komputindo.

Yudhanto, Y. a. A. W., 2018. *Mudah Membuat dan Berbisnis Aplikasi Android dengan Android Studio.* s.l.:Elex Media Komputindo.

Zhang, J.-D. & Chow, C.-Y., 2016. Enabling Kernel-based Attribute-aware Matrix Factorization for Rating Prediction. p. 1.

Zhang, M. & Yang, Z., 2019. GACOforRec: Session-Based Graph Convolutional Neural Networks Recommendation Model. p. 114077.

Zhang, Q. et al., 2018. CoupledCF: Learning Explicit and Implicit User-item Couplingsin Recommendation for Deep Collaborative Filtering. *International Joint Conferences on Artificial Intelligence.*

Zhang, S., 2019. *Deep Learning based Recommender System: A Survey and New Perspectives.*

Zulkarnain, T. D. A. d. A., 2013. SUGGESTIONS FRIENDS ENGINE BERBASIS HYBRID RECOMMENDER SYSTEM UNTUK MENDAPATKAN REKOMENDASI TEMAN TERBAIK PADA WEB JEJARING SOSIAL. *Jurnal Ilmiah Teknologi dan Informasi ASIA.*