# 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 *item* perpustakaan digital cenderung jarang diberikan oleh *user* (Wenige & Ruhland, 2018).

*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, sehingga membuat sistem rekomendasi akanlebih 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 matriks *factorization* dengan algoritma *deep learning*. *Deep collaborative filtering* yang merupakan model *hybrid*, menggunakan matriks *rating* dan *side information* serta mengintegrasikan matriks *factorization* dan fitur *learning* dengan memodelkan pemetaan antara *latent factors* yang digunakan dalam *collaborative filtering* dan *latent layers* dalam *deep models* (Li, et al., 2015).

*Deep collaborative filtering* dapat memberikan hasil yang lebih baik daripada *collaborative filtering* (Li, et al., 2015). Karena penerapan *deep learning* pada perpustakaan digital dapat membantu sistem memberikan rekomendasi lebih akurat (Li, et al., 2020). Oleh karena itu, *deep collaborative filtering* diharapkan dapat mengatasi masalah *sparsity* pada perpustakaan digital.

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

## Rumusan Masalah

Berdasarkan uraian latar belakang di atas, maka yang menjadi permasalahan pada penelitian ini adalah:

* 1. *User* jarang memberikan *rating* pada perpustakaan digital sehingga mengalami *sparsity* matriks *rating*.
  2. *User* tidak mendapat *item* yang disarankan atau rekomendasi yang dapat menjadi pertimbangan dalam menentukan *item* yang sesuai dengan kebutuhan dan keinginan *user*.

## 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 *e-book* pada 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. Hasil penerapan *deep collaborative filtering* dapat memberikan rekomendasi *item* digital yang sesuai dengan kebutuhan dan 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. *Mobile application* sebagai produk dari Tugas Akhir ini hanya dapat dijalankan pada *smartphone* dengan *operating system* android versi minimal Lollipop 5.0+.
3. Jumlah aktor dalam sistem ini terdiri dari 3 (tiga), yakni *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, *member* 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 populer *item* digital.
2. Mencari dan melihat rincian *item* digital.
3. Melakukan pendaftaran pada sistem.
4. Dataset yang digunakan sejumlah 100 data buku digital yang diperoleh dari [www.kaggle.com](https://www.kaggle.com/ruchi798/bookcrossing-dataset) yang berisi metadata produk buku pada *Book-Crossing: User review ratings* pada agustus 2020 yang dikumpulkan oleh Ruchi Bhatia. Dataset terdiri dari 3 *file* yaitu *file* *BX\_Books* (.csv) dengan atribut *ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL,* *file BX-Book-Ratings* (.csv) dengan atribut *User-ID, ISBN, Book-Rating,* dan *file BX-Users* (.csv) dengan atribut *User-ID, Location, Age* dengan jumlah 1,149,780 *metadata.*
5. Format *item* digital yang digunakan berupa Portable Document Format (PDF).
6. *Source code* algoritma *deep collaborative filtering* diperoleh dari <https://github.com/smayru/mDACF/commits/master>.

## Metodologi Pengembangan Sistem

Metodologi yang digunakan dalam penyusunan tugas akhir ini menggunakan model *waterfall* dengan setiap fase sebagai berikut:

1. Analisis Proses

Melakukan perhitungan secara manual dengan contoh masalah yang sederhana dan menggunakan *flowchart* untuk menganalisis proses algoritma yang digunakan.

1. Analisis Kebutuhan

Melakukan analisis kebutuhan fungsional dan kebutuhan *non*-fungsional. Untuk kebutuhan fungsional menggunakan *use case diagram*, kebutuhan *non-*fungsional memanfaatkan *PIECES* (*Performance*, *Information, Economy, Control, Efficiency, Service*).

1. Perancangan

Melakukan perancangan tampilan (interface) menggunakan software Adobe XD, pemodelan basis data menggunakan Microsoft Visio, yang akan diimplementasikan ke dalam basis data menggunakan *firebase*.

1. Implementasi

Melakukan penulisan kode program, aplikasi *website* menggunakan bahasa pemrograman *PHP (Hypertext Preprocessor), Javascript* dan *Python*, dan aplikasi *mobile* menggunakan bahasa pemrograman *Kotlin.*

1. Pengujian Program

Melakukan pengujian untuk memastikan perangkat lunak yang dibuat dapat berjalan sesuai dengan fungsionalitasnya.

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

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