**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](#_Toc73923078)

[DAFTAR GAMBAR iii](#_Toc73923079)

[DAFTAR TABEL v](#_Toc73923080)

[BAB I PENDAHULUAN 1](#_Toc73923081)

[1.1 Latar Belakang 1](#_Toc73923082)

[1.2 Rumusan Masalah 2](#_Toc73923083)

[1.3 Tujuan 2](#_Toc73923084)

[1.4 Manfaat 2](#_Toc73923085)

[1.5 Batasan Masalah 2](#_Toc73923086)

[1.6 Metodologi Pengembangan Sistem 3](#_Toc73923087)

[BAB II TINJAUAN PUSTAKA 5](#_Toc73923088)

[2.1 Perpustakaan Digital 5](#_Toc73923089)

[2.1.1 Jenis Koleksi Digital 5](#_Toc73923090)

[2.1.2 Format File Digital 6](#_Toc73923091)

[2.1.3 Digitalisasi File Digital 7](#_Toc73923092)

[2.1.4 Klasifikasi Item Perpustakaan 9](#_Toc73923093)

[2.2 Sistem Rekomendasi 10](#_Toc73923094)

[2.2.1 Content-based Recommendation 11](#_Toc73923095)

[2.2.2 Collaborative Filtering Recommendation 13](#_Toc73923096)

[2.2.3 Hybrid Approaches 15](#_Toc73923097)

[2.3 Deep Learning 16](#_Toc73923098)

[2.3.1 Multilayer Perceptron 17](#_Toc73923099)

[2.3.2 Autoencoder Based Recommendation 17](#_Toc73923100)

[*2.3.3* Restricted Boltzmann Machine based Recommendation 18](#_Toc73923101)

[2.4 Deep Collaborative Filtering 19](#_Toc73923102)

[2.4.1 Matriks Factorization 20](#_Toc73923103)

[2.4.2 Marginalized Denoising Auto-encoder 22](#_Toc73923104)

[2.4.3 Normalisasi Data Pada Deep Collaborative Filtering 22](#_Toc73923105)

[2.4.4 Menghitung Kesalahan Model (*Loss Function*) 24](#_Toc73923106)

[2.4.5 Tahapan Rekomendasi Pada Deep Collaborative Filtering 24](#_Toc73923107)

[2.5 Pengujian Perangkat Lunak dan Pengujian Keakuratan Hasil Rekomendasi 28](#_Toc73923108)

[2.5.1 Pengujian Perangkat Lunak dengan Black Box Testing 28](#_Toc73923109)

[2.5.2 Pengujian Keakuratan Hasil Rekomendasi 28](#_Toc73923110)

[BAB III ANALISIS DAN PERANCANGAN 30](#_Toc73923111)

[3.1 Analisis 30](#_Toc73923112)

[3.1.1 Analisis Proses 30](#_Toc73923113)

[3.1.1.1 Deep Collaborative Filtering 30](#_Toc73923114)

[3.1.2 Analisis Kebutuhan 46](#_Toc73923115)

[3.1.2.1 Analisis Kebutuhan Fungsional 46](#_Toc73923116)

[3.1.2.2 Analisis Kebutuhan *Non*-Fungsional 58](#_Toc73923117)

[3.2 Perancangan 59](#_Toc73923118)

[3.2.1 Perancangan Tampilan 59](#_Toc73923119)

[3.2.1.1 Tampilan Website 59](#_Toc73923120)

[3.2.1.2 Tampilan Mobile 77](#_Toc73923121)

[3.2.2 Perancangan Basis Data 89](#_Toc73923122)

[DAFTAR PUSTAKA 96](#_Toc73923123)

# DAFTAR GAMBAR

[Gambar 2.2 Ilustrasi Penerapan *Deep Collaborative Filtering* 18](#_Toc73106191)

[Gambar 2.1 Contoh *Matriks Factorization* 19](#_Toc73106192)

[Gambar 2.3 Perkalian Matriks *Latent Factor User* dan *Item* 24](#_Toc73106193)

[Gambar 3.1 *Flowchart* dari *Deep Collaborative Filtering* 28](#_Toc73125514)

[Gambar 3.2 *Use case Diagram Web* 45](#_Toc73125515)

[Gambar 3.3 *Use case Diagram Mobile* 45](#_Toc73125516)

[Gambar 3.4 Perancangan Halaman Utama *Website* 58](#_Toc73125517)

[Gambar 3.5 Perancangan Halaman Pendaftaran 60](#_Toc73125518)

[Gambar 3.6 Perancangan Halaman Masuk 61](#_Toc73125519)

[Gambar 3.7 Perancangan Halaman Beranda *Member* 62](#_Toc73125520)

[Gambar 3.8 Perancangan Halaman Profil 63](#_Toc73125521)

[Gambar 3.9 Perancangan Halaman Kategori *Item* Digital 64](#_Toc73125522)

[Gambar 3.10 Perancangan Halaman Pencarian 65](#_Toc73125523)

[Gambar 3.11 Rancangan Halaman Detail *Item* Digital 66](#_Toc73125524)

[Gambar 3.12 Perancangan Halaman Baca *Item* Digital 67](#_Toc73125525)

[Gambar 3.13 Rancangan Halaman Tentang 68](#_Toc73125526)

[Gambar 3.14 Rancangan Halaman Dashboard Admin 69](#_Toc73125527)

[Gambar 3.15 Rancangan Halaman *Profil* *Admin* 70](#_Toc73125528)

[Gambar 3.16 Rancangan Halaman *Item* Digital 70](#_Toc73125529)

[Gambar 3.17 Rancangan Halaman Tambah *Item* digital 71](#_Toc73125530)

[Gambar 3.18 Rancangan Halaman Data *Member* 72](#_Toc73125531)

[Gambar 3.19 Rancangan Halaman Verifikasi *Member* Baru 73](#_Toc73125532)

[Gambar 3.20 Rancangan Halaman Peminjaman 73](#_Toc73125533)

[Gambar 3.21 Rancangan Halaman Laporan 74](#_Toc73125534)

[Gambar 3.22 Rancangan Pengujian Keakuratan Sistem Rekomendasi 75](#_Toc73125535)

[Gambar 3.23 Rancangan Halaman Masuk *Mobile* 76](#_Toc73125536)

[Gambar 3.24 Rancangan Halaman Pendaftaran *Mobile* 77](#_Toc73125537)

[Gambar 3.25 Rancangan Halaman Beranda *Mobile* 78](#_Toc73125538)

[Gambar 3.26 Rancangan Halaman Pencarian *Mobile* 79](#_Toc73125539)

[Gambar 3.27 Rancangan Halaman Detail *Item* Digital *Mobile* 80](#_Toc73125540)

[Gambar 3.28 Rancangan Halaman Baca *Item* Digital *Mobile* 81](#_Toc73125541)

[Gambar 3.29 Rancangan Halaman Kategori *Item* Digital *Mobile* 82](#_Toc73125542)

[Gambar 3.30 Rancangan Halaman *Item* Digital Saya *Mobile* 83](#_Toc73125543)

[Gambar 3.31 Rancangan Halaman Pemberitahuan *Mobile* 84](#_Toc73125544)

[Gambar 3.32 Rancangan Halaman Menu *Profil* *Mobile* 85](#_Toc73125545)

[Gambar 3.33 Rancangan Halaman *Profil Member* *Mobile* 86](#_Toc73125546)

[Gambar 3.34 Rancangan Halaman Riwayat Peminjaman *Mobile* 87](#_Toc73125547)

[Gambar 3.35 Relasi Antar Tabel Basis Data Perpustakaan Digital 88](#_Toc73125548)

# DAFTAR TABEL

[Tabel 2.1 Kelas Umum *Dewey Decimal Classification* 23 9](#_Toc73106281)

[Tabel 2.2 Kelas Utama *Universal Decimal Classification* 10](#_Toc73106282)

[Tabel 2.3 Contoh Matriks *Rating* *Content-based Recommendation* 12](#_Toc73106283)

[Tabel 2.4 Contoh Matriks *User* x *Item* Sistem Rekomendasi *Collaborative Filtering* 14](#_Toc73106284)

[Tabel 2.5 Ringkasan Notasi 18](#_Toc73106285)

[Tabel 2.6 Data Sebelum Normalisasi 21](#_Toc73106286)

[Tabel 2.7 Data Setelah Normalisasi 21](#_Toc73106287)

[Tabel 2.8 Matriks *Rating* Hasil Prediksi 25](#_Toc73106288)

[Tabel 2.9 Matriks *Rating* Prediksi Setelah Normalisasi 25](#_Toc73106289)

[Tabel 3.1 Contoh Inisialisasi Matriks *Rating* (R) 29](#_Toc73106290)

[Tabel 3.2 Contoh Inisialisasi Fitur *User* (X) 29](#_Toc73106291)

[Tabel 3.3 Contoh Inisialisasi Fitur *Item* (Y) 29](#_Toc73106292)

[Tabel 3.4 Contoh Inisialisasi *Mapping Funtion User* (W1) 29](#_Toc73106293)

[Tabel 3.5 Contoh Inisialisasi *Mapping Funtion* *Item* (W2) 30](#_Toc73106294)

[Tabel 3.6 Contoh Inisialisasi Matriks *Projection* *User* (P1) 30](#_Toc73106295)

[Tabel 3.7 Contoh Inisialisasi Matriks *Projection Item* (P2) 30](#_Toc73106296)

[Tabel 3.8 *Latent Factor User* (U) 30](#_Toc73106297)

[Tabel 3.9 *Latent Factor Item* (V) 30](#_Toc73106298)

[Tabel 3.10 Contoh Inisialisasi Matriks *Rating Non-empty Entries* (A) 31](#_Toc73106299)

[Tabel 3.11 Matriks *Rating* Setelah Normalisasi 31](#_Toc73106300)

[Tabel 3.12 Hasil Perkalian Matriks dan 32](#_Toc73106301)

[Tabel 3.13 Hasil Perkalian Matriks W1 dan X 32](#_Toc73106302)

[Tabel 3.14 Hasil Pengurangan dan 32](#_Toc73106303)

[Tabel 3.15 Hasil Pengkuadratan Matriks . 32](#_Toc73106304)

[Tabel 3.16 Hasil Perkalian Matriks dan 33](#_Toc73106305)

[Tabel 3.17 Hasil Perkalian Matriks W2 dan Y 33](#_Toc73106306)

[Tabel 3.18 Hasil Pengurangan dan 33](#_Toc73106307)

[Tabel 3.19 Hasil Pengkuadratan Matriks . 33](#_Toc73106308)

[Tabel 3.20 Hasil Perkalian Matriks U dan 34](#_Toc73106309)

[Tabel 3.21 Hasil Pengurangan Matriks *rating* R dengan . 34](#_Toc73106310)

[Tabel 3.22 Hasil Perhitungan 34](#_Toc73106311)

[Tabel 3.23 Hasil Perkalian Matriks 35](#_Toc73106312)

[Tabel 3.24 Hasil Perhitungan 35](#_Toc73106313)

[Tabel 3.25 Nilai Hasil Penjumlahan dan 36](#_Toc73106314)

[Tabel 3.26 Hasil Perkalian Matriks 36](#_Toc73106315)

[Tabel 3.27 Hasil Perhitungan 36](#_Toc73106316)

[Tabel 3.28 Nilai Hasil Penjumlahan dan 36](#_Toc73106317)

[Tabel 3.29 Hasil *Update* W1 Perulangan 1 37](#_Toc73106318)

[Tabel 3.30 Hasil Perkalian Matriks 37](#_Toc73106319)

[Tabel 3.31 Hasil Perhitungan 37](#_Toc73106320)

[Tabel 3.32 Hasil Perkalian Matriks 38](#_Toc73106321)

[Tabel 3.33 Hasil Perhitungan 38](#_Toc73106322)

[Tabel 3.34 Nilai Hasil Penjumlahan dan 38](#_Toc73106323)

[Tabel 3.35 Hasil *Update* W2 Perulangan 1 38](#_Toc73106324)

[Tabel 3.36 Hasil Perhitungan 39](#_Toc73106325)

[Tabel 3.37 Hasil Perhitungan 39](#_Toc73106326)

[Tabel 3.38 Hasil *Update* P1 Perulangan 1 39](#_Toc73106327)

[Tabel 3.39 Hasil Perhitungan 39](#_Toc73106328)

[Tabel 3.40 Hasil Perhitungan 40](#_Toc73106329)

[Tabel 3.41 Hasil *Update* P2 Perulangan 1 40](#_Toc73106330)

[Tabel 3.42 *Latent Factor User* Perulangan 1 40](#_Toc73106331)

[Tabel 3.43 *Latent Factor Item* Perulangan 1 40](#_Toc73106332)

[Tabel 3.44 Hasil *Update* W1 Perulangan 2 41](#_Toc73106333)

[Tabel 3.45 Hasil *Update* W2 Perulangan 2 41](#_Toc73106334)

[Tabel 3.46 Hasil *Update* P1 Perulangan 2 41](#_Toc73106335)

[Tabel 3.47 Hasil *Update* P1 Perulangan 2 41](#_Toc73106336)

[Tabel 3.48 *Latent Factor User* Perulangan 2 42](#_Toc73106337)

[Tabel 3.49 *Latent Factor Item* Perulangan 2 42](#_Toc73106338)

[Tabel 3.50 Hasil *Update* W1 Perulangan 3 42](#_Toc73106339)

[Tabel 3.51 Hasil *Update* W2 Perulangan 3 43](#_Toc73106340)

[Tabel 3.52 Hasil *Update* P1 Perulangan 3 43](#_Toc73106341)

[Tabel 3.53 Hasil *Update* P1 Perulangan 3 43](#_Toc73106342)

[Tabel 3.54 *Latent Factor User* Perulangan 3 43](#_Toc73106343)

[Tabel 3.55 *Latent Factor Item* Perulangan 3 43](#_Toc73106344)

[Tabel 3.56 Matriks *Rating* Hasil Perkalian *Latent Factor User* dan *Item* 44](#_Toc73106345)

[Tabel 3.57 Hasil Normalisasi Matriks *Rating* Mengunakan *Min-max Normalization* 44](#_Toc73106346)

[Tabel 3.58 Aktor Pada *Use Case* *Diagram Web* dan *Mobile* 46](#_Toc73106347)

[Tabel 3.59 *Use Case* pada Web dan Mobile 46](#_Toc73106348)

[Tabel 3.60 Narasi *Use Case* *Login* 48](#_Toc73106349)

[Tabel 3.61 Narasi *Use Case* MengelolaData *Member* 48](#_Toc73106350)

[Tabel 3.62 Narasi *Use Case* Melakukan Verifikasi *Member* 49](#_Toc73106351)

[Tabel 3.63 Narasi *Use Case* Mengelola Data Pustaka 49](#_Toc73106352)

[Tabel 3.64 Narasi *Use Case* Mengelola Transaksi Peminjaman 50](#_Toc73106353)

[Tabel 3.65 Narasi *Use Case* Membuat Laporan 50](#_Toc73106354)

[Tabel 3.66 Narasi *Use Case* Mendapat Rekomendasi Koleksi dari Algoritma DCF 51](#_Toc73106355)

[Tabel 3.67 Narasi Use Case Meminjam Koleksi 51](#_Toc73106356)

[Tabel 3.68 Narasi *Use Case* Membaca Koleksi 52](#_Toc73106357)

[Tabel 3.69 Narasi *Use Case* Memberi Ulasan & *Rating* Koleksi 52](#_Toc73106358)

[Tabel 3.70 Narasi *Use Case* Mendaftar Sebagai *Member* 53](#_Toc73106359)

[Tabel 3.71 Narasi *Use Case* Mendaftar Sebagai *Member* 54](#_Toc73106360)

[Tabel 3.72 Narasi *Use Case* Melihat Detail Koleksi 54](#_Toc73106361)

[Tabel 3.73 Narasi *Use Case* Menyaring dan Mencari Koleksi 54](#_Toc73106362)

[Tabel 3.74 Narasi *Use Case* Mengedit Profil *Member* 55](#_Toc73106363)

[Tabel 3.75 Narasi *Use Case* Melakukan Pengujian Rekomendasi 55](#_Toc73106364)

[Tabel 3.76 Analisis Kebutuhan Non-Fungsionalitas dengan Kerangka PIECES 56](#_Toc73106365)

[Tabel 3.77 Struktur Tabel *Member* 89](#_Toc73106366)

[Tabel 3.78 Struktur Tabel *Admin* 89](#_Toc73106367)

[Tabel 3.79 Struktur Tabel *Item* 90](#_Toc73106368)

[Tabel 3.80 Struktur Tabel *Rating* 91](#_Toc73106369)

[Tabel 3.81 Struktur Tabel Ulasan 91](#_Toc73106370)

[Tabel 3.82 Struktur Tabel Transaksi Meminjam 92](#_Toc73106371)

[Tabel 3.83 Struktur Tabel Pengembalian 92](#_Toc73106372)

[Tabel 3.84 Struktur Tabel Laporan 92](#_Toc73106373)

[Tabel 3.85 Struktur Tabel Pengujian 93](#_Toc73106374)

# 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, 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 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 *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 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, *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 *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. Kriteria aturan dalam aplikasi perpustakaan digital tentang peminjaman *item* digital yaitu, batas akhir penggunaan yaitu 1 minggu menggunakan model bisnis meminjam (rental) dengan ketentuan maksimal jumlah peminjaman aktif setiap *user* adalah 5 (lima) *item.*

## 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 Figma, pemodelan basis data menggunakan Microsoft Visio, yang akan diimplementasikan ke dalam basis data menggunakan server MySQL.

1. Implementasi

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

1. Pengujian Program

Melakukan pengujian 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 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.
2. Pengujian fungsionalitas dari aplikasi perpustakaan digital, 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, 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 *user* yang tersebar secara geografis. Objek digital yang dimaksud seperti teks, gambar, peta, suara, video, katalog, dan jurnal ilmiah (Khiste, et al., 2018). Setiap *user* 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 *rating* 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 *rating* dari *user*. Namun, pemberian *rating* pada perpustakaan digital jarang dilakukan oleh *user* sehingga mempengaruhi kualitas rekomendasi (Wenige & Ruhland, 2018).

### Jenis Koleksi Digital

Perubahan perpustakaan konvensional menjadi perpustakaan digital telah mengubah operasi, fungsi dan layanan perpustakaan (Kapoor & Tiwari, 2020). Perpustakaan digital kini berperan sebagai media informasi yang menyediakan jenis koleksi digital seperti:

1. E-book

*E-book* adalah publikasi buku dalam bentuk digital yang terdiri dari teks, gambar, atau keduanya dan dapat dibaca di komputer, *e-readers* atau perangkat elektronik lainnya. *E-book* disukai oleh *user* karena fitur-fitur yang ada di dalamnya, seperti ukuran font yang dapat diubah, membuat kutipan, tautan ke situs lain yang relevan, pencarian, *sharing*, dll (Kapoor & Tiwari, 2020).

1. E-Journal

Seperti makalah penelitian, komunikasi ilmiah, terbitan berkala ilmiah dalam bentuk elektronik merupakan *tools* yang berguna bagi para peneliti. *E-journal* berdampak tidak hanya pada perpustakaan tetapi juga pada penulis dan penerbit. Oleh karena itu, mayoritas *user* sekarang mengharapkan informasi terkini dan tepat waktu yang dapat diperoleh dengan mudah, cepat, tepat dan dari jarak jauh. Perpustakaan akademik dan perpustakaan khusus biasanya tidak dapat menolak *e-journal* dalam koleksinya. Terdapat jenis jurnal online lainnya seperti artikel akses terbuka yang tersedia di web untuk dilihat dan di *download* tanpa biaya (Kapoor & Tiwari, 2020).

1. E-Thesis

*E-Thesis* merupakan layanan yang sangat berguna bagi *user* atau sebagian besar peneliti dalam mengumpulkan data yang besar untuk mata pelajaran tertentu. Pengadaan dan pengaksesan *e-thesis* dapat mengurangi duplikasi pekerjaan penelitian dan memberikan wilayah penelitian kepada para *user* (Kapoor & Tiwari, 2020).

### Format File Digital

Terdapat banyak format *file* digital, di antaranya sebagai berikut:

1. PDF (*Portable Document Format*)

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 (Zeng, et al., 2016).

1. EPUB (*Electronic Publication*)

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* (Zeng, et al., 2016)*.*

1. HTML (*Hyper Text Markup Language*)

HTML merupakan tulang punggung dari *world wide web*. Ada banyak teks yang didistribusikan dalam format ini karena dapat digunakan pada berbagai platform komputer dengan sistem operasi yang berbeda-beda. HTML memiliki *style* utama CSS (*cascading style sheet*) dan dapat disisipi bahasa pemrograman lainnya seperti *PHP (Hypertext Preprocessor)*, *Perl*, *JavaScript*, *VBScript*, dll. Namun, HTML tidak bisa berdiri sendiri sebagai bahasa pemrograman web. HTML juga memiliki penggunaan *tag* yang banyak dan versi HTML yang baru belum tentu bisa dijalankan pada semua *browser* (Fitri & Syah, 2020).

1. AZW (*Amazon World*)

Format AZW merupakan format *file* milik amazon yang menyerupai format MOBI dengan menyertakan DRM (*digital rights management*) yang di khususkan untuk *kindle* *amazon* yang tidak di dukung secara luas di *e-reader* (Fitri & Syah, 2020)*.*

### Digitalisasi File Digital

Digitalisasi didefinisikan sebagai proses pengambilan bahan pustaka tradisional berupa buku dan kertas dan mengubahnya menjadi bentuk elektronik sehingga dapat disimpan dan dimanipulasi oleh komputer (Singh, 2017). Terdapat enam tahapan digitalisasi *file* digital yaitu:

1. *Registering*

Sebelum memindai dokumen perlu untuk mendaftarkannya terlebih dahulu dan menggunakan sistem pengarsipan untuk melacaknya. Ini berfungsi untuk menghindari risiko kehilangan *hardcopy*, kehilangan *file*, melewatkan langkah-langkah dalam proses atau menggandakan pekerjaan yang mungkin tanpa disadari. Ada juga risiko kehilangan *file* versi elektronik karena salah nama atau disimpan di *subdirectory* yang salah.

1. *Scanning* Dokumen

Sebelum melakukan pemindaian dokumen, bersihkan terlebih dahulu, pastikan semua halaman ada dan terurut, jika terdapat dokumen dalam kondisi buruk carilah salinan baru. Lakukan pemindaian sesuai dengan *tools* dan teknik pemindaian yang ada.

1. *Optical Character Recognition* (OCR)

OCR merupakan perangkat lunak yang berfungsi mengubah gambar yang dipindai menjadi *file* teks yang dapat dibaca oleh *software* pengolah kata. OCR akan memecah blok teks menjadi beberapa baris atau menjadi karakter individu dan mencocokkan gambar dari setiap huruf dengan pola yang dikenali sebagai "a", "b", dll. Penggunaan OCR yang dikhususkan untuk menghindari masalah penggunaan bahasa skrip latin dengan karakter beraksen.

1. *Proofreading*

*Proofreading* merupakan tahapan koreksi teks dan tata letak dokumen. Ini dapat dilakukan dengan dua cara yaitu:

1. Membandingkan teks yang dipindai pada layar dengan *hardcopy* dan memasukkan koreksi langsung ke komputer. Pemeriksa ejaan pengolah kata akan membantu dalam kesalahan ejaan dengan cepat.
2. Mencetak teks pindaian dan membandingkannya dengan salinan aslinya. Tandai koreksi apapun pada hasil cetak, lalu masukkan ke dalam komputer. Ini adalah metode yang lebih lambat, tetapi mungkin merupakan opsi terbaik jika tidak ada cukup komputer untuk setiap korektor.
3. *Reformatting*

OCR dapat menghasilkan dokumen yang terdiri dari teks lurus, tanpa kolom, *header* dan *footer*. Untuk itu diperlukan pembaruan untuk mengoreksi dan memasukkannya ulang dalam halaman, atau juga adanya pembaharuan jenis huruf, *style* judul, dan sebagainya, untuk membuat dokumen lebih menarik dan mudah dibaca. Sebagai alternatif, sesuaikan pengaturan program OCR untuk mempertahankan tata letak halaman.

1. Versi Final

Kebanyakan dokumen memerlukan beberapa informasi tambahan agar pembaca dapat mengidentifikasinya dengan mudah. Contohnya, untuk sebuah buku diperlukan penambahan judul buku, penulis atau editor, penerbit, dan tanggal penerbitan pada dokumen terkait. Untuk artikel jurnal perlu mencantumkan judul jurnal, tanggal, volume dan nomor terbitan, judul artikel dan pengarang serta nomor halaman pada jurnal asli yang dicetak. Dengan kata lain ada kebutuhan untuk menambahkan metadata untuk mendeskripsikan setiap dokumen.

### Klasifikasi Item Perpustakaan

Klasifikasi merupakan salah satu *tools* tertua yang dikembangkan oleh ahli pustakawan dan dianggap sebagai dasar dari kepustakawanan karena fungsinya yang serbaguna. 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). Kelas umum *dewey decimal classification* 23 dapat dilihat pada tabel 2.1.

Tabel 2.1 Kelas Umum *Dewey Decimal Classification* 23

Sumber (Suresha & Narayanaswamy, 2016)

|  |  |  |
| --- | --- | --- |
| No | Kode | Subjek |
| 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* (Satija, 2008). UDC mencakup seluruh domain pengetahuan, dengan 10 kelas utama dan kelas utama ke-4 yang masih kosong, dapat dilihat pada tabel 2.2.

Tabel 2.2 Kelas Utama *Universal Decimal Classification*

Sumber (Satija, 2008)

|  |  |  |
| --- | --- | --- |
| No | Kode | Subjek |
| 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 tabel 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 lokal (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, et al., 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 penggunaannya. 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 (Andini & Zulkarnain, 2013) yaitu dengan cara eksplisit dan implisit. Metode 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*: content-based recommendation, collaborative filtering recommendation, dan hybrid recommendation* (Lü, et al., 2012).

### Content-based Recommendation

*Content-based recommendation* menggunakan ketersediaan konten (sering juga disebut dengan fitur, atribut atau karakteristik) sebuah *item* sebagai basis dalam pemberian rekomendasi (Ricci, et al., 2015). 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.

*Content-based recommendation* mencoba untuk melakukan mencocokkan (*matching*) antara profil *user* (*user* *profile*) dengan konten *item* (*item* *content*). Misalnya, seorang *user* telah memberi *rating* kepada lima 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. Contoh matriks *rating* *content-based recommendation* dapat dilihat pada tabel 2.3.

Tabel 2.3 Contoh Matriks *Rating* *Content-based Recommendation*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| # |  |  |  |  | *Rating* |
|  | 1 | 1 | 1 | 1 | 4 |
|  | 1 | 0 | 1 | 1 | 3 |
|  | 1 | 1 | 0 | 0 | 2 |
|  | 1 | 0 | 1 | 1 | 3 |
|  | 1 | 1 | 0 | 1 | ? |

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

(1)

Di mana:

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 1 atau 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.

(2)

Di mana:

merupakan prediksi *rating* *user* u terhadap *item* i*.*

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

*Content-based recommendation* memiliki beberapa kelebihan (Ricci, et al., 2015), yaitu:

1. *Content-based recommendation* hanya memanfaatkan *rating* yang diberikan oleh *user* aktif.
2. *Content-based recommendation* dapat memberikan rekomendasi dari *user* lain yang menyukai *item* serupa.
3. *Content-based recommendation* dapat memberikan rekomendasi *item* yang belum dinilai *user* mana pun.

Namun, *content-based recommendation* juga memiliki beberapa kelemahan (Ricci, et al., 2015) , yaitu:

1. *Content-based recommendation* memiliki batasan dengan jenis fitur dengan objek yang direkomendasikan.
2. *Content-based recommendation* tidak dapat menemukan sesuatu yang sesuatu yang tidak terduga, sehingga sistem akan merekomendasikan *item* yang serupa dengan yang sudah di *rating* sebelumnya.
3. Sistem rekomendasi harus memiliki *rating* agar dapat memberikan rekomendasi yang akurat, oleh karena itu hanya sedikit *item* yang dapat direkomendasikan sistem.

### Collaborative Filtering Recommendation

Sistem rekomendasi *collaborative filtering* memberikan rekomendasi *item* berdasarkan kumpulan *user* dengan preferensi yang sama atau *similarity* dari setiap *user*. Proses *collaborative* *filtering* dilakukan dengan melihat persamaan *rating* pada matriks *rating* dengan metode *user to user*. *Similarity* antara *user* 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 *heuristic*) membuat prediksi berdasarkan kesamaan antara *user* dan *item*. Pendekatan berbasis model berusaha membuat model prediksi melalui *machine learning*. Secara khusus, model berbasis matriks *factorization* telah memperoleh popularitas karena memiliki akurasi dan skalabilitas yang relatif tinggi. Meski demikian, *collaborative filtering* masih memiliki tantangan terhadap *sparsity* matriks *rating* (Yang, et al., 2017).

*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 *rating* 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 matriks *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 diberi *rating* sebelumnya.

Misalnya, terdapat lima *user* 𝑢1,u2,..𝑢5 dan lima *item* 𝑖1,*i*2,..𝑖5. Sistem rekomendasi akan memprediksi berapa *rating* yang akan diberikan oleh u1 kepada 𝑖5 seperti pada tabel 2.4.

Tabel 2.4 Contoh Matriks *User* x *Item* Sistem Rekomendasi *Collaborative Filtering*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | *Item* | *Item* | *Item* | *Item* | *Item* |
| *User* | 5 | 7 | 5 | 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 *user* yang memiliki ketertarikan yang sama dengan menggunakan persamaan (Rahmawati, et al., 2018):

(3)

Keterangan:

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

adalah interaksi *user* u pada pekerjaan x

adalah rata-rata interaksi dari *user* u

adalah interaksi dari *user* j pada pekerjaan x

adalah rata-rata interaksi dari pengguna j

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

(4)

Keterangan:

U’x adalah prediksi interaksi *user* u pada pekerjaan x

adalah rata-rata interaksi dari *user* u

adalah interaksi dari *user* j pada pekerjaan x

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

Pada praktiknya sistem rekomendasi *collaborative filtering* mengalami berbagai macam masalah. Salah satu masalahnya yaitu *sparsity* 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)*. Sparsity* identik dengan data yang jarang atau banyaknya data yang masih kosong. Hal ini diakibatkan oleh adanya *item* baru yang muncul, *item* yang baru muncul bisa jadi kurang sesuai dengan keinginan *user* sehingga *user* baru membiarkan begitu saja *item* yang ditampilkan dan tidak memberikan *rating-*nya pada suatu *item* karena kurangnya referensi. *Rating* *user* merupakan bagian penting untuk menghasilkan rekomendasi yang andal, *item* yang tidak mendapatkan *rating* akan sulit untuk ditemukan kaitannya dengan *item* yang lain. Banyak lagi faktor yang menyebabkan data *sparsity*, sehingga menjadi tantangan terbesar dalam *collaborative filtering* yang paling sulit untuk diatasi (Li, et al., 2012). Sistem rekomendasi *collaborative filtering* memiliki beberapa kekurangan, diantaranya adalah (Lucas, et al., 2013):

1. Hasil rekomendasi yang tidak andal yang dikarenakan matriks *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 ada pada *content-based* dan *collaborative filtering recommendation*, dan menjadi target penelitian yang sedang marak dikembangkan (Lü, et al., 2012). Terdapat beberapa cara penggabungan yang dapat dilakukan dalam metode *hybrid* (Li & Kim, 2003) 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.*

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*).

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

Penggabungan ini menggabungkan informasi *item* dan *rating* *user* untuk menghitung kemiripan *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-start problem*).

## 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 (Kwanggi, 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 rekomendasi 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)*.*

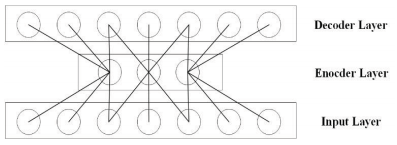
Salah satu teknik yang digunakan pada *deep learning* adalah *auto-encoder. Auto-encoder* merupakan metode *unsupervised learning* yang mencobamerekonstruksi masukan data di lapisankeluaran secara umum, lapisan *bottleneck* (lapisan paling tengah) digunakan sebagairepresentasi fitur menonjol dari datamasukan. Ada banyak varian dari *auto-encoders* seperti *denoising auto-encoder, marginalized denoising auto-encoder, sparse auto-encoder, contractive 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 akan dapat lebih memahami apa yang dibutuhkan *user* dan hal tersebut akan meningkatkan hasil rekomendasi (Ikasari, et al., 2018). Terdapat beberapa teknik *deep learning* yang digunakan pada sistem rekomendasi (Zayyad & Kurniawardhani, 2021) yaitu:

### Multilayer Perceptron

*Multilayer Perceptron (*MLP*)* adalah bentuk khusus dari *Neural Network* (NN), MPL merupakan model komputasi yang sering digunakan untuk memecahkan masalah pembelajaran. Ini terdiri dari grafik terarah berbobot dari satu set neuron yang saling berhubungan yang diatur dalam lapisan terpisah: lapisan *input*, lapisan tersembunyi, dan lapisan *output*. Setiap lapisan memiliki satu atau lebih neuron. Setiap neuron *i* terhubung ke neuron *j* dari lapisan berikutnya dengan satu set link berbobot yang dilambangkan dengan *w1i,...,wji*. Pada lapisan input, {*a1, a2,...,am*} mewakili *m* sinyal *input* yang terkait dengan m atribut. Pada lapisan tersembunyi dan keluaran, setiap neuron j menerima sinyal masukan sebagai kombinasi linier dari keluaran yang diberikan oleh: . Kombinasi linier ditransformasikan menjadi sinyal keluaran menggunakan fungsi aktivasi (*vj*). Sinyal-sinyal ini dikirim dalam arah maju lapis demi lapis ke lapisan keluaran yang memberikan keluaran *yj* untuk setiap neuron keluaran j. Dalam klasifikasi, setiap kelas dikaitkan dengan neuron keluaran dan prediksi biasanya diberikan oleh yang memiliki tingkat aktivasi tertinggi (Ribeiro, et al., 2012).

### Autoencoder Based Recommendation

*Autoencoder* adalah jenis jaringan saraf yang digunakan untuk mengekstraksi fitur dari data, atau merekonstruksi data *input* dari data yang rusak, yang mirip dengan tugas rekomendasi. Jaringan *autoencoder* klasik memiliki lapisan *input* dan *output* yang terhubung penuh dengan ukuran yang sama, dan lapisan tersembunyi yang terhubung penuh dengan ukuran yang lebih kecil. Saat dilatih, model menggunakan data *input* dan data *output* yang sama. Gambar 2.1 adalah struktur *autoencoder* yang menunjukkan bahwa *autoencoder* dapat digunakan dalam sistem rekomendasi. Sebagai contoh, setelah melatih jaringan dengan kode vektor dalam basis data kode, jaringan dapat melakukan pekerjaan rekomendasi dengan cara berikut: Diberikan *input* vektor kode 0-1, *output* jaringan dapat dianggap sebagai *recommendation score vector*. Dengan kata lain, nilai setiap dimensi dari vektor keluaran adalah skor rekomendasi dari API yang bersangkutan. API diberi peringkat berdasarkan skor, dan API dengan peringkat tinggi akan menjadi hasil rekomendasi (Liu, et al., 2019). Berikut struktur dari *autoencoder*:

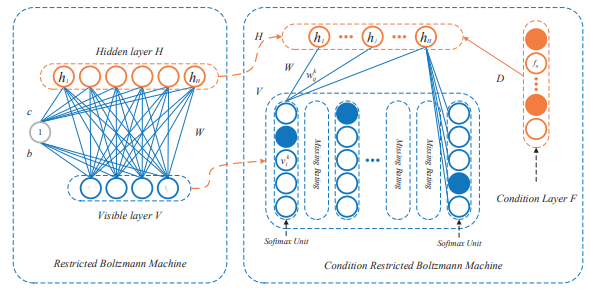


Gambar 2.1 Struktur *Autoencoder*

Sumber (Liu, et al., 2019)

### Restricted Boltzmann Machine based Recommendation

*Restricted Boltzmann Machine* (RBM) adalah model graf generatif tak berarah dua lapis, yang terdiri dari *m* unit tampak *V = (v1, ..., vm )* mewakili data yang dapat diamati dan *n* unit tersembunyi *H = (h1, ..., hn )* menangkap dependensi antara variabel yang dapat diamati yang ditunjukkan seperti pada gambar 2.2. Ketika model RBM dasar digunakan untuk CF, setiap *user* atau *item* dapat dianggap sebagai kasus pelatihan tunggal untuk RBM. Setiap RBM mungkin memiliki jumlah unit *softmax* yang terlihat berbeda tetapi memiliki jumlah unit tersembunyi yang sama, yaitu, semua bobot dan bias yang terhubung dibagi di antara model RBM (Xie, et al., 2016). Berikut model dari RBM:



Gambar 2.2 *Restricted Boltzmann Machine* dengan unit tersembunyi biner dan unit terlihat *softmax*

Sumber (Xie, et al., 2016)

## 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 menjembatanimatriks *factorization* dan *deep feature learning* merupakan *hybrid collaborative filtering model*. *Deep collaborative filtering* menggabungkan *probabilistic matrix factorization* dan *marginalized denoising auto-encoders. Probabilistic matrix factorization* adalah pendekatan *collaborative filtering* yang diterapkan secara luas dengan kinerja yang sangat baik, dan *marginalized denoising auto-encoders* adalah metode yang andal dalam mengekstraksi fitur tingkat tinggi dari *input*. *Marginalized denoising auto-encoders* merupakan varian dari *auto-encoder* yang merupakan metode *unsupervised learning.* Kombinasi keduanya memanfaatkan fungsinya masing-masing untuk mempelajari model yang lebih kompleks (Li, et al., 2015). Notasi yang digunakan pada *deep collaborative filtering* dapat dilihat pada tabel 2.5 dan ilustrasi penerapan *deep collaborative filtering* dapat dilihat pada gambar 2.2.

Tabel 2.5 Ringkasan Notasi

Sumber (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* € | Matriks *Projection* untuk *U* |



Gambar 2.3 Ilustrasi Penerapan *Deep Collaborative Filtering*

Sumber (Li, et al., 2015)

### Matriks Factorization

Matriks *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* *user-item feature*. Matriks *factorization* menguraikan matriks *rating* *R* € yang terdiri dari *rating yang diberikan* oleh *m* *user* untuk *n* *item* menjadi dua *low-rank* *U* € dan *V* € yang terdiri dari fitur *user* dan *item*. Contoh matriks *factorization* dapat dilihat pada gambar 2.1.



Gambar 2.4 Contoh *Matriks Factorization*

(Sumber: <https://developers.google.com/machinelearning/recommendation/collaborative/matrix>)

Salah satu metode matriks *factorization* klasik adalah *probabilistic matrix factorization* (PMF). Asumsi yang mendasari metode ini adalah bahwa distribusi probabilitas sebelumnya dari *latent factor* dan probabilitas *rating* yang diamati mengikuti *Gaussian distribution*. Banyak algoritma yang telah dikembangkan untuk meningkatkan kinerja PMF, salah satunya dengan memasukkan *side information* seperti hubungan sosial. Tetapi metode matriks *factorization* masih 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 *sparsity* data. Memasukkan *side information* telah menunjukkan kinerja yang menjanjikan dalam *collaborative filtering*. Tetapi akan bermasalah jika *side information* tidak lengkap. Oleh karena itu, menggabungkan *deap learning* dengan matriks *factorization* akan memberikan hasil yang lebih baik. Model pada penelitian ini didasarkan pada pendekatan *probabilistic matrix factorization* karena telah terbukti memiliki kinerja yang sangat baik pada beberapa domain.

Sistem mempelajari *latent factor* dengan meminimalkan fungsi objektif berikut:

(5)

Di mana adalah *loss* *function* untuk memprediksi *rating* menggunakan *latent factor* U dan V dan dua istilah terakhir adalah regularisasi yang digunakan untuk menghindari *overfitting*. merupakan *frobenius norm* dan . C merupakan bobot matriks (Li, et al., 2015).

### Marginalized Denoising Auto-encoder

Sebagai bentuk dari jaringan neural, *auto-encoder* mengambil suatu masukan dan memetakannya (menyandikan) ke representasi tersembunyi melalui pemetaan deterministik. *Denoising* *auto-encoders* merekonstruksi masukan dari versi data yang rusak dengan mempelajari pemetaan yang lebih kuat dari data. Berbagai jenis *auto-encoder* telah dikembangkan dalam literatur dan telah menunjukkan hasil yang menjanjikan di beberapa domain. Selain itu, *denoising auto-encoder* dapat ditumpuk untuk membangun *deep network* yang juga dikenal sebagai *stacked denoising autoencoder* (SDA) yang memungkinkan pembelajaran representasi tingkat yang lebih tinggi. Salah satu kelemahan utama SDA adalah biaya pelatihan komputasi yang tinggi, karena SDA mengandalkan teknik pengoptimalan berulang dan numerik untuk mempelajari sejumlah besar parameter model.

*Marginalized denoising auto-encoder* (mDA) adalah varian dari SDA yang menghindari biaya komputasi yang tinggi dengan meminggirkan kerusakan fitur secara acak dan dengan demikian memiliki solusi *closed-form* untuk mempelajari parameter model. Oleh karena itu, mDA sangat skalabel dan lebih cepat dari SDA.

Diberikan sampel set mDA mempertimbangkan beberapa lintasan (misalnya, c-*times*) dari *corruptions* acak untuk mendapatkan . Kemudian merekonstruksi *input* dengan pemetaan W yang meminimalkan *squared loss* sebagai berikut:

, (6)

Di mana merepresentasikan *corrupted version* ke- dari *input original* dan W merepresentasikan pemetaan yang diharapkan dapat meminimalkan *loss function*.

Objektif di atas dapat ditulis ulang dalam bentuk matriks sebagai

, (7)

Di mana merupakan *c-times repeated version* dari X, dan merupakan *corrupted version*. , di mana dan (Li, et al., 2015).

### Normalisasi Data Pada Deep Collaborative Filtering

Normalisasi data dilakukan pada matriks *rating* yang mengindikasikan *rating* oleh *user* terhadap *item*. Normalisasi dilakukan dengan menggunakan *zero-mean normalization*. Nilai *rating* dari dataset dalam skala 1 – 10 akan dinormalisasikan menjadi *zero-mean* (Li, et al., 2015). Berikut rumus untuk metode *zero-mean normalization* (Henderi, et al., 2021).

(8)

Di mana:

adalah nilai hasil normalisasi

adalah nilai *rating* yang lama

adalah rata-rata

adalah nilai *standard deviation*

U1-U10 adalah *user* dan J1-J10 adalah *item* yang di *rating* ataupun yang tidak di *rating* oleh *user*. Data yang belum dinormalisasi dapat dilihat pada tabel 2.6 dan data yang sudah dinormalisasi dapat dilihat pada tabel 2.7.

Tabel 2.6 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 |

Tabel 2.7 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 |

### Menghitung Kesalahan Model (*Loss Function*)

Inti dari *machine learning* adalah model yang dilatih dengan data pelatihan, dan metode untuk menyesuaikan parameter model adalah berdasarkan *loss function*. Tujuan dari pelatihan ini adalah untuk meminimalkan *loss* akibat salah prediksi rata-rata model. *Loss function* diterapkan untuk menilai kualitas dan kapabilitas model pada pelatihan model *machine learning* (Chen, et al., 2018). *Deep collaborative filtering* memanfaatkan *loss function* dari *probabilistic matrix factorization* untuk menguraikan matriks *rating* R yaitu, . Berikut fungsi objektif yang digunakan pada *deep collaborative filtering*:

(9)

Di mana:

A adalah matriks indikator yang menunjukkan entri yang tidak kosong di matriks *rating* dan melambangkan produk *hadamard* atau *pointwise product*. adalah pemetaan rekonstruktif, adalah matriks *Projection*, adalah parameter *tradeoff*. Istilah pertama menunjukkan proses pembelajaran *marginalized denoising auto encoders.* Ini mengukur kesalahan rekonstruksi antara input fitur *user* dan fitur yang dipetakan dari *input* yang rusak yakni, . adalah *learned mapping* yang diharapkan untuk meminimalkan *loss data.* Istilah kedua berfungsi untuk menghubungkan *hidden layer feature* dan *latent factor* U (Li, et al., 2015).

### Tahapan Rekomendasi Pada Deep Collaborative Filtering

* + - 1. Mencari *latent factor user* dan *item*

*Latent factor* *model* telah menjadi teknik yang sukses digunakan untuk membangun sistem rekomendasi. Sebagian besar penelitian difokuskan untuk mempelajari l*atent factor* dalam situasi data *sparsity* (Rao, et al., 2017). Model *latent factor* mencoba menemukan perkiraan *rating* di matriks *user-item*, di mana *rating* digunakan untuk mengisi entri yang hilang (Sammut & Webb, 2011). Pada proses untuk mendapatkan *latent factor* akan terdapat *noisy data* yang tak terhingga jumlahnya sehingga perlu dilakukan *mapping function* untuk *side information user* dan *item* (Li, et al., 2015). Dilakukan *mapping function* W1 untuk *side information* *user* dan W2 untuk *side information item*.

(10)

(11)

,

.

Pada dan Misalkan = Z, dan p = *corrupt\_ratio*, untuk mendapatkan nilai Z setiap nilai yang bukan di diagonal akan dikalikan dengan (1 - p)2 dan yang berada di diagonal akan dikalikan dengan (1 - p) (Chen, et al., 2012).

(12)

Demikian pula dengan dan , setiap elemen pada akan dikalikan dengan (1 - p).

(13)

Lalu diperlukan suatu matriks *projection* untuk memetakan *latent factor user* dan *item* ke *feature space.* *Feature space* merupakan kumpulan fitur yang digunakan untuk mengkarakterisasi data yang akan dilatih (Mohamed, 2017). Berikut rumus matriks *projection*:

(14)

(15)

Di mana untuk matriks *projection* *latent factor user* dan untuk matriks *projection* *latent factor item*.

Untuk menyelesaikan permasalahan *latent factor* U dan V, digunakan algoritma *stochastic gradient descent (SGD).* SGD adalah metode yang befungsi untuk mengoptimalkan fungsi *objective* *machine-learning* (Li & Orabona, 2019). Berikut rumus untuk mencari SGD:

(16)

(17)

Di mana η adalah *learning rate*, dan turunan dari rumus diatas didefinisikan sebagai berikut:

(18)

(19)

**Algoritma DCF untuk menemukan *latent factor* *user* dan *item***

**Input:** Matriks *rating* *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* (10);

4: *Update* *using* (11);

5: *Update* *using* (14);

6: *Update* *using* (15);

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

8: *Update* *using* (16);

9: *Update* *using* (17);

10: ***end for***

11: ***end while***

* + - 1. Menghasilkan rekomendasi *set*

Setelah *latent factor user* dan *item* ditemukan, dilakukan perkalian matriks untuk kedua *latent factor* seperti pada gambar 2.3, sehingga didapatkan *user-item* matriks *rating* yang baru. Matriks *rating* yang baru dapat dilihat pada tabel 2.8.



Gambar 2.5 Perkalian Matriks *Latent Factor User* dan *Item*

Tabel 2.8 Matriks *Rating* Hasil Prediksi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Item1 | Item2 | Item3 | Item4 |
| User1 | 2,36704 | 3,55596 | 1,68560 | 2,30801 |
| User2 | 3,61134 | 4,85569 | 3,06585 | 3,81974 |
| User3 | 3,13898 | 3,82720 | 2,66200 | 3,31269 |
| User4 | 2,88840 | 3,23815 | 2,30140 | 3,13117 |

Matriks *rating* yang didapatkan akan dilakukan normalisasi kembali menggunakan metode *min-max normalization* sehingga nilai *rating* yang didapat berada di dalam skala 1-10. Berikut rumus metode *min-max normalization* (Gajera, et al., 2016).

(20)

Di mana:

adalah nilai hasil normalisasi

adalah nilai *rating* yang lama

adalah nilai minimal pada matriks *rating*

adalah nilai maksimal pada matriks *rating*

adalah nilai minimal yang baru

adalah nilai maksimal yang baru

Normalisasi matriks *rating* menggunakan *min-max normalization* dapat dilihat pada tabel 2.9.

Tabel 2.9 Matriks *Rating* Prediksi Setelah Normalisasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Item1 | Item2 | Item3 | Item4 |
| User1 | 2,934633 | 6,31002 | 1 | 2,767044 |
| User2 | 6,467245 | 10 | 4,91858 | 7,058901 |
| User3 | 5,126198 | 7,08008 | 3,772035 | 5,619367 |
| User4 | 4,414793 | 5,407746 | 2,748278 | 5,104025 |

Jadi, setiap *user* akan mendapat rekomendasi *item* berdasarkan prediksi *rating user* terhadap *item* di mulai dari *item* dengan *rating* tertinggi sampai yang terendah.

## 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 & Rojas, 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 & Dondeti, 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 & Dondeti, 2012).

### 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 *rating* 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). MAE dan RMSE disebut sebagai 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 *rating* yang diprediksi dan *rating* sebenarnya. Persamaan MAE yaitu sebagai berikut (Nilashi, et al., 2013):

(21)

Di mana*,*

*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)

RMSE adalah metrik lain dalam akurasi prediksi, yaitu menghitung statistik akurasi. Secara konsep ini mirip dengan MAE, perbedaannya RMSE mendapatkan hasil dengan mengkuadratkan *error* (Nilashi, et al., 2013). Berikut rumus persamaan RMSE:

(22)

Di mana,

𝑓𝑡 = Permintaan aktual periode *t*

= Ramalan permintaan periode *t*

m = Jumlah periode peramalan

# BAB III ANALISIS DAN PERANCANGAN

## Analisis

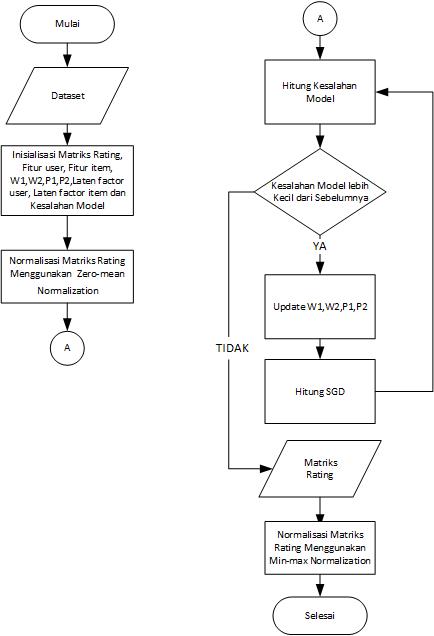
Tahapan analisis terdiri dari analisis proses dan analisis kebutuhan. Analisis proses menjelaskan cara kerja *deep collaborative filtering* dan alur sistem rekomendasi*.* Analisis kebutuhan terdiri dari analisis kebutuhan fungsional dan *non*-fungsional.

### Analisis Proses

Analisis proses digunakan untuk menjelaskan proses kerja pada perangkat lunak untuk menyelesaikan permasalahan yang ada, yaitu rekomendasi *item* dengan *deep collaborative filtering*.Proses ini dimulai dari membaca *dataset* dari *database* kemudian proses rekomendasi *item* digital dengan *deep collaborative filtering*.

#### Deep Collaborative Filtering

Analisis proses cara kerja *deep collaborative filtering* dalam fitur rekomendasi *item* digital untuk memberikan rekomendasi *item* digital yang sesuai berdasarkan *rating* yang sudah diberikan oleh *user* dan juga berdasarkan profil *user* dan *item*. Penjelasan lebih lanjut mengenai alur proses dalam aplikasi sistem rekomendasi ini dijelaskan dengan *flowchart* berikut:



Gambar 3.1 *Flowchart* dari *Deep Collaborative Filtering*

Tahapan pertama yang dilakukan adalah inisialisasi matriks *rating R* € , fitur *user*  *X* € , fitur *item* *Y* € , *W1* €  *, W2* € *, P1* € *, P1* € , *latent factor* *user* *U* € dan *item* *V* € *, A* € ,parameter λ=0,02, α= 0,02, β= 0,02, d=5, *corrupt\_ratio*=0,004, *learning\_rate*=0,002. Kesalahan model = 9223372036854775807. Berikut inisialisasi matriks *rating*, fitur *user,* fitur *item*, *latent factor user, latent factor item,*

Tabel 3.1 Contoh Inisialisasi Matriks *Rating* (R)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **item1** | **item2** | **item3** | **item4** | **item5** |
| **user1** | 2,0 | 4,0 | 1,0 | 0,0 | 0 |
| **user2** | 0,0 | 0 | 0,0 | 1,0 | 2,0 |
| **user3** | 5,0 | 1,0 | 1,0 | 0,0 | 0 |
| **user4** | 0 | 0,0 | 3,0 | 7,0 | 0,0 |
| **user5** | 0,0 | 0 | 0,0 | 4,0 | 0,0 |

Tabel 3.2 Contoh Inisialisasi Fitur *User* (X)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **user1** | **user2** | **user3** | **user4** | **user5** |
| **indonesia** | 1 | 0 | 0 | 0 | 1 |
| **usa** | 0 | 1 | 1 | 0 | 0 |
| **russia** | 0 | 0 | 0 | 1 | 0 |
| **age5-50** | 0 | 1 | 1 | 0 | 1 |
| **age51-100** | 1 | 0 | 0 | 1 | 0 |

Tabel 3.3 Contoh Inisialisasi Fitur *Item* (Y)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **#** | **item1** | **item2** | **item3** | **item4** | **item5** |
| **fiction** | 1 | 0 | 0 | 0 | 0 |
| **humor** | 0 | 0 | 1 | 0 | 0 |
| **biography** | 0 | 1 | 0 | 0 | 0 |
| **fantasy** | 0 | 0 | 0 | 0 | 1 |
| **science** | 0 | 0 | 0 | 1 | 0 |

Tabel 3.4 Contoh Inisialisasi *Mapping Funtion User* (W1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,54340 | 0,27837 | 0,42452 | 0,84478 | 0,00472 |
| 0,12157 | 0,67075 | 0,82585 | 0,13671 | 0,57509 |
| 0,89132 | 0,20920 | 0,18533 | 0,10838 | 0,21970 |
| 0,97862 | 0,81168 | 0,17194 | 0,81622 | 0,27407 |
| 0,43170 | 0,94003 | 0,81765 | 0,33611 | 0,17541 |

Tabel 3.5 Contoh Inisialisasi *Mapping Funtion* *Item* (W2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,97501 | 0,88485 | 0,35951 | 0,59886 | 0,35480 |
| 0,34019 | 0,17808 | 0,23769 | 0,04486 | 0,50543 |
| 0,37625 | 0,59281 | 0,62994 | 0,14260 | 0,93384 |
| 0,94638 | 0,60230 | 0,38777 | 0,36319 | 0,20435 |
| 0,27677 | 0,24654 | 0,17361 | 0,96661 | 0,95701 |

Tabel 3.6 Contoh Inisialisasi Matriks *Projection* *User* (P1)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,37283 | 0,00569 | 0,25243 | 0,79566 | 0,01525 |
| 0,59884 | 0,60380 | 0,10515 | 0,38194 | 0,03648 |
| 0,89041 | 0,98092 | 0,05994 | 0,89055 | 0,57690 |
| 0,74248 | 0,63018 | 0,58184 | 0,02044 | 0,21003 |
| 0,54468 | 0,76912 | 0,25070 | 0,28590 | 0,85240 |

Tabel 3.7 Contoh Inisialisasi Matriks *Projection Item* (P2)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,59797 | 0,73130 | 0,34039 | 0,09206 | 0,46350 |
| 0,50870 | 0,08846 | 0,52804 | 0,99216 | 0,39504 |
| 0,33560 | 0,80545 | 0,75435 | 0,31307 | 0,63404 |
| 0,54040 | 0,29679 | 0,11079 | 0,31264 | 0,45698 |
| 0,65894 | 0,25426 | 0,64110 | 0,20012 | 0,65762 |

Tabel 3.8 *Latent Factor User* (U)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,30863 | 0,30719 | -0,70064 | 0,40472 | 0,34830 |
| 0,19151 | 0,04869 | 0,01265 | -0,25886 | -0,53641 |
| -0,76292 | 0,16510 | 0,83923 | -0,16856 | 0,81927 |
| 0,00135 | -0,03400 | 0,00177 | -0,08274 | -1,11229 |
| -0,76234 | -0,50815 | -1,32971 | 0,01490 | -0,11131 |

Tabel 3.9 *Latent Factor Item* (V)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,20133 | 0,05922 | 0,00993 | 0,14193 | -0,33649 |
| -0,57976 | 0,04255 | -0,18949 | -0,53039 | -0,16344 |
| -0,56842 | 0,70936 | 0,31009 | -0,87568 | -0,06028 |
| -0,68898 | 0,91532 | -0,62476 | -0,49067 | -0,10676 |
| -0,63910 | 0,42441 | -0,00867 | 0,40008 | 0,33975 |

Tabel 3.10 Contoh Inisialisasi Matriks *Rating Non-empty Entries* (A)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 0 | 1 | 0 |

Tahapan selanjutnya menghitung nilai normalisasi matriks *rating* menggunakan metode *zero-mean normalization* terlihat pada tabel 3.11.

Tabel 3.11 Matriks *Rating* Setelah Normalisasi

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,40380 | 1,46643 | -0,12752 | -0,65883 | -0,65883 |
| -0,65883 | -0,65883 | -0,65883 | -0,12752 | 0,40380 |
| 1,99774 | -0,12752 | -0,12752 | -0,65883 | -0,65883 |
| -0,65883 | -0,65883 | 0,93511 | 3,06037 | -0,65883 |
| -0,65883 | -0,65883 | -0,65883 | 1,46643 | -0,65883 |

Tahapan selanjutnya dilakukan perulangan. Di setiap perulangan dilakukan perhitungan untuk kesalahan model, melakukan *update* terhadap W1, W2, P1, P2, U, V dan dilakukan perhitungan *Stochastic Gradient Descent* (SGD) untuk mendapat *latent factor user* dan *item* yang baru. Perulangan akan berhenti apabila kesalahan model yang didapat lebih besar dari pada sebelumnya.

1. Perulangan 1
2. Menghitung kesalahan model

Untuk menghitung kesalahan model terdapat 4 bagian perhitungan yaitu :

Pertama dilakukan perhitungan . Dilakukan perkalian matriks dan

Tabel 3.12 Hasil Perkalian Matriks dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,26729 | -0,13928 | -0,19328 | -0,08205 | -0,61261 |
| 0,46392 | 0,02698 | -0,30344 | -0,09171 | -0,90153 |
| 1,09549 | -0,32094 | -0,14454 | -0,74741 | -1,30790 |
| 0,09650 | 0,06228 | 0,19451 | -0,25469 | -1,68301 |
| 0,64132 | -0,38631 | 0,57196 | -0,99673 | -1,23003 |

Setelah itu, dilakukan perkalian matriks W1 dengan matriks X

Tabel 3.13 Hasil Perkalian Matriks W1 dan X

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,54812 | 1,12315 | 1,12315 | 0,42924 | 1,38818 |
| 0,69666 | 0,80746 | 0,80746 | 1,40095 | 0,25828 |
| 1,11102 | 0,31758 | 0,31758 | 0,40503 | 0,99970 |
| 1,25270 | 1,62791 | 1,62791 | 0,44601 | 1,79485 |
| 0,60711 | 1,27614 | 1,27614 | 0,99306 | 0,76782 |

Setelah itu, dilakukan pengurangan

Tabel 3.14 Hasil Pengurangan dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,28084 | -1,26242 | -1,31642 | -0,51128 | -2,00079 |
| -0,23275 | -0,78048 | -1,11090 | -1,49265 | -1,15981 |
| -0,01553 | -0,63852 | -0,46212 | -1,15243 | -2,30760 |
| -1,15620 | -1,56562 | -1,43340 | -0,70071 | -3,47785 |
| 0,03420 | -1,66245 | -0,70418 | -1,98979 | -1,99785 |

Untuk mendapatkan *frobenius* *norm* dari , dilakukan pengkuadratan setiap elemen matriks lalu dihitung jumlah dari semua elemennya. Berikut hasil pengkuadratan matriks .

Tabel 3.15 Hasil Pengkuadratan Matriks .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,07887 | 1,59371 | 1,73297 | 0,26141 | 4,00317 |
| 0,05417 | 0,60914 | 1,23409 | 2,22802 | 1,34516 |
| 0,00024 | 0,40771 | 0,21355 | 1,32810 | 5,32503 |
| 1,33680 | 2,45118 | 2,05464 | 0,49099 | 12,09547 |
| 0,00117 | 2,76374 | 0,49587 | 3,95928 | 3,99140 |

Semua elemen hasil pengkuadratan matriks dijumlahkan sehingga didapatkan *frobenius* *norm* dari .

50,0558780

= 0,02 \* 50,0558780 = 1,00111756

= 1,00111756

Kedua, dilakukan perhitungan = . Dilakukan perkalian matriks dan

Tabel 3.16 Hasil Perkalian Matriks dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,21659 | -0,50465 | 0,17585 | -0,04993 | 0,11955 |
| -0,08404 | -0,98201 | -0,95529 | -1,12840 | 0,23900 |
| -0,18128 | -0,57291 | 0,30214 | -0,18656 | 0,46148 |
| -0,19951 | -0,56218 | -0,36361 | -0,37207 | 0,05997 |
| -0,30411 | -0,70632 | -0,21028 | -0,79021 | -0,01529 |

Setelah itu, perkalian matriks W2 dengan matriks Y

Tabel 3.17 Hasil Perkalian Matriks W2 dan Y

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,97501 | 0,35951 | 0,88485 | 0,35480 | 0,59886 |
| 0,34019 | 0,23769 | 0,17808 | 0,50543 | 0,04486 |
| 0,37625 | 0,62994 | 0,59281 | 0,93384 | 0,14260 |
| 0,94638 | 0,38777 | 0,60230 | 0,20435 | 0,36319 |
| 0,27677 | 0,17361 | 0,24654 | 0,95701 | 0,96661 |

Dilakukan pengurangan

Tabel 3.18 Hasil Pengurangan dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -1,19160 | -0,86416 | -0,70900 | -0,40473 | -0,47931 |
| -0,42423 | -1,21971 | -1,13337 | -1,63384 | 0,19414 |
| -0,55753 | -1,20285 | -0,29066 | -1,12040 | 0,31888 |
| -1,14589 | -0,94995 | -0,96591 | -0,57642 | -0,30322 |
| -0,58088 | -0,87993 | -0,45682 | -1,74722 | -0,98190 |

Untuk mendapatkan *frobenius* *norm* dari , dilakukan pengkuadratan setiap elemen matriks lalu dihitung jumlah dari semua elemennya. Berikut hasil pengkuadratan matriks .

Tabel 3.19 Hasil Pengkuadratan Matriks .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1,41991 | 0,74676 | 0,50268 | 0,16380 | 0,22973 |
| 0,17997 | 1,48769 | 1,28453 | 2,66942 | 0,03769 |
| 0,31084 | 1,44686 | 0,08448 | 1,25530 | 0,10168 |
| 1,31307 | 0,90240 | 0,93298 | 0,33226 | 0,09194 |
| 0,33742 | 0,77427 | 0,20868 | 3,05277 | 0,96413 |

Semua elemen hasil pengkuadratan matriks dijumlahkan sehingga didapatkan *frobenius* *norm* dari .

= 20,8312940

0,02 \* 20,8312940 = 0,41662588.

0,41662588.

Ketiga, dilakukan perhitungan . Dilakukan perkalian matriks U dan .

Tabel 3.20 Hasil Perkalian Matriks U dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,11066 | -0,30468 | -0,55019 | 0,27050 | 0,21946 |
| 0,10821 | 0,11361 | 0,18862 | 0,08899 | -0,38765 |
| -0,12789 | 0,24581 | 0,90922 | 0,14769 | 0,76128 |
| 0,36026 | 0,22311 | 0,11517 | 0,12619 | -0,42630 |
| 0,14974 | 0,68262 | -0,34580 | 0,89544 | 0,25123 |

Lalu dilakukan pengurangan matriks *rating* R dengan .

Tabel 3.21 Hasil Pengurangan Matriks *rating* R dengan .

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,51446 | 1,77111 | 0,42268 | -0,92933 | -0,87828 |
| -0,76704 | -0,77244 | -0,84745 | -0,21651 | 0,79145 |
| 2,12563 | -0,37333 | -1,03674 | -0,80652 | -1,42011 |
| -1,01909 | -0,88194 | 0,81994 | 2,93418 | -0,23253 |
| -0,80857 | -1,34144 | -0,31303 | 0,57099 | -0,91006 |

Setelah itu dilakukan perhitungan

Tabel 3.22 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,514459111 | 1,77110702 | 0,422676857 | 0 | 0 |
| 0 | 0 | 0 | -0,2165096 | 0,791445207 |
| 2,125628124 | -0,3733293 | -1,036739326 | 0 | 0 |
| 0 | 0 | 0,819939172 | 2,93418189 | 0 |
| 0 | 0 | 0 | 0,5709885 | 0 |

19,5936555

0,02 \* 19,5936555 = 0,39187311

Keempat, dilakukan perhitungan .

= 7,2381135

5,3713859

= 0,02 \* (7,2381135+ 5,3713859) = 0,25218999

Untuk mendapatkan kesalahan model, dilakukan penjumlahan untuk keempat nilai, sehingga nilai *loss* menjadi :

*Loss* = 1,00111756 + 0,41662588 + 0,39187311 + 0,25218999

*Loss* = 2,061807

1. *Update* W1

Untuk melakukan update W1 dilakukan perhitungan untuk mencari . Pertama lakukan perhitungan untuk = . Untuk mendapatkan nilai , semua nilai hasil perkalian matriks *X* dengan dikalikan dengan (1 - *corrupt\_ratio*). Berikut hasil perkalian matriks .

Tabel 3.23 Hasil Perkalian Matriks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1,992 | 0 | 0 | 0,996 | 0,996 |
| 0 | 1,992 | 0 | 1,992 | 0 |
| 0 | 0 | 0,996 | 0 | 0,996 |
| 0,996 | 1,992 | 0 | 2,988 | 0 |
| 0,996 | 0 | 0,996 | 0 | 1,992 |

Selanjutnya dilakukan perhitungan .

Tabel 3.24 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,00691 | -0,00665 | -0,00164 | -0,01890 | 0,00370 |
| -0,00875 | -0,00553 | -0,00183 | -0,02356 | 0,00744 |
| -0,00425 | -0,00931 | -0,01495 | -0,03547 | 0,00696 |
| -0,03173 | 0,00514 | -0,00509 | -0,02852 | -0,00316 |
| -0,01177 | 0,00371 | -0,01993 | -0,02089 | -0,00711 |

Sehingga didapatkan nilai sebagai berikut:

Tabel 3.25 Nilai Hasil Penjumlahan dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1,98509 | -0,00665 | -0,00164 | 0,97710 | 0,99970 |
| -0,00875 | 1,98647 | -0,00183 | 1,96844 | 0,00744 |
| -0,00425 | -0,00931 | 0,98105 | -0,03547 | 1,00296 |
| 0,96427 | 1,99714 | -0,00509 | 2,95948 | -0,00316 |
| 0,98423 | 0,00371 | 0,97607 | -0,02089 | 1,98489 |

Setelah itu lakukan perhitungan untuk mendapatkan nilai . Untuk mendapatkan nilai nilai hasil perkalian matriks *X* dengan untuk setiap nilai yang bukan di diagonal akan dikalikan dengan (1 – *corrupt\_ratio*)2  dan untuk semua nilai di diagonal, dikalikan dengan (1 - *corrupt\_ratio*).

Tabel 3.26 Hasil Perkalian Matriks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1,992 | 0 | 0 | 0,992016 | 0,992016 |
| 0 | 1,992 | 0 | 1,984032 | 0 |
| 0 | 0 | 0,996 | 0 | 0,992016 |
| 0,992016 | 1,984032 | 0 | 2,988 | 0 |
| 0,992016 | 0 | 0,992016 | 0 | 1,992 |

Selanjutnya dilakukan perhitungan untuk

Tabel 3.27 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,04 | 0 | 0 | 0,02 | 0,02 |
| 0 | 0,04 | 0 | 0,04 | 0 |
| 0 | 0 | 0,02 | 0 | 0,02 |
| 0,02 | 0,04 | 0 | 0,06 | 0 |
| 0,02 | 0 | 0,02 | 0 | 0,04 |

Sehingga didapatkan nilai sebagai berikut:

Tabel 3.28 Nilai Hasil Penjumlahan dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2,032 | 0 | 0 | 1,012016 | 1,012016 |
| 0 | 2,032 | 0 | 2,024032 | 0 |
| 0 | 0 | 1,016 | 0 | 1,012016 |
| 1,012016 | 2,024032 | 0 | 3,048 | 0 |
| 1,012016 | 0 | 1,012016 | 0 | 2,032 |

Juga, didapatkan hasil *update* W1 sebagai berikut:

Tabel 3.29 Hasil *Update* W1 Perulangan 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,78087 | -0,19002 | -0,20695 | 0,18748 | 0,20615 |
| -0,20044 | 0,79167 | -0,20816 | 0,18666 | 0,20716 |
| -0,10224 | -0,07917 | 0,83989 | 0,07489 | 0,12620 |
| 0,27781 | 0,31774 | 0,26663 | 0,66772 | -0,27271 |
| 0,19070 | 0,21185 | 0,16334 | -0,21085 | 0,80049 |

1. *Update* W2

Untuk melakukan update W2 dilakukan perhitungan untuk mencari . Pertama lakukan perhitungan untuk . Untuk mendapatkan nilai semua nilai hasil perkalian matriks *Y* dengan dikalikan dengan (1 - *corrupt\_ratio*).

Tabel 3.30 Hasil Perkalian Matriks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,996 | 0 | 0 | 0 | 0 |
| 0 | 0,996 | 0 | 0 | 0 |
| 0 | 0 | 0,996 | 0 | 0 |
| 0 | 0 | 0 | 0,996 | 0 |
| 0 | 0 | 0 | 0 | 0,996 |

Selanjutnya dilakukan perhitungan .

Tabel 3.31 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,00433 | 0,00352 | -0,01009 | 0,00239 | -0,00100 |
| -0,00168 | -0,01911 | -0,01964 | 0,00478 | -0,02257 |
| -0,00363 | 0,00604 | -0,01146 | 0,00923 | -0,00373 |
| -0,00399 | -0,00727 | -0,01124 | 0,00120 | -0,00744 |
| -0,00608 | -0,00421 | -0,01413 | -0,00031 | -0,01580 |

Setelah itu lakukan perhitungan untuk mendapatkan nilai . Untuk mendapatkan nilai nilai hasil perkalian matriks *Y* dengan untuk setiap nilai yang bukan di diagonal akan dikalikan dengan (1 – *corrupt\_ratio*)2  dan untuk semua nilai yang di diagonal dikalikan dengan (1 - *corrupt\_ratio*).

Tabel 3.32 Hasil Perkalian Matriks

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,996 | 0 | 0 | 0 | 0 |
| 0 | 0,996 | 0 | 0 | 0 |
| 0 | 0 | 0,996 | 0 | 0 |
| 0 | 0 | 0 | 0,996 | 0 |
| 0 | 0 | 0 | 0 | 0,996 |

Selanjutnya dilakukan perhitungan untuk

Tabel 3.33 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,02 | 0 | 0 | 0 | 0 |
| 0 | 0,02 | 0 | 0 | 0 |
| 0 | 0 | 0,02 | 0 | 0 |
| 0 | 0 | 0 | 0,02 | 0 |
| 0 | 0 | 0 | 0 | 0,02 |

Sehingga didapatkan nilai sebagai berikut:

Tabel 3.34 Nilai Hasil Penjumlahan dan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 1,016 | 0 | 0 | 0 | 0 |
| 0 | 1,016 | 0 | 0 | 0 |
| 0 | 0 | 1,016 | 0 | 0 |
| 0 | 0 | 0 | 1,016 | 0 |
| 0 | 0 | 0 | 0 | 1,016 |

Juga, didapatkan hasil *update* W1 sebagai berikut:

Tabel 3.35 Hasil *Update* W2 Perulangan 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2,00767 | 0,00352 | -0,01009 | 0,00239 | -0,00100 |
| -0,00168 | 1,99289 | -0,01964 | 0,00478 | -0,02257 |
| -0,00363 | 0,00604 | 2,00054 | 0,00923 | -0,00373 |
| -0,00399 | -0,00727 | -0,01124 | 2,01320 | -0,00744 |
| -0,00608 | -0,00421 | -0,01413 | -0,00031 | 1,99620 |

1. *Update* P1

Untuk melakukan update P1 dilakukan dengan rumus .

Tabel 3.36 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -1,53030 | -0,31151 | -1,27237 | -0,27305 | -0,12335 |
| -0,44138 | 0,20775 | -0,14121 | -0,17523 | -1,11596 |
| -0,60014 | -0,11259 | -1,83648 | 0,29529 | -0,08498 |
| -1,91127 | -0,19439 | -1,87675 | -0,19897 | 0,20090 |
| -1,12583 | 0,03539 | -0,35747 | -0,37046 | -0,61761 |

Tabel 3.37 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2,18065 | -2,00783 | -0,28821 | -2,40774 | 1,08209 |
| -2,00783 | 10,70983 | -2,86138 | -6,05008 | -0,57163 |
| -0,28821 | -2,86138 | 1,70846 | 4,41605 | -0,55234 |
| -2,40774 | -6,05008 | 4,41605 | 16,94219 | -2,54094 |
| 1,08209 | -0,57163 | -0,55234 | -2,54094 | 1,14606 |

Sehingga didapatkan hasil *update*  P1 hasil perkalian dan sebagai berikut:

Tabel 3.38 Hasil *Update* P1 Perulangan 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -1,82094 | 5,09954 | -1,97905 | -4,36218 | -0,22264 |
| -2,12459 | 5,21338 | -0,86596 | -0,95105 | -1,35208 |
| -1,35631 | 3,51612 | -1,29147 | -0,76501 | -0,41841 |
| -2,54019 | 8,21469 | -3,08890 | -6,39130 | -0,18464 |
| -2,19941 | 6,25666 | -1,68234 | -3,78906 | -0,80754 |

1. *Update* P2

Untuk melakukan update P1 dilakukan dengan rumus .

Tabel 3.39 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -1,53487 | 1,27963 | -0,01091 | -0,76164 | -0,27460 |
| -0,68443 | 0,63825 | -0,30260 | -0,46378 | -0,20277 |
| -1,51247 | 1,38488 | -0,51647 | -1,20098 | -0,31654 |
| -1,13060 | 0,84097 | -0,00812 | -0,55373 | -0,31656 |
| -1,57363 | 1,48487 | -0,55998 | -0,35154 | 0,08987 |

Tabel 3.40 Hasil Perhitungan

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2,54155 | 2,06762 | -0,30031 | -0,15632 | -0,08017 |
| 2,06762 | 3,20656 | 0,70932 | 0,97734 | -1,33476 |
| -0,30031 | 0,70932 | 2,63140 | 0,61073 | -1,21994 |
| -0,15632 | 0,97734 | 0,61073 | 1,66282 | -1,77169 |
| -0,08017 | -1,33476 | -1,21994 | -1,77169 | 5,76238 |

Sehingga didapatkan hasil *update* P2 hasil perkalian dan sebagai berikut:

Tabel 3.41 Hasil *Update* P2 Perulangan 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -1,11081 | 0,54408 | 1,20973 | 0,70393 | -1,80458 |
| -0,24021 | 0,23420 | -0,17388 | 0,13404 | -0,77465 |
| -0,61239 | 0,19590 | -0,26981 | -0,16168 | -0,79345 |
| -1,02030 | 0,23454 | 0,96267 | 0,63378 | -1,86502 |
| -0,71340 | 0,64690 | -0,27206 | 0,61143 | -0,03194 |

1. Menghitung *Stochastic Gradient Descent* (SGD)

Menghitung SGD untuk mendapatkan *latent factor user* dan *item* yang baru.

Tabel 3.42 *Latent Factor User* Perulangan 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,30866 | 0,30722 | -0,70055 | 0,40468 | 0,34828 |
| 0,19182 | 0,04890 | 0,01283 | -0,25861 | -0,53618 |
| -0,76252 | 0,16520 | 0,83926 | -0,16820 | 0,81928 |
| 0,00152 | -0,03351 | 0,00177 | -0,08262 | -1,11201 |
| -0,76160 | -0,50772 | -1,32924 | 0,01535 | -0,11097 |

Tabel 3.43 *Latent Factor Item* Perulangan 1

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,20123 | 0,05938 | 0,01016 | 0,14199 | -0,33622 |
| -0,57942 | 0,04273 | -0,18940 | -0,53013 | -0,16325 |
| -0,56812 | 0,70937 | 0,31009 | -0,87546 | -0,06027 |
| -0,68859 | 0,91547 | -0,62426 | -0,49043 | -0,10665 |
| -0,63897 | 0,42441 | -0,00864 | 0,40004 | 0,33977 |

Perhitungan untuk perulangan 2 dan seterusnya sama seperti perulangan 1.

1. Perulangan 2
2. Menghitung kesalahan model

Nilai kesalahan model yang didapat setelah dilakukan perhitungan adalah:

0,314579721

0,31752178

0,391803354

= 0,25204013

*Loss* = 0,314579721+ 0,31752178+ 0,391803354+ 0,25204013

*Loss* = 1,275945

1. *Update* W1

Tabel 3.44 Hasil *Update* W1 Perulangan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,79610 | -0,18857 | -0,18737 | 0,21148 | 0,19662 |
| -0,19434 | 0,79372 | -0,17737 | 0,20328 | 0,20579 |
| -0,08393 | -0,09509 | 0,88024 | 0,10173 | 0,10850 |
| 0,31153 | 0,31285 | 0,29112 | 0,70223 | -0,28340 |
| 0,20153 | 0,20698 | 0,21372 | -0,18260 | 0,78927 |

1. *Update* W2

Tabel 3.45 Hasil *Update* W2 Perulangan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2,03150 | 0,01769 | 0,00718 | 0,01197 | 0,00710 |
| 0,00680 | 2,01556 | 0,00475 | 0,00090 | 0,01010 |
| 0,00752 | 0,01185 | 2,02459 | 0,00285 | 0,01867 |
| 0,01892 | 0,01204 | 0,00775 | 2,01926 | 0,00409 |
| 0,00554 | 0,00493 | 0,00347 | 0,01933 | 2,03114 |

1. *Update* P1

Tabel 3.46 Hasil *Update* P1 Perulangan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,73439 | 1,86256 | -1,01668 | -0,13345 | -0,04811 |
| -0,56901 | 3,36587 | -1,01212 | -4,02562 | 0,19618 |
| -0,89584 | 0,77543 | 0,34090 | 1,61566 | -1,01288 |
| -0,96689 | 3,25655 | -1,47261 | -4,47471 | 0,23614 |
| -1,22700 | 2,75044 | -0,22001 | 1,85382 | -1,08939 |

1. *Update* P2

Tabel 3.47 Hasil *Update* P1 Perulangan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,76868 | 0,75546 | 1,26657 | 1,86883 | -4,55035 |
| 0,15241 | 1,09551 | 2,08092 | -0,74074 | -0,14268 |
| -2,47503 | -2,99588 | -0,82873 | -1,15286 | 0,45107 |
| -1,66703 | -0,05272 | 0,60594 | 1,15821 | 1,50773 |
| 0,84731 | 1,47509 | -1,91363 | -0,00768 | -0,28957 |

1. Menghitung *Stochastic Gradient Descent* (SGD)

Menghitung SGD untuk mendapatkan *latent factor user* dan *item* yang baru.

Tabel 3.48 *Latent Factor User* Perulangan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,30859 | 0,30719 | -0,70051 | 0,40461 | 0,34824 |
| 0,19185 | 0,04884 | 0,01287 | -0,25855 | -0,53613 |
| -0,76236 | 0,16509 | 0,83916 | -0,16801 | 0,81920 |
| 0,00140 | -0,03334 | 0,00166 | -0,08271 | -1,11194 |
| -0,76143 | -0,50782 | -1,32904 | 0,01543 | -0,11095 |

Tabel 3.49 *Latent Factor Item* Perulangan 2

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,20130 | 0,05939 | 0,01026 | 0,14196 | -0,33610 |
| -0,57929 | 0,04274 | -0,18951 | -0,53004 | -0,16327 |
| -0,56798 | 0,70929 | 0,30996 | -0,87538 | -0,06039 |
| -0,68843 | 0,91547 | -0,62400 | -0,49039 | -0,10678 |
| -0,63892 | 0,42437 | -0,00863 | 0,40001 | 0,33974 |

1. Perulangan 3
2. Menghitung kesalahan model

Nilai kesalahan model yang didapat setelah dilakukan perhitungan adalah:

0,000389419

0,00016785

0,079340855

= 0,25197603

*Loss* = 0,000389419+ 0,00016785+ 0,079340855+ 0,25197603

*Loss* = 0,331874

1. *Update* W1

Tabel 3.50 Hasil *Update* W1 Perulangan 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,79986 | -0,19879 | -0,20066 | 0,19952 | 0,20145 |
| -0,19901 | 0,79775 | -0,19605 | 0,20260 | 0,19682 |
| -0,10031 | -0,09754 | 0,89624 | 0,09785 | 0,10350 |
| 0,29921 | 0,30465 | 0,29437 | 0,69778 | -0,29562 |
| 0,20114 | 0,19690 | 0,20527 | -0,19766 | 0,79716 |

1. *Update* W2

Tabel 3.51 Hasil *Update* W2 Perulangan 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 2,05215 | 0,00008 | -0,00020 | 0,00005 | 0,00000 |
| -0,00003 | 2,05185 | -0,00040 | 0,00010 | -0,00044 |
| -0,00007 | 0,00012 | 2,05200 | 0,00018 | -0,00009 |
| -0,00007 | -0,00015 | -0,00023 | 2,05226 | -0,00015 |
| -0,00013 | -0,00008 | -0,00028 | -0,00001 | 2,05192 |

1. *Update* P1

Tabel 3.52 Hasil *Update* P1 Perulangan 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,77685 | 1,96416 | -1,06199 | -0,23910 | -0,05199 |
| -0,61864 | 3,46021 | -1,03592 | -4,06754 | 0,17093 |
| -0,94227 | 0,82534 | 0,31576 | 1,58837 | -1,03281 |
| -1,03351 | 3,40758 | -1,54528 | -4,60229 | 0,22961 |
| -1,29319 | 2,85473 | -0,24938 | 1,80313 | -1,12374 |

1. *Update* P2

Tabel 3.53 Hasil *Update* P1 Perulangan 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,80335 | 0,75106 | 1,28465 | 1,88159 | -4,59649 |
| 0,13698 | 1,09775 | 2,06753 | -0,75840 | -0,16549 |
| -2,49412 | -3,00810 | -0,85001 | -1,16277 | 0,42282 |
| -1,69842 | -0,05402 | 0,62289 | 1,16416 | 1,46210 |
| 0,82030 | 1,48366 | -1,93218 | 0,00111 | -0,30411 |

1. Menghitung *Stochastic Gradient Descent* (SGD)

Menghitung SGD untuk mendapatkan *latent factor user* dan *item* yang baru.

Tabel 3.54 *Latent Factor User* Perulangan 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 0,30973 | 0,26338 | -0,67591 | 0,52806 | 0,32958 |
| 0,18227 | 0,39896 | -0,18358 | -1,24640 | -0,38709 |
| -0,76340 | 0,20876 | 0,81451 | -0,29130 | 0,83774 |
| 0,00277 | -0,08789 | 0,03226 | 0,07156 | -1,13515 |
| -0,76397 | -0,40943 | -1,38410 | -0,26235 | -0,06902 |

Tabel 3.55 *Latent Factor Item* Perulangan 3

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,20137 | 0,05940 | 0,01036 | 0,14192 | -0,33599 |
| -0,57915 | 0,04275 | -0,18963 | -0,52995 | -0,16330 |
| -0,56784 | 0,70921 | 0,30984 | -0,87529 | -0,06052 |
| -0,68827 | 0,91547 | -0,62374 | -0,49035 | -0,10691 |
| -0,63887 | 0,42434 | -0,00863 | 0,39998 | 0,33970 |

1. Perulangan 4
2. Menghitung kesalahan model

Nilai kesalahan model yang didapat setelah dilakukan perhitungan adalah:

1,616704161

0,00012902

0,08526366

= 0,28848225

*Loss* = 1,616704161 + 0,00012902+ 0,08526366+ 0,28848225

*Loss* = 1,990579

Karena nilai kesalahan model yang didapat meningkat atau lebih tinggi dari sebelumnya maka perulangan berhenti dan *latent factor user* dan *item* diambil dari perulangan sebelumnya. Lalu dilakukan perkalian *latent factor user* dengan *latent factor item*  untuk mendapatkan matriks *rating*. Matriks *rating* dapat dilihat pada tabel 3.56.

Tabel 3.56 Matriks *Rating* Hasil Perkalian *Latent Factor User* dan *Item*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| -0,0895 | -0,3736 | -0,6806637 | 0,15536192 | 0,242889 |
| -0,0617 | 0,67004 | 1,23695782 | 1,00685072 | -0,575585 |
| -0,1482 | 0,31417 | 1,03818364 | 0,26176598 | 0,737335 |
| 0,38611 | 0,13596 | -0,0478527 | -0,01622271 | -0,396337 |
| 0,10114 | 0,83771 | -0,0515934 | 1,15033684 | 0,197905 |

Matriks *rating* yang didapatkan akan dilakukan normalisasi kembali menggunakan metode *min-max normalization* sehingga nilai *rating* yang didapatkan berada di dalam skala 1-10. Hasil normalisasi matriks dapat dilihat pada tabel 3.57.

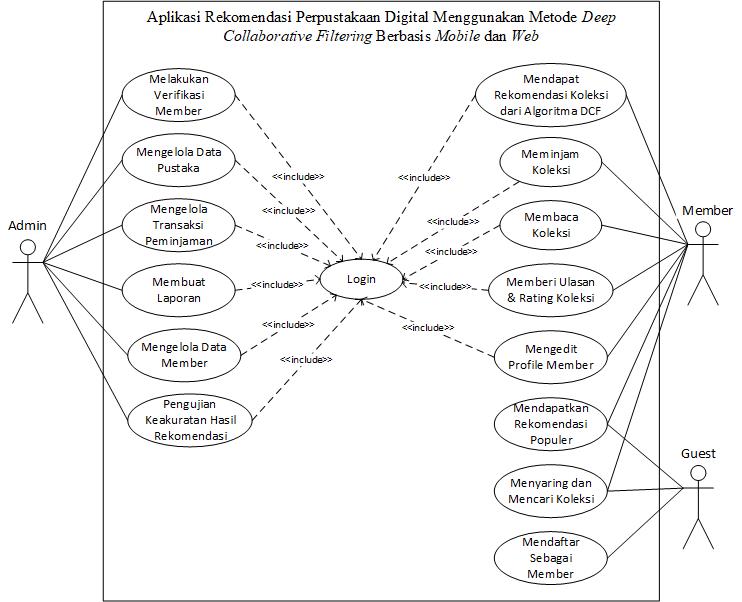
Tabel 3.57 Hasil Normalisasi Matriks *Rating* Menggunakan *Min-max Normalization*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3,774517 | 2,441146 | 1 | 4,923731 | 5,334523 |
| 3,904991 | 7,339277 | 10 | 8,920035 | 1,493167 |
| 3,499019 | 5,669067 | 9,06709 | 5,423118 | 7,655113 |
| 6,006704 | 4,832671 | 3,969981 | 4,11843 | 2,334434 |
| 4,66925 | 8,126205 | 3,952424 | 9,593461 | 5,123399 |

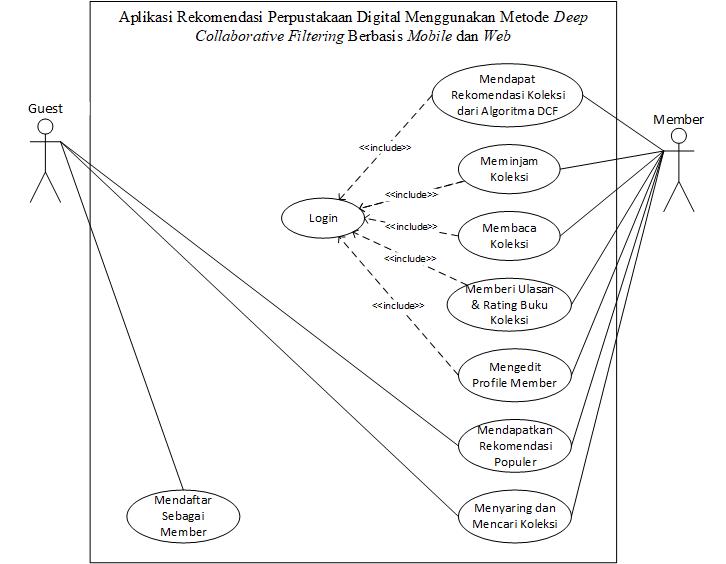
### Analisis Kebutuhan

#### Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan dengan menggunakan *use case diagram* untuk memperlihatkan interaksi antar *user* dan sistem. Berikut *use case diagram web* dan *mobile* pada aplikasi perpustakaan digital menggunakan *deep collaborative filtering*.



Gambar 3.2 *Use case Diagram Web*



Gambar 3.3 *Use case Diagram Mobile*

1. Aktor pada *Use Case Diagram* web dan *mobile.*

Tabel 3.58 Aktor Pada *Use Case* *Diagram Web* dan *Mobile*

|  |  |  |
| --- | --- | --- |
| No. | Aktor | Deskripsi |
| 1 | *Admin* | *User* sistem yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan data pustaka, *Member*, dan proses peminjaman pustaka pada sistem. |
| 2 | *Member* | *User* sistem yang telah mendaftar pada sistem dan telah melengkapi identitas diri dan terverifikasi serta memiliki hak meminjam dan membaca *item* digital sesuai hak aksesnya. |
| 3 | *Guest* | *User* sistem yang tidak terdaftar pada sistem dan memiliki akses untuk melihat daftar koleksi, mencari koleksi tanpa memiliki hak meminjam koleksi. |

1. *Use Case* pada *Web* dan *Mobile*

Tabel 3.59 *Use Case* pada Web dan Mobile

|  |  |  |
| --- | --- | --- |
| No. | *Use Case* | Deskripsi |
| 1 | *Login* | Proses untuk masuk pada sistem sesuai hak akses yang dimiliki *member* sistem. |
| 2 | Mengelola Data *Member* | Proses yang dilakukan *admin* untuk melakukan verifikasi penanda identitas *member*. |
| 3 | Melakukan verifikasi *Member* | Proses yang dapat dilakukan oleh *admin* untuk kontrol *Member* yang telah mendaftar ke sistem meliputi melihat data *member*, dan menghapus *member*. |
| 4 | Mengelola Data Pustaka | Proses yang dapat dilakukan oleh *admin* untuk mengelola data Pustaka meliputi menambah koleksi digital, mengubah informasi koleksi digital, hingga menghapus data pustaka |
| 5 | Mengelola Transaksi Peminjaman | Proses yang dapat dilakukan oleh *admin* untuk memantau data peminjaman meliputi mencari peminjaman dan menghapus data peminjaman |
| 6 | Membuat Laporan | Proses yang dapat dilakukan oleh *admin* untuk mengetahui laporan transaksi peminjaman, dan *member* |
| 7 | Mendapat Rekomendasi Koleksi dari Algoritma DCF | Proses untuk mendapatkan rekomendasi berdasarkan algoritma deep collaborative filtering. |
| 8 | Mendapatkan Rekomendasi Populer | Proses untuk mendapatkan rekomendasi berdasarkan Trend Populer |
| 9 | Meminjam Koleksi | Proses yang dapat dilakukan oleh *member* untuk meminjam koleksi digital yang kemudian masuk pada daftar bacaan *member*. |
| 10 | Membaca Koleksi | Proses yang dapat dilakukan oleh *member* untuk membaca koleksi digital secara langsung pada sistem. |
| 11 | Memberi Ulasan & *Rating* Koleksi | Proses yang dapat dilakukan oleh *member* untuk memberikan *rating* dan ulasan tentang koleksi yang telah dipinjam. |
| 12 | Mendaftar Sebagai *Member* | Proses yang dapat dilakukan oleh guest untuk mendaftar menjadi *member* dengan mengisi informasi yang telah disediakan oleh sistem. |
| 13 | Menyaring dan Mencari Koleksi Digital | Proses yang dapat dilakukan oleh guest dan *member* untuk menyaring koleksi berdasarkan kategori koleksi, tahun terbit, penulis, dan penerbit. |
| 14 | Mengedit Profile *Member* | Proses yang dapat dilakukan oleh *member* untuk mengedit profile. |
| 15 | Pengujian Keakuratan Hasil Rekomendasi | Proses yang dapat dilakukan oleh *admin* untuk melakukan pengujian rekomendasi *item* digital dengan menggunakan *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE) |

1. *Use Case* Skenario Pada *Use Case Web* Dan *Mobile*

Untuk menjelaskan setiap *use case* yang terdapat pada *use case diagram*, maka digunakan narasi pada *use case* Aplikasi Rekomendasi Perpustakaan Digital Menggunakan Metode *Deep Collaborative Filtering* Berbasis *Mobile* dan *Web*.

Tabel 3.60 Narasi *Use Case* *Login*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | ***Login*** | |
| Aktor | ***Member* dan *Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses untuk masuk pada sistem sesuai hak akses yang dimiliki *user* sistem. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman *login* |  |
|  | Menampilkan halaman *login*. |
| Mengisi data pada *form* *login*.  mengetuk tombol “*login*” |  |
|  | Melakukan pengecekan berdasarkan *Username* dan *password* |
|  | Jika *login* sebagai *member* akan masuk ke halaman beranda *member*; Jika *login* sebagai admin akan masuk ke halaman *dashboard admin* |
| *Alternative Flow Event* |  | Menampilkan *form login* dengan pesan kesalahan |

Tabel 3.61 Narasi *Use Case* MengelolaData *Member*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Mengelola Data *Member*** | |
| Aktor | ***Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh a*dmin* untuk kontrol pada setiap *member* yang telah mendaftar ke sistem meliputi melihat data *member*, mengubah data *member*, hingga menghapus *member*. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman *login* |  |
| Masuk ke halaman kelola *member*. |  |
|  | Menampilkan halaman kelola *member*. |
| Melakukan kontrol seperti melihat data *member*, mengubah data *member*, atau menghapus *member*. |  |
|  | Menyimpan perubahan. |

Tabel 3.62 Narasi *Use Case* Melakukan Verifikasi *Member*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Melakukan Verifikasi *Member*** | |
| Aktor | ***Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dilakukan *admin* untuk melakukan verifikasi pada identitas *member* yang baru mendaftar. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman kelola *member*. |  |
|  | Menampilkan halaman kelola *member*. |
| Mengetuk tombol “verifikasi *member* baru” |  |
|  | Menampilkan halaman verifikasi *member* baru. |
| Mengecek identitas *member* yang baru mendaftar |  |
| Mengetuk tombol verifikasi |  |
|  | Menampilkan pesan verifikasi *member* berhasil. |

Tabel 3.63 Narasi *Use Case* Mengelola Data Pustaka

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Mengelola Data Pustaka** | |
| Aktor | ***Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *admin* untuk mengelola koleksi digital meliputi penambahan koleksi digital, mengubah informasi koleksi digital, hingga menghapus koleksi digital. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman koleksi digital |  |
|  | Menampilkan halaman koleksi digital |
| Melakukan kontrol meliputi penambahan koleksi digital, mengubah informasi koleksi digital, hingga menghapus koleksi digital. |  |
|  | Menyimpan perubahan. |

Tabel 3.64 Narasi *Use Case* Mengelola Transaksi Peminjaman

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Mengelola Transaksi Peminjaman** | |
| Aktor | ***Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *admin* untuk melihat data peminjaman dan mencari data peminjaman. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman kelola peminjaman. |  |
|  | Menampilkan halaman kelola peminjaman. |
| Melakukan kontrol |  |

Tabel 3.65 Narasi *Use Case* Membuat Laporan

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Membuat Laporan** | |
| Aktor | ***Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *admin* untuk mengetahui laporan transaksi peminjaman, dan *member*. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman kelola laporan. |  |
|  | Menampilkan ke halaman kelola laporan |
| Melihat dan mencetak laporan yang akan dibuat. |  |
|  | Mendapatkan laporan peminjaman dalam bentuk file pdf. |

Tabel 3.66 Narasi *Use Case* Mendapat Rekomendasi Koleksi dari Algoritma DCF

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Mendapat Rekomendasi Koleksi dari Algoritma DCF** | |
| Aktor | ***Member*** | |
| Deskripsi | *Use case* ini menjelaskan proses untuk mendapatkan rekomendasi berdasarkan algoritma DCF. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Melakukan login sistem sebagai *member*. |  |
|  | Menampilkan halaman daftar koleksi dari hasil rekomendasi koleksi berdasarkan algoritma DCF. |
| *Member* meminjam koleksi dan memberi *rating* atau ulasan. |  |
|  | Menampilkan halaman daftar koleksi dari hasil rekomendasi berdasarkan algoritma DCF. |

Tabel 3.67 Narasi Use Case Meminjam Koleksi

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Meminjam Koleksi** | |
| Aktor | ***Member*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *member* untuk meminjam koleksi digital dan masuk pada daftar baca *member* | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman rincian koleksi. |  |
|  | Menampilkan informasi koleksi yang dipilih. |
| Mengetuk tombol pinjam koleksi. |  |
|  | Kembali ke halaman rincian koleksi dan menampilkan pesan bahwa peminjaman berhasil. |

Tabel 3.68 Narasi *Use Case* Membaca Koleksi

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Membaca Koleksi** | |
| Aktor | ***Member*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *member* untuk membaca koleksi digital secara langsung pada sistem. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow Event* | Masuk ke halaman daftar koleksi bacaan (terpinjam) atau halaman rincian koleksi. |  |
|  | Menampilkan daftar koleksi bacaan yang telah dipinjam. |
| Mengetuk Tombol baca. |  |
|  | Menampilkan koleksi digital dalam book reader. |

Tabel 3.69 Narasi *Use Case* Memberi Ulasan & *Rating* Koleksi

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Memberi Ulasan & *Rating* Koleksi** | |
| Aktor | ***Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *member* untuk memberikan ulasan dan *rating* tentang koleksi yang telah dipinjam. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman rincian koleksi yang telah dipinjam. |  |
|  | Menampilkan rincian koleksi bacaan yang telah dipinjam. |
| Memasukkan *rating* & ulasan terhadap koleksi yang telah dipinjam. |  |
| Mengetuk tombol post ulasan. |  |
|  | Menampilkan kembali laman rincian koleksi dengan pesan ulasan berhasil di tambah. |

Tabel 3.70 Narasi *Use Case* Mendaftar Sebagai *Member*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Mendaftar Sebagai *Member*** | |
| Aktor | ***Guest*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *user* untuk mendaftar menjadi *member* dengan mengisi informasi yang telah disediakan oleh sistem. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman pendaftaran *member* baru. |  |
|  | Menampilkan halaman pendaftaran *member* baru. |
| Mengisi data pada form pendaftaran. |  |
| Mengetuk tombol daftar. |  |
|  | Sistem menyimpan data. |
| Melakukan verifikasi pendaftaran melalui e-mail. |  |
|  | Menampilkan pesan konfirmasi e-mail berhasil. |
| *Alternative Flow Event* |  | Menampilkan form pendaftaran dengan pesan kesalahan |

Tabel 3.71 Narasi *Use Case* Mendaftar Sebagai *Member*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Mendapatkan Rekomendasi Populer** | |
| Aktor | ***Member* dan *Guest*** | |
| Deskripsi | *Use case* ini menjelaskan proses menampilkan / melihat rekomendasi koleksi dari hasil *rating* koleksi yang telah ada untuk ditampilkan kepada *member* dan *guest*. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman koleksi populer. |  |
|  | Menampilkan rekomendasi koleksi dengan DCF. |

Tabel 3.72 Narasi *Use Case* Melihat Detail Koleksi

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Melihat Detail Koleksi** | |
| Aktor | ***Member* dan *Guest*** | |
| Deskripsi | *Use case* ini menjelaskan proses menampilkan / melihat rincian informasi koleksi untuk ditampilkan kepada *member* dan *user*. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman *homepage* |  |
|  | Melihat daftar koleksi |
| Mengetuk koleksi |  |
|  | Menampilkan rincian informasi koleksi |

Tabel 3.73 Narasi *Use Case* Menyaring dan Mencari Koleksi

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Menyaring dan Mencari Koleksi** | |
| Aktor | ***Member* dan *Guest*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *member* dan *guest* untuk menyaring dan mencari koleksi berdasarkan kategori koleksi, tahun terbit, penulis, dan penerbit yang akan ditampilkan pada hasil pencarian koleksi. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman homepage |  |
|  | Melihat daftar koleksi |
| Mengisi informasi penyaringan pada jendela filtering |  |
|  | Menampilkan hasil penyaringan atau pencarian koleksi. |

Tabel 3.74 Narasi *Use Case* Mengedit Profil *Member*

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Mengedit Profil *Member*** | |
| Aktor | ***Member*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *member* untuk mengedit profile | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Melakukan login sistem sebagai *member* |  |
|  | Menampilkan halaman beranda *member* |
| Masuk ke halaman edit profil |  |
|  | Menampilkan halaman edit profil. |
| Melakukan pengeditan informasi profile. |  |
| Mengetuk tombol save. |  |
|  | Menyimpan hasil pengeditan profile |

Tabel 3.75 Narasi *Use Case* Melakukan Pengujian Rekomendasi

|  |  |  |
| --- | --- | --- |
| Nama *use case* | **Melakukan Pengujian Rekomendasi** | |
| Aktor | ***Admin*** | |
| Deskripsi | *Use case* ini menjelaskan proses yang dapat dilakukan oleh *admin* untuk melakukan pengujian menggunakan *Mean Absolute Error* (MAE) dan *Root Mean Square Error* (RMSE). | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow* | Masuk ke halaman Pengujian Rekomendasi. |  |
|  | Menampilkan halaman pengujian |
| Mengetuk combo box “MAE” |  |
|  | Menampilkan halaman “MAE”. |
| Mengetuk tombol “Hitung” |  |
|  | Menampilkan Pengujian “MAE”. |
| Mengetuk combo box “RMSE” |  |
|  | Menampilkan halaman “RMSE”. |
| Mengetuk tombol “Hitung” |  |
|  | Menampilkan Pengujian “MAE”. |

#### Analisis Kebutuhan *Non*-Fungsional

Analisis kebutuhan *non*-fungsional dilakukan menggunakan metode analisis PIECES (*Performance, Information, Economy, Control, Efficiency, Services*). PIECES digunakan untuk menganalisa aplikasi yang akan dibangun agar dapat diketahui permasalahannya dengan spesifik dan lebih jelas. Hasil analisanya adalah sebagai berikut:

Tabel 3.76 Analisis Kebutuhan Non-Fungsionalitas dengan Kerangka PIECES

|  |  |
| --- | --- |
| Parameter | Penjelasan |
| *Performance* | Dengan adanya sistem rekomendasi ini, akan membantu *user* di dalam menemukan *item* digital dengan tidak memakan waktu yang lama. Algoritma *Deep Collaborative Filtering* yang diterapkan dalam aplikasi dapat memberikan rekomendasi andal. |
| *Information* | Memberikan informasi berupa rekomendasi *item*, informasi *item*, *list item* yang dipinjam, daftar peminjaman, serta layanan yang disediakan oleh setiap *admin* yang terdapat di dalam sistem. |
| *Economy* | *User* aplikasi hanya memerlukan koneksi internet dengan akses browser untuk menggunakan aplikasi |
| *Control* | Adanya pemberitahuan seperti notifikasi pada aplikasi di setiap perubahan yang terjadi. |
| *Efficiency* | *User* aplikasi dapat mengakses informasi *item* yang akan dipinjam dan mendapatkan rekomendasi *item* berdasarkan *rating* dan *item* terkait melalui *website* maupun *mobile-android.* |
| *Services* | Tampilan antarmuka dibuat *user-friendly* agar *user* atau *admin* dapat dengan mudah memahami dan menggunakan aplikasi. |

## Perancangan

Pada tahapan perancangan pada pengembangan aplikasi terbagi menjadi dua bagian yaitu sebagai berikut.

### Perancangan Tampilan

Perancangan tampilan berperan menampilkan *mockup* tampilan dalam program yang akan dirancang. Tampilan dirancang dengan jelas agar dapat memudahkan *user* dalam mengakses-nya. Perancangan tampilan terbagi menjadi 2 (dua) bagian yaitu sebagai berikut.

#### Tampilan Website

1. Halaman Utama *Web*

Halaman utama *web* merupakan halaman yang pertama kali dilihat oleh *user* ketika mengakses *web.*



Gambar 3.4 Perancangan Halaman Utama *Website*

Keterangan:

1. Logo *website* yang berfungsi mengarahkan *user* ke halaman utama *website.*
2. Teks *input* untuk pencarian *item* digital.
3. Teks *link* “kategori” yang berfungsi mengarahkan *user* ke halaman kategori *item* digital.
4. Teks *link* “tentang” yang berfungsi mengarahkan *user* ke halaman informasi *website.*
5. Teks *link* “kontak” yang berfungsi mengarahkan *user* ke halaman kontak *website.*
6. *Icon* “notifikasi” yang menampilkan daftar notifikasi dari *website.*
7. *Icon* “*user*” yang menampilkan *pop up menu* yang terdiri dari teks *link* masuk dan daftar.
8. Teks *link* “masuk” yang berfungsi mengarahkan *user* ke halaman masuk.
9. Teks *link* “daftar” yang berfungsi mengarahkan *user* ke halaman pendaftaran.
10. *Slider* *item* digital berdasarkan *rating* tertinggi.
11. Tombol “pratinjau” yang berfungsi mengarahkan *user* ke halaman detail *item* digital.
12. *Pagination bullet* sebagai petunjuk *item* yang sedang tampil pada *slider.*
13. *Item* digital yang baru ditambahkan, yang terdiri daftar *item* digital dan informasi ringkas dari *item* digital terbaru.
14. *Rekomendasi item* digital berdasarkan *rating* tertinggi, yang terdiri dari daftar *item* digital dan informasi ringkas dari *item* digital.
15. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
16. Halaman Pendaftaran

Halaman pendaftaran merupakan halaman yang muncul saat *user* mengklik teks *link* daftar pada tampilan utama *web*. Pada halaman ini *user* dapat mendaftar sebagai *member* dengan mengisi *form* pendaftaran dan menekan tombol daftar.



Gambar 3.5 Perancangan Halaman Pendaftaran

Keterangan

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Teks *input* yang terdiri dari nama depan, nama belakang, *Username*, jenis kelamin, tempat lahir, tanggal lahir, agama, hobi, negara, provinsi, kota, alamat, *email*, nomor telepon, password, konfirmasi password, *upload* foto profil dan upload foto KTP.
3. Tombol “daftar” yang berfungsi untuk mengirim data diri yang telah di isi oleh *user*.
4. Teks *link* yang berfungsi mengarahkan *user* ke halaman masuk.
5. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
6. Halaman Masuk

Tampilan login merupakan tampilan yang muncul saat *user* mengklik teks *link* masuk pada tampilan utama *web*. Pada halaman ini *user* dapat masuk ke dalam sistem dengan mengisi *Username* dan *password* yang telah terdaftar.



Gambar 3.6 Perancangan Halaman Masuk

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Tampilan ilustrasi *login*.
3. Teks *input* yang terdiri dari *Username* dan *password*.
4. Teks *link* yang berfungsi mengarahkan *user* ke halaman lupa kata sandi.
5. Tombol “masuk” yang berfungsi untuk mengirim data yang telah di isi oleh *user* dan mengarahkan *user* ke halaman beranda *member*.
6. Tombol “daftar” yang berfungsi mengarahkan *user* ke halaman pendaftaran.
7. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
8. Halaman Beranda *Member*

Halaman beranda *member* merupakan halaman yang muncul sesudah *member* telah masuk ke dalam sistem. Pada halaman ini *member* akan mendapatkan rekomendasi *item* digital.



Gambar 3.7 Perancangan Halaman Beranda *Member*

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Informasi yang terdiri dari foto profil, nama, dan *Username* dari *member*, berfungsi mengarahkan *member* ke halaman *profil* *member*.
3. Teks *link* “*item* digital tersimpan” yang berfungsi mengarahkan *member* ke halaman *item* digital terpinjam.
4. Teks *link* “keluar” yang berfungsi mengarahkan *member* keluar dari *website*.
5. *Slider* *item* digital berdasarkan *rating* tertinggi.
6. *Item* digital yang baru ditambahkan, yang terdiri daftar *item* digital dan informasi ringkas dari *item* digital terbaru.
7. Rekomendasi *item* digital, yang terdiri dari daftar *item* digital yang direkomendasikan berdasarkan algoritma *deep collaborative filtering*.
8. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
9. Halaman Profil

Halaman *profil* merupakan halaman yang digunakan *member* untuk melihat informasi data diri *member* di dalam *website*.



Gambar 3.8 Perancangan Halaman Profil

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Tampilan dari foto *profil member.*
3. Informasi data diri *member* yang terdiri dari nama lengkap, *Username*, jenis kelamin, tempat lahir, tanggal lahir, agama, alamat, kota, provinsi, negara, dan *email*.
4. Tampilan dari foto KTP *member*.
5. Tombol “*edit profil*” yang berfungsi mengarahkan *user* ke halaman *edit profil*.
6. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
7. Halaman Kategori *Item* Digital

Halaman kategori *item* digital merupakan halaman yang digunakan *user* untuk melihat daftar *item* digital sesuai dengan kategori *item* digital.



Gambar 3.9 Perancangan Halaman Kategori *Item* Digital

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Combo box “kategori” yang terdiri dari semua kategori *item* digital.
3. Daftar *item* digital yang ditampilkan sesuai dengan kategori dan pengurutan yang terpilih.
4. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
5. Halaman Pencarian

Halaman pencarian merupakan halaman yang digunakan *user* untuk melakukan pencarian terhadap *item* digital, berdasarkan judul *item* digital, penulis *item* digital atau nomor ISBN *item* digital.



Gambar 3.10 Perancangan Halaman Pencarian

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Teks *input* yang terdiri dari judul *item* digital, penulis, dan ISBN *item* digital yang dicari.
3. Tombol “cari” yang berfungsi menampilkan daftar *item* digital sesuai dengan judul *item* digital, penulis, atau ISBN yang di input oleh *user*.
4. Daftar *item* digital yang ditampilkan sesuai dengan judul *item* digital, penulis, atau ISBN yang di input oleh *user*.
5. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
6. Halaman Detail *Item* Digital

Halaman detail *item* digital merupakan halaman yang digunakan *user* untuk melihat informasi detail dari suatu *item* digital. Pada halaman ini juga *member* dapat memberikan *rating* dan ulasan terhadap suatu *item* digital



Gambar 3.11 Rancangan Halaman Detail *Item* Digital

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Tampilan foto sampul dari *item* digital.
3. Tombol “pinjam” yang berfungsi untuk melakukan peminjaman *item* digital.
4. Informasi dari *rating* *item* digital.
5. Informasi detail dari *item* digital yang terdiri dari judul, kategori, penulis, penerbit, tahun terbit, ISBN, dan ringkasan.
6. Tombol “bintang” untuk memberikan *rating* terhadap *item* digital.
7. Teks *input* “ulasan” untuk memberikan komentar terhadap *item* digital.
8. Tombol “kirim *feedback*” yang berfungsi untuk mengirim *rating* dan ulasan yang diberikan oleh *member*.
9. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
10. Halaman Baca *Item* Digital

Halaman baca *item* digital merupakan halaman yang digunakan *member* untuk membaca *item* digital yang telah di pinjam.



Gambar 3.12 Perancangan Halaman Baca *Item* Digital

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Informasi ringkas dari *item* digital yang dibaca, terdiri dari judul *item* digital, penulis dan tahun terbit.
3. *Book reader* yang berfungsi menampilkan konten dari *item* digital yang dibaca.
4. Navigasi yang berfungsi mengarahkan *member* ke halaman sebelum atau selanjutnya dari *item* digital yang dibaca.
5. Tombol “layar penuh” yang berfungsi menampilkan *book reader* dalam *mode* penuh.
6. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
7. Halaman Tentang

Halaman kontak merupakan halaman yang digunakan *user* untuk melihat informasi tentang perpustakaan digital.

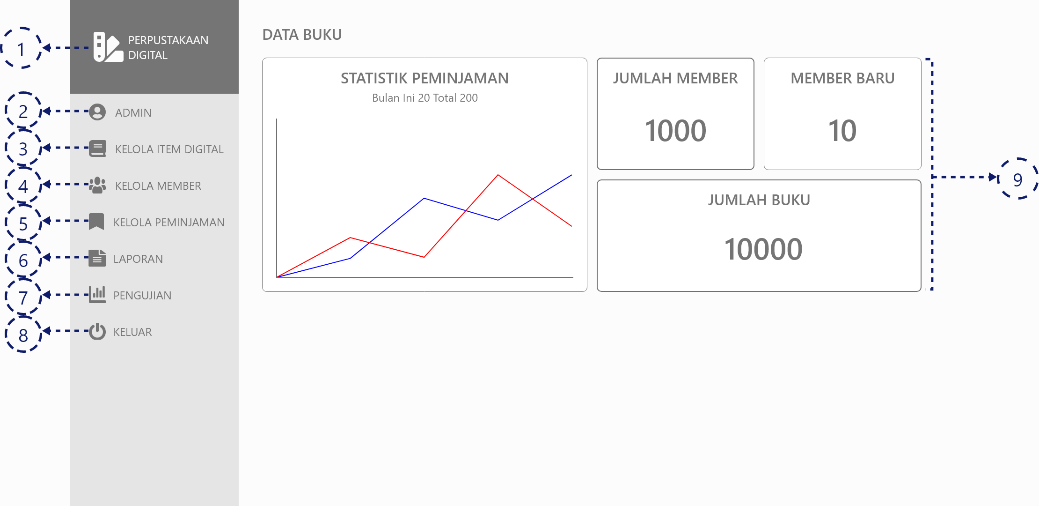


Gambar 3.13 Rancangan Halaman Tentang

Keterangan:

1. Header yang berisi logo, pencarian, dan navigasi dari *website*.
2. Tampilan logo perpustakaan digital.
3. Teks informasi tentang perpustakaan digital
4. *Footer* yang terdiri dari info, *profil* dan navigasi *website.*
5. Halaman Dashboard Admin

Halaman *dashboard admin* merupakan halaman yang muncul sesudah *admin* melakukan *login* telah masuk ke dalam sistem.

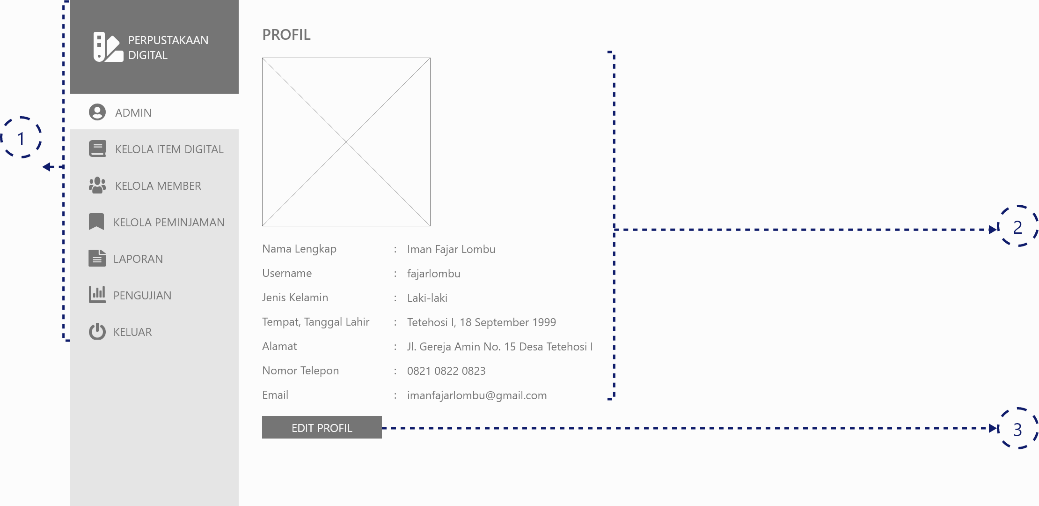


Gambar 3.14 Rancangan Halaman Dashboard Admin

Keterangan:

1. Logo *website* yang berfungsi mengarahkan *admin* ke halaman *dashboard admin.*
2. Teks *link* “*admin*” yang berfungsi mengarahkan *admin* ke halaman *profil* *admin*.
3. Teks *link* “kelola *item* digital” yang berfungsi mengarahkan *admin* ke halaman kelola item digital.
4. Teks *link* “kelola *member*” yang berfungsi mengarahkan *admin* ke halaman kelola *member*.
5. Teks *link* “kelola peminjaman” yang berfungsi mengarahkan *admin* ke halaman peminjaman.
6. Teks *link* “laporan” yang berfungsi mengarahkan *admin* ke halaman laporan.
7. Teks *link* “pengujian” yang berfungsi mengarahkan *admin* ke halaman pengujian.
8. Teks *link* “keluar” yang berfungsi mengarahkan *admin* keluar dari *website*.
9. Dashboard *admin* yang terdiri dari statistik peminjaman, jumlah *member*, jumlah *member* baru, dan jumlah buku.
10. Halaman *Profil* *Admin*

Halaman *profil admin* merupakan halaman yang digunakan *admin* untuk melihat informasi data diri *admin* di dalam website.



Gambar 3.15 Rancangan Halaman *Profil* *Admin*

Keterangan:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. Informasi data diri *admin* yang terdiri dari foto *profil*, nama lengkap, *username*, jenis kelamin, tempat lahir, tanggal lahir, nomor telepon, dan *email*.
3. Tombol “edit *profil*” yang berfungsi mengarahkan *admin* ke halaman edit *profil*.
4. Halaman *Item* Digital

Halaman *item* digital merupakan halaman yang digunakan *admin* untuk melihat semua daftar *item* digital yang telah terdaftar.

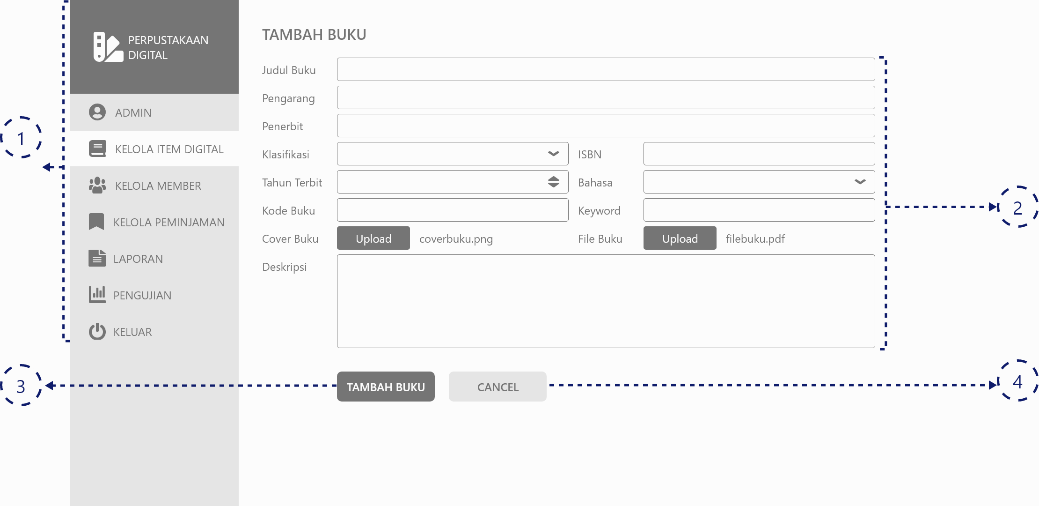


Gambar 3.16 Rancangan Halaman *Item* Digital

Keterangan:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. Tombol “tambah *item* digital” yang berfungsi mengarahkan *admin* ke halaman tambah *item* digital.
3. Teks *input* “cari *item* digital” yang berfungsi untuk melakukan pencarian *item* digital.
4. Tabel daftar *item* digital yang telah terdaftar yang terdiri dari kode *item* digital, ISBN, judul *item* digital, kategori, penulis dan penerbit.
5. *Pagination* yang berfungsi melakukan perpindahan antar - halaman daftar *item* digital.
6. Halaman Tambah *Item* digital

Halaman tambah *item* digital merupakan halaman yang digunakan *admin* untuk melakukan penambahan *item* digital baru.

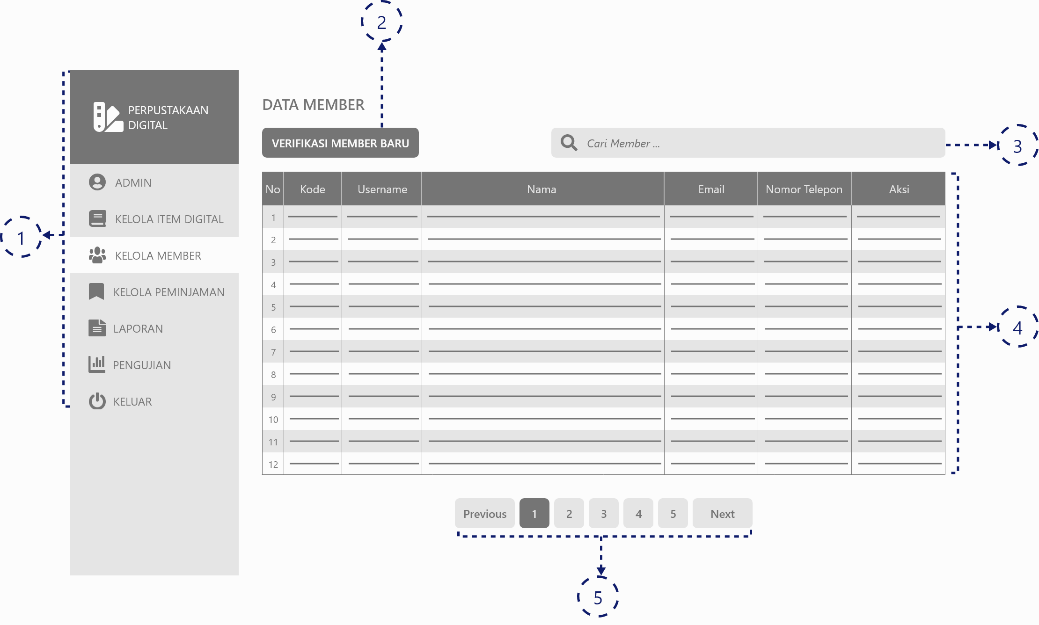


Gambar 3.17 Rancangan Halaman Tambah *Item* digital

Keterangan:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. Teks *input* yang terdiri dari judul *item* digital, pengarang, penerbit, kategori, ISBN, tahun terbit, bahasa, kode *item* digital, *keyword*, deskripsi, upload cover *item* digital dan upload file *item* digital.
3. Tombol “tambah *item* digital” berfungsi untuk menambahkan *item* digital.
4. Tombol “*cancel*” berfungsi untuk membatalkan penambahan *item* digital dan mengarahkan *admin* ke halaman daftar *item* digital.
5. Halaman Data *Member*

Halaman data *member* merupakan halaman yang digunakan *admin* untuk melihat semua daftar *member* yang telah terdaftar.

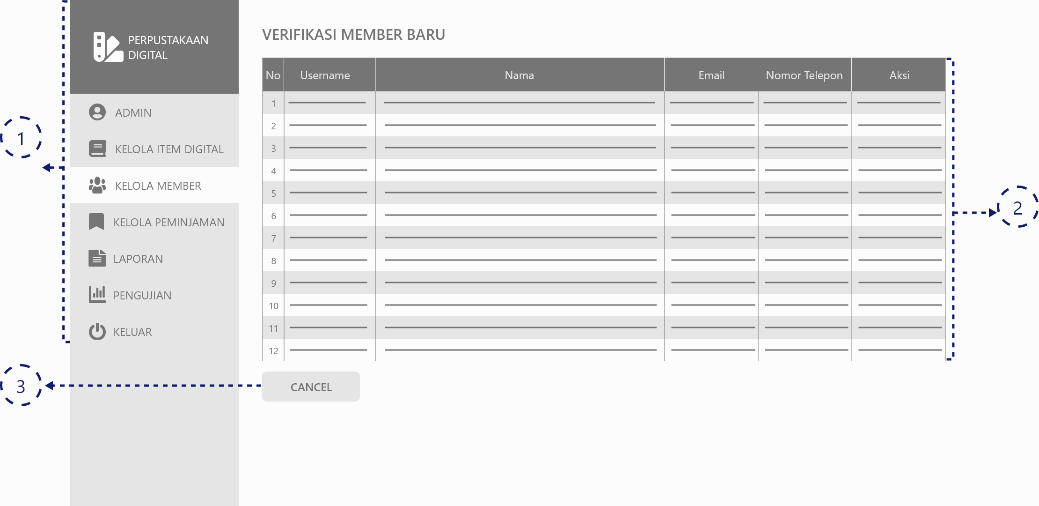


Gambar 3.18 Rancangan Halaman Data *Member*

Keterangan:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. Tombol “verifikasi *member* baru” yang berfungsi mengarahkan *admin* ke halaman verifikasi *member* baru.
3. Teks *input* “cari *member*” yang berfungsi untuk melakukan pencarian *member*.
4. Tabel daftar *member* yang telah mendaftar dan diverifikasi oleh admin, yang terdiri dari kode, *Username*, nama, *email*, nomor telepon dan aksi.
5. *Pagination* yang berfungsi melakukan perpindahan antar - halaman daftar *member*.
6. Halaman Verifikasi *Member* Baru

Halaman verifikasi *member* baru merupakan halaman yang digunakan *admin* untuk memverifikasi *member* baru yang telah mendaftar.

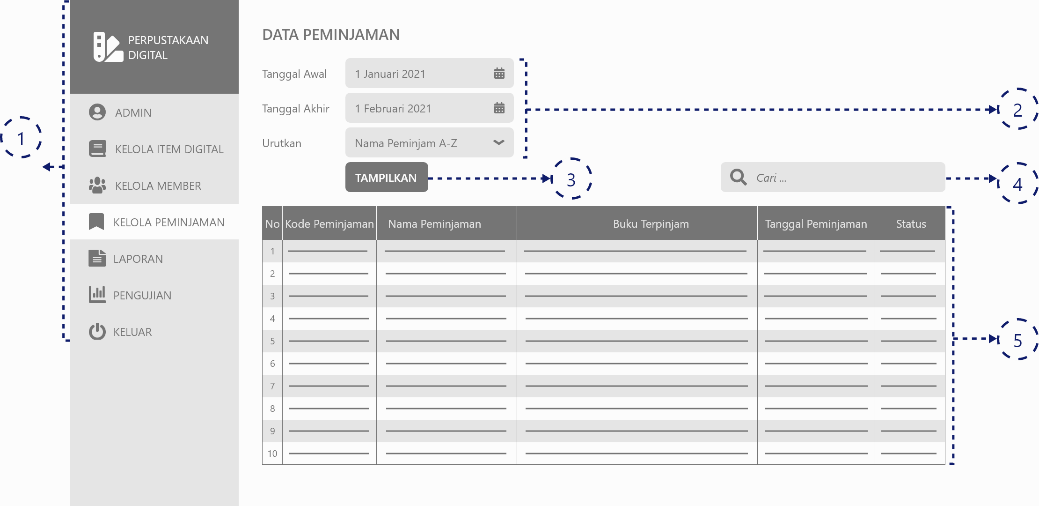


Gambar 3.19 Rancangan Halaman Verifikasi *Member* Baru

Keterangan:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. Tabel daftar *member* yang telah mendaftar, yang terdiri dari *Username*, nama, *email*, nomor telepon dan aksi.
3. Tombol “*cancel*” berfungsi untuk membatalkan verifikasi *member* dan mengarahkan *admin* ke halaman daftar *member*.
4. Halaman Data Peminjaman

Halaman data peminjaman merupakan halaman yang digunakan *admin* untuk melihat daftar transaksi peminjaman *item* digital oleh *member*

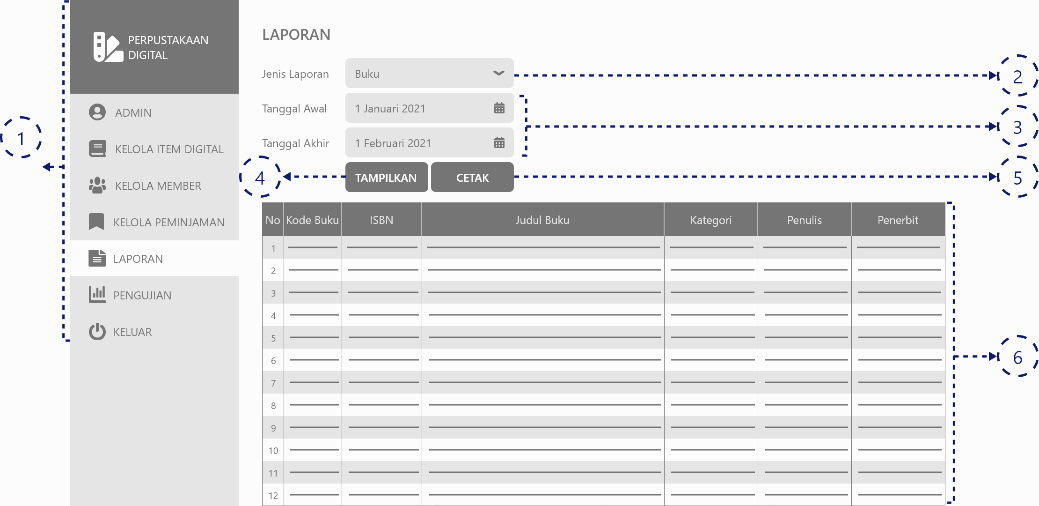


Gambar 3.20 Rancangan Halaman Peminjaman

Keterangan:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. Teks *input* yang terdiri dari tanggal awal, tanggal akhir, dan pengurutan.
3. Tombol “tampilkan” yang berfungsi menampilkan daftar tabel sesuai dengan tanggal awal, tanggal akhir, dan pengurutan.
4. Teks *input* pencarian untuk melakukan pencarian dalam tabel transaksi peminjaman
5. Tabel transaksi peminjaman yang terdiri dari kode peminjaman, nama *member*, *item* digital terpinjam, tanggal peminjaman, dan status
6. Halaman Laporan

Halaman laporan merupakan halaman yang digunakan *admin* untuk mencetak laporan *item* digital dan *member* perpustakaan digital.

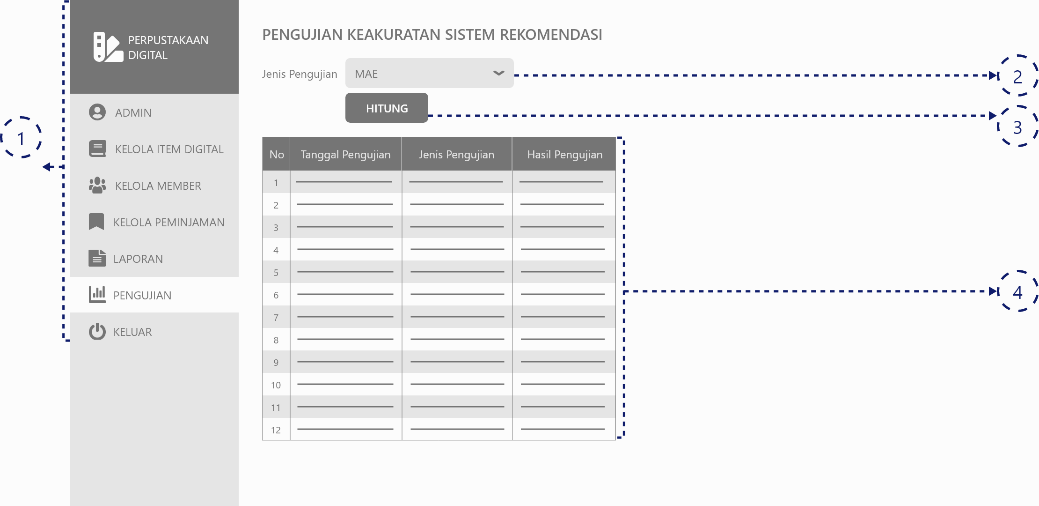


Gambar 3.21 Rancangan Halaman Laporan

Keterangan:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. *Combo box* “jenis laporan” yang berfungsi untuk memilih jenis laporan yang akan di tampilkan atau di cetak.
3. Teks *input* yang terdiri dari tanggal awal dan tanggal akhir laporan.
4. Tombol “tampilkan” yang berfungsi untuk menampilkan daftar laporan.
5. Tombol “cetak” yang berfungsi untuk mencetak laporan dalam bentuk pdf.
6. Tabel laporan *admin*.
7. Halaman Pengujian Keakuratan Sistem Rekomendasi

Halaman pengujian keakuratan sistem rekomendasi merupakan halaman yang digunakan *admin* untuk melakukan pengujian keakuratan hasil sistem rekomendasi *deep collaborative filtering*.



Gambar 3.22 Rancangan Pengujian Keakuratan Sistem Rekomendasi

Kategori:

1. Menu navigasi admin yang berfungsi mengarahkan *admin* ke halaman beranda, *profil admin*, kelola *item* digital, kelola *member*, kelola peminjaman, laporan, pengujian dan keluar dari *website*.
2. *Combo box* “jenis pengujian” yang terdiri dari pengujian MAE dan RMSE.
3. Tombol “hitung” yang berfungsi melakukan proses pengujian sistem rekomendasi.
4. Tabel hasil pengujian yang terdiri dari tanggal pengujian, jenis pengujian, dan hasil pengujian.

#### Tampilan Mobile

1. Halaman Masuk

Halaman masuk merupakan halaman pertama yang muncul saat *user* membuka aplikasi *mobile*. Pada halaman ini *user* dapat masuk ke dalam sistem dengan mengisi *Username* dan *password* yang telah terdaftar.



Gambar 3.23 Rancangan Halaman Masuk *Mobile*

Keterangan:

1. Tampilan logo aplikasi
2. Teks *input* yang terdiri dari *Username* dan *password*.
3. Teks *link* “lupa password” yang berfungsi mengarahkan *user* ke halaman lupa *password*.
4. Tombol “daftar” yang berfungsi mengarahkan *user* ke halaman pendaftaran.
5. Tombol “masuk” yang berfungsi untuk mengirim data yang telah di isi oleh *user* dan mengarahkan *user* ke halaman beranda *member*.
6. Halaman Pendaftaran

Halaman pendaftaran merupakan halaman yang muncul saat *user* mengklik tombol daftar pada tampilan masuk. Pada halaman ini *user* dapat mendaftar sebagai *member* dengan mengisi *form* pendaftaran dan menekan tombol daftar.



Gambar 3.24 Rancangan Halaman Pendaftaran *Mobile*

Keterangan:

1. Teks *input* yang terdiri dari nama depan, nama belakang, *Username*, jenis kelamin, tempat lahir, tanggal lahir, agama, hobi, negara, provinsi, kota, alamat, *email*, nomor telepon, password, konfirmasi password, *upload* foto profil dan upload foto KTP.
2. Tombol “daftar” yang berfungsi untuk mengirim data diri yang telah di isi oleh *user*.
3. Teks *link* “*login* disini” yang berfungsi mengarahkan *user* ke halaman *login*.
4. Halaman Beranda

Halaman beranda *member* merupakan halaman yang muncul sesudah *member* telah masuk ke dalam sistem. Pada halaman ini *member* akan mendapatkan rekomendasi *item* digital.

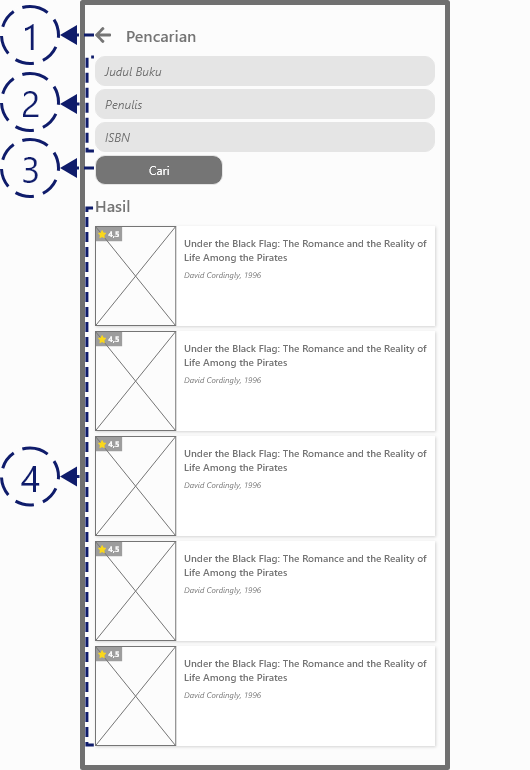


Gambar 3.25 Rancangan Halaman Beranda *Mobile*

Keterangan:

1. Teks *input* untuk pencarian *item* digital.
2. *Slider item* digital berdasarkan *rating* tertinggi.
3. Tombol “pratinjau” yang berfungsi mengarahkan *member* ke halaman detail *item* digital.
4. *Pagination bullet* sebagai petunjuk *item* yang sedang tampil pada *slider.*
5. *Item* digital yang baru ditambahkan, yang terdiri daftar *item* digital dan informasi ringkas dari *item* digital terbaru.
6. Rekomendasi *item* digital, yang terdiri dari daftar *item* digital yang direkomendasikan berdasarkan algoritma *deep collaborative filtering*.
7. *Icon* menu “beranda” yang berfungsi mengarahkan *member* ke halaman beranda.
8. *Icon* menu “kategori *item* digital” yang berfungsi mengarahkan *member* ke halaman kategori *item* digital.
9. *Icon* menu “*item* digital saya” yang berfungsi mengarahkan *member* ke halaman *item* digital yang telah terpinjam.
10. *Icon* menu “pemberitahuan” yang berfungsi mengarahkan *member* ke halaman pemberitahuan
11. *Icon* menu “profil” yang berfungsi mengarahkan *member* ke halaman profil.
12. Halaman Pencarian

Halaman pencarian merupakan halaman yang digunakan *member* untuk melakukan pencarian terhadap *item* digital, berdasarkan judul *item* digital, penulis *item* digital atau nomor ISBN *item* digital.



Gambar 3.26 Rancangan Halaman Pencarian *Mobile*

Keterangan:

1. *Icon* “*left arrow*” yang berfungsi mengarahkan *member* ke halaman sebelumnya.
2. Teks *input* yang terdiri dari judul *item* digital, penulis, dan ISBN *item* digital yang dicari.
3. Tombol “cari” yang berfungsi menampilkan daftar *item* digital sesuai dengan judul *item* digital, penulis, atau ISBN yang di input oleh *user*.
4. Daftar *item* digital yang ditampilkan sesuai dengan judul *item* digital, penulis, atau ISBN yang di input oleh *user*.
5. Halaman Detail *Item* Digital

Halaman detail *item* digital merupakan halaman yang digunakan *member* untuk melihat informasi detail dari suatu *item* digital. Pada halaman ini juga *member* dapat memberikan *rating* dan ulasan terhadap suatu *item* digital



Gambar 3.27 Rancangan Halaman Detail *Item* Digital *Mobile*

Keterangan:

1. *Icon* “*left arrow*” yang berfungsi mengarahkan *member* ke halaman sebelumnya.
2. Informasi detail dari *item* digital yang terdiri dari foto sampul, judul, *rating*, penulis, penerbit, kategori, tahun terbit, ISBN, dan ringkasan.
3. Tombol “pinjam” yang berfungsi untuk melakukan peminjaman *item* digital
4. Tombol “bintang” untuk memberikan *rating* terhadap *item* digital.
5. Teks *input* “ulasan” untuk memberikan komentar terhadap *item* digital.
6. Tombol “kirim *feedback*” yang berfungsi untuk mengirim *rating* dan ulasan yang diberikan oleh *member*.
7. Halaman Baca *Item* Digital

Halaman baca *item* digital merupakan halaman yang digunakan *member* untuk membaca *item* digital yang telah di pinjam.



Gambar 3.28 Rancangan Halaman Baca *Item* Digital *Mobile*

Keterangan:

1. *Icon* “*left arrow*” yang berfungsi mengarahkan *member* ke halaman sebelumnya.
2. Judul *item* digital yang sedang dibaca
3. Navigasi yang berfungsi mengarahkan *member* ke halaman sebelum atau selanjutnya dari *item* digital yang dibaca.
4. Halaman Kategori *Item* Digital

Halaman kategori *item* digital merupakan halaman yang digunakan *member* untuk melihat daftar *item* digital sesuai dengan kategori *item* digital.



Gambar 3.29 Rancangan Halaman Kategori *Item* Digital *Mobile*

Keterangan:

1. Teks *input* untuk pencarian *item* digital.
2. Combo box “kategori” yang terdiri dari semua kategori *item* digital.
3. Daftar *item* digital yang ditampilkan sesuai dengan kategori yang terpilih.
4. Daftar menu yang terdiri dari beranda, kategori *item* digital, *item* digital saya, pemberitahuan, dan *profil*.
5. Halaman *Item* Digital Saya

Halaman *item* digital saya merupakan halaman yang digunakan *member* untuk melihat daftar *item* digital yang telah dipinjam.



Gambar 3.30 Rancangan Halaman *Item* Digital Saya *Mobile*

Keterangan:

1. Teks *input* untuk pencarian *item* digital.
2. Daftar *item* digital yang sedang dipinjam oleh *member*.
3. Daftar menu yang terdiri dari beranda, kategori *item* digital, *item* digital saya, pemberitahuan, dan *profil*.
4. Halaman Pemberitahuan

Halaman pemberitahuan merupakan halaman yang digunakan *member* untuk melihat daftar pemberitahuan.



Gambar 3.31 Rancangan Halaman Pemberitahuan *Mobile*

Keterangan:

1. Teks *input* untuk pencarian *item* digital
2. Daftar pemberitahuan yang di dapat oleh *member*.
3. Daftar menu yang terdiri dari beranda, kategori *item* digital, *item* digital saya, pemberitahuan, dan *profil*.
4. Halaman Menu *Profil*

Halaman menu *profil* merupakan halaman yang digunakan *member* untuk melihat *profil member* dan menu lainnya.



Gambar 3.32 Rancangan Halaman Menu *Profil* *Mobile*

Keterangan:

1. Teks *input* untuk pencarian *item* digital.
2. Informasi *member* yang terdiri dari foto profil, nama, dan *Username*.
3. Teks *link* “lihat profil” yang berfungsi mengarahkan *member* ke halaman profil *member*.
4. Teks *link* “riwayat peminjaman” yang berfungsi mengarahkan *member* ke halaman riwayat peminjaman.
5. Teks *link* “tentang aplikasi” yang berfungsi mengarahkan *member* ke halaman tentang aplikasi.
6. Teks *link* “kontak” yang berfungsi mengarahkan *member* ke halaman kontak.
7. Teks *link* “keluar” yang berfungsi mengarahkan *member* keluar dari sistem dan menuju halaman login.
8. Daftar menu yang terdiri dari beranda, kategori *item* digital, *item* digital saya, pemberitahuan, dan *profil*.
9. Halaman Profil *Member*

Halaman *profil* merupakan halaman yang digunakan *member* untuk melihat informasi data diri *member* di dalam aplikasi.



Gambar 3.33 Rancangan Halaman *Profil Member* *Mobile*

Keterangan:

1. *Icon* “*left arrow*” yang berfungsi mengarahkan *member* ke halaman menu profil.
2. Tombol “*edit profil*” yang berfungsi mengarahkan *member* ke halaman *edit profil*.
3. Informasi data diri *member* yang terdiri dari foto profil, nama, *Username*, jenis kelamin, tempat lahir, tanggal lahir, agama, alamat, kota, provinsi, negara, *email*, dan foto KTP.
4. Halaman Riwayat Peminjaman

Halaman riwayat peminjaman merupakan halaman yang digunakan *member* untuk melihat daftar riwayat peminjaman *item* digital.



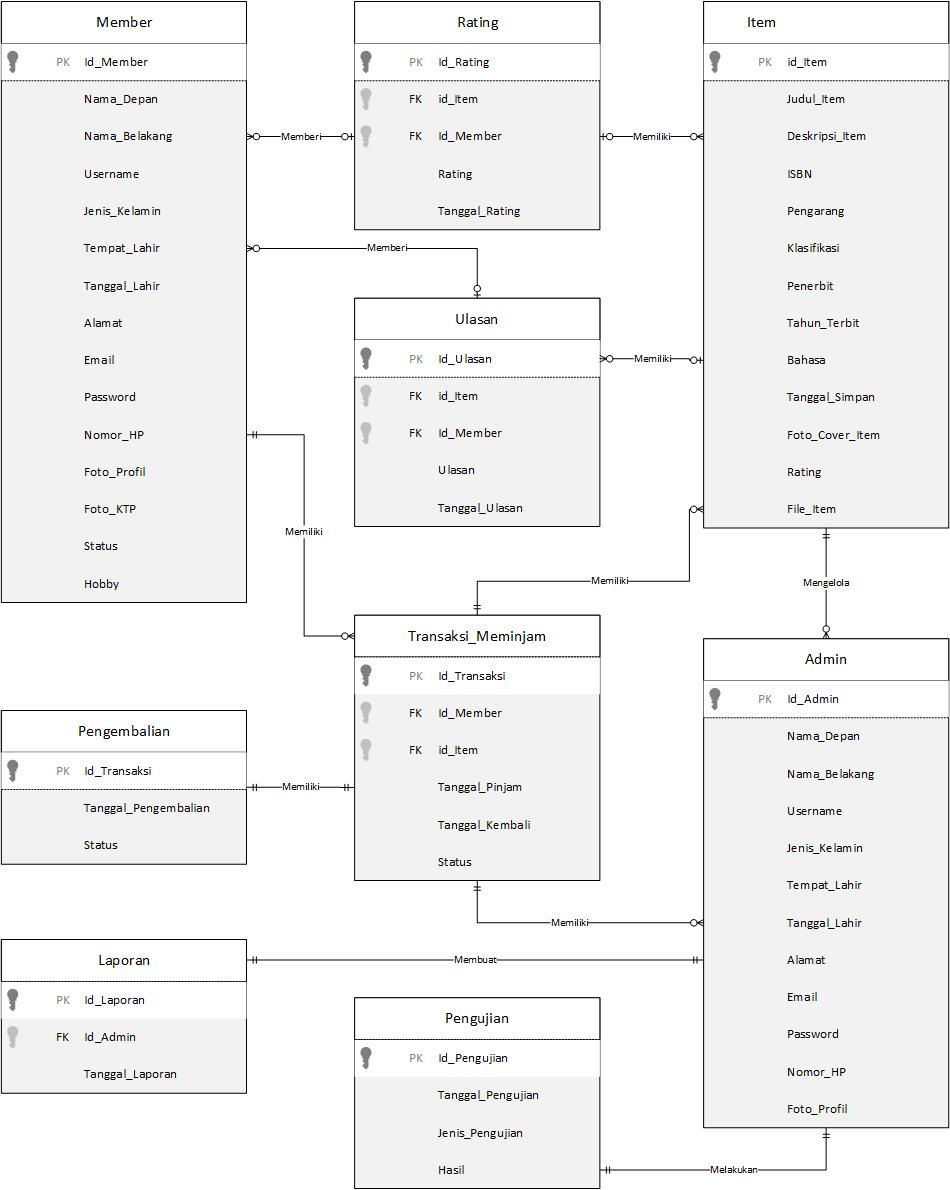
Gambar 3.34 Rancangan Halaman Riwayat Peminjaman *Mobile*

Keterangan:

1. *Icon* “*left arrow*” yang berfungsi mengarahkan *member* ke halaman menu profil.
2. Daftar riwayat *item* digital yang telah dipinjam oleh *member*.

### Perancangan Basis Data

Dalam perancangan basis data aplikasi rekomendasi perpustakaan digital diimplementasikan dalam bentuk *Class Diagram*. *Class Diagram* digunakan untuk menunjukkan hubungan antara entitas pada sebuah data dan bagaimana hubungan yang terjadi di antara objek-objek tersebut. Entitas menunjukkan objek-objek dasar yang terkait di dalam sistem yang terdiri dari: *Item*, *Rating*, Ulasan, *Member*, *Admin*, log\_peminjaman, Pengembalian, dan Laporan. Untuk relasi antara entitas dengan entitas lainnya pada sistem ini yaitu sebagai berikut.



Gambar 3.35 Relasi Antar Tabel Basis Data Perpustakaan Digital

Dari gambar 3.35 dapat dijelaskan dalam tabel 3.77 sampai tabel 3.85:

1. *Member*

Tabel *Member* digunakan untuk menyimpan data *user* terdaftar di aplikasi.

Tabel 3.77 Struktur Tabel *Member*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| ID\_*Member* | INT | ID\_*Member* [*primary key*, *auto increment*, *not null*] |
| Nama*\_Depan* | VARCHAR (100) | Nama depan dari *Member*  [*not null*] |
| Nama*\_Belakang* | VARCHAR (100) | Nama belakang dari *Member* |
| *Username* | VARCHAR(100) | *Username* dari *Member*  [*not null*] |
| Jenis\_Kelamin | ENUM(‘L’,’P’) | Jenis kelamin (L/P) |
| Tempat\_Lahir | VARCHAR(200) | Tempat Lahir *member* |
| Tanggal\_Lahir | DATETIME | Tanggal Lahir *member* |
| Alamat | VARCHAR(200) | Alamat dari *member* |
| *E-mail* | VARCHAR(50) |  |
| *Password* | VARCHAR(20) | Kata sandi yang digunakan *member* saat *login* |
| Nomor\_HP | VARCHAR(20) | Nomor HP *member* |
| Foto\_Profil | VARCHAR(100) | Foto profil *member* |
| Scan\_KTP | VARCHAR(100) | Scan KTP *member* |
| Status | ENUM(‘Terverifikasi’,’Tertunda’) |  |

1. *Admin*

Tabel *Admin* digunakan untuk menyimpan data *admin*.

Tabel 3.78 Struktur Tabel *Admin*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| ID\_*Admin* | INT | ID\_*Admin* [*primary key*, *auto increment*, *not null*] |
| Nama*\_Depan* | VARCHAR(100) | Nama depan dari *Admin*  [*not null*] |
| Nama*\_Belakang* | VARCHAR(100) | Nama belakang dari *Admin* |
| *Username* | VARCHAR(100) | *Username* dari *Member*  [*not null*] |
| Jenis\_Kelamin | ENUM(‘L’,’P’) | Jenis kelamin (L/P) |
| Tempat\_Lahir | VARCHAR(200) | Tempat Lahir *Admin* |
| Tanggal\_Lahir | VARCHAR(200) | Tanggal Lahir *Admin* |
| Alamat | VARCHAR(200) | Alamat dari *Admin* |
| *E-mail* | VARCHAR(50) |  |
| *Password* | VARCHAR(20) | Kata sandi yang digunakan *Admin* saat *login* |
| Nomor\_HP | VARCHAR(100) | Nomor HP *Admin* |
| Foto\_Profil | VARCHAR(100) | Foto profil *Admin* |

1. *Item*

Tabel *Item* digunakan untuk menyimpan data *item* digital.

Tabel 3.79 Struktur Tabel *Item*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_*Item* | INT | Id\_*Item* [*Primary Key*] |
| Judul\_*Item* | VARCHAR(100) | Nama dari *Item* |
| Deskripsi\_*Item* | VARCHAR(200) | Deskripsi Singkat dari *Item* |
| *File\_Item* | VARCHAR(100) | Nama *file item* digital |
| ISBN | VARCHAR(100) | Nomor Seri Standart Internasional |
| Pengarang | VARCHAR(200) | Nama dari pengarang / penulis *item* |
| Klasifikasi | VARCHAR(200) | Pengelompokan *item* digital |
| Penerbit | VARCHAR(200) | Nama penerbit *item* |
| Tahun\_Terbit | DATE | Tahun terbit *item* |
| Bahasa | VARCHAR(50) | Bahasa yang digunakan dalam *item* |
| Tanggal\_Simpan | DATETIME | Tanggal simpan *item* |
| Foto\_Cover\_*Item* | VARCHAR(200) | Menyimpan link foto cover *item* |
| *Rating* | INT | Penilaian *item* |

1. *Rating*

Tabel *Rating* digunakan untuk menyimpan data *rating*.

Tabel 3.80 Struktur Tabel *Rating*

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_*Rating* | INT | Id\_*Rating*[*Primary* Key] |
| Id\_*Item* | INT | Id\_*Item*[*Foreign* Key] |
| Id\_*Member* | INT | Id\_*Member*[*Foreign* Key] |
| *Rating* | INT | Penilaian *item* |
| Tanggal\_*Rating* | DATETIME | Tanggal *member*  memberi *rating item* |

1. Ulasan

Tabel Ulasan digunakan untuk menyimpan data ulasan.

Tabel 3.81 Struktur Tabel Ulasan

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_Ulasan | INT | Id\_Ulasan[*Primary* Key] |
| Id\_*Item* | INT | Id\_*Item*[*Foreign* Key] |
| Id\_*Member* | INT | Id\_*Member*[*Foreign* Key] |
| Ulasan | VARCHAR(300) | Ulasan *item* |
| Tanggal\_Ulas | DATETIME | Tanggal *member*  mengulas *item* |

1. Transaksi\_Meminjam

Tabel Log Peminjaman digunakan untuk menyimpan data transaksi meminjam yang dilakukan oleh *member*.

Tabel 3.82 Struktur Tabel Transaksi Meminjam

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_Transaksi | INT | Id\_Transaksi[*Primary* Key] |
| Id\_*Member* | INT | Id\_*Member*[*Foreign* Key] |
| Id\_*Item* | INT | Id\_*Item*[*Foreign* Key] |
| Tanggal\_Pinjam | DATETIME | Tanggal *member* melakukan peminjaman *item* |
| Tanggal\_Kembali | DATETIME | Tanggal *item* digitalharus dikembalikan |
| Status | ENUM(‘Pinjam’,’Kembali’) | Status Peminjaman *item* oleh *member* |

1. Pengembalian

Tabel Pengembalian digunakan untuk menyimpan data pengembalian *item*.

Tabel 3.83 Struktur Tabel Pengembalian

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_Transaksi | INT | Id\_Transaksi[*Primary* Key] |
| Tanggal\_Pengembalian | DATETIME | Tanggal telah dikembalikan |
| Status | ENUM(‘Kembali,’Perpanjang’) | Status peminjaman *item* digital oleh *member* |

1. Laporan

Tabel Laporan digunakan untuk menyimpan data Laporan Transaksi.

Tabel 3.84 Struktur Tabel Laporan

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_Laporan | INT | Id\_Laporan[*Primary* Key] |
| Id\_*Admin* | INT | Id\_Laporan[*Foreign* Key] |
| Tanggal\_Laporan | DATETIME | Tanggal laporan dibuat |

1. Pengujian

Tabel Pengujian digunakan untuk menyimpan data hasil Pengujian.

Tabel 3.85 Struktur Tabel Pengujian

|  |  |  |
| --- | --- | --- |
| Nama Atribut | Tipe Data | Keterangan |
| Id\_Pengujian | INT | Id\_Laporan[*Primary* Key] |
| Tanggal\_Pengujian | DATETIME | Tanggal Pengujian dibuat |
| Jenis\_Pengujian | VARCHAR(50) | Jenis Pengujian yang dipilih |
| Hasil | DOUBLE | Hasil Pengujian |

# DAFTAR PUSTAKA

Aithal, D. P. S., 2016. Smart Library Models For Future Generations. *International Journal Of Engineering Research And Modern Education (IJERME),* 1(1), p. 695.

Andini, T. D. & Zulkarnain, A., 2013. Suggestions Friends Engine Berbasis Hybrid Recommender System Untuk Mendapatkan Rekomendasi Teman Terbaik Pada Web Jejaring Sosial. *Jurnal Ilmiah Teknologi Informasi Asia,* 7(2).

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

Chen, S., Liu, X. & Li, B., 2018. A Cost-Sensitive Loss Function for Machine Learning. p. 1.

Elahi, M., Ricci, F. & Rubens, N., 2016. A Survey Of Active Learning In Collaborative Filtering Recommender Systems. *Computer Science Review,* Volume 20.

Fitri, A. A. D. & Syah, W. A., 2020. Pembelajaran Bahasa Arab Berbasis E-book. *Trends and Future Perspectives on Arabic Education, Linguistics, Leterature, Culture, and Translation,* p. 225.

Fraser, G. & Rojas, J. M., 2019. Software Testing. In: S. Cha, R. N. Taylor & K. C. Kang, eds. *Handbook of Software Engineering.* s.l.:Springer International Publishing.

Gajera, V., Shubham, Gupta, R. & Jana, P. K., 2016. An Effective Multi-Objective Task Scheduling Algorithm using Min-Max Normalization in Cloud Computing. p. 813.

Han, J. et al., 2019. Adaptive Deep Modeling of Users andItems Using Side Informationfor Recommendation. *IEEE Transactions On Neural Networks And Learning Systems.*

Henderi, Wahyuningsih, T. & Rahwanto, E., 2021. Comparison of Min-Max normalization and Z-Score Normalization in the K-nearest neighbor (kNN) Algorithm to Test the Accuracy of Types of Breast Cancer. *International Journal of Informatics and Information System,* Volume 4, p. 16.

Ikasari, D., Suhendra, A. & Farida, N., 2018. Metode Deep Learning Pada Sistem Rekomendasi : Review Paper. Volume 2, pp. 46-47.

Isinkaye, F., Folajimi, Y. & Ojokoh, B., 2015. Recommendation Systems: Principles, Methods And Evaluation. *Egyptian Informatics Journal,* p. 4.

Kapoor, N. & Tiwari, P., 2020. Library Collection Development In Digital Era. In: S. Dhar, U. Dhar & S. D. Mishra, eds. *Impact Of Research On Society: Evolving Perspectives.* New Delhi: Excellent Publishing House, pp. 150-151.

Khiste, G. P., Deshmukh, D. R. K. & Awate, A. P., 2018. Literature Audit of 'Digital Library': an Overview. *Vidyawarta,* p. 403.

Kumbhar, R., 2012. *Library Classification Trends in the 21st Century.* s.l.:Chandos Publishing.

Kwanggi, K., 2016. Book Review: Deep Learning. *Healthcare Informatics Research,* Volume 22, p. 1.

Li, L.-H., Hsu, R.-W. & Lee, F.-M., 2012. Review ofRecommender Systems and Their Applications. *Department of Information Management,* p. 13.

Li, Q. & Kim, B., 2003. An Approach For Combining Content-Based And Collaborative Filters. *J. Intell. Inf. Syst..*

Li, S., Hao, Z. & Ding, L., 2020. Research on the application of information technology of big data in chinese digital library. Volume 40, p. 525.

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

Liu, J. & Wu, C., 2017. Deep Learning Based Recommendation: A Survey. p. 452.

Li, X. & Orabona, F., 2019. On the Convergence of Stochastic Gradient Descent with Adaptive Stepsizes. p. 1.

Lucas, J. P. et al., 2013. A hybrid recommendation approach for a tourism system. *Department of Computing and Automatic,* pp. 1-4.

Lü, L. et al., 2012. Recommender Systems. *Physics Reports,* Volume 519.

Mallawa, S., 2012. Media Pustakawan. *Dewey Decimal Classification Edisi Ke-23 : Perubahan dan Perluasan Notasi Tentang Indonesia,* 19(3), p. 11.

Mohamed, A. E., 2017. Comparative Study of Four Supervised Machine Learning Techniques for Classification. Volume 7, p. 3.

Nahak, B. & Padhi, S., 2019. The Role Of Smart Library And Smart Librarian For E- Library Services. p. 89.

Nidhra, S. & Dondeti, J., 2012. Black Box And White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications,* Volume 2.

Nilashi, M. et al., 2013. Collaborative Filtering Recommender Systems. *Journal of Applied Sciences, Engineering and Technology,* p. 4173.

Rahmawati, S., Nurjanah, D. & Rismala, R., 2018. Analisis dan Implementasi Pendekatan Hybrid untuk Sistem Rekomendasi dengan Metode Knowledge Based Recommender System dan Collaborative Filtering. *Indonesian Journal Of Computing,* 3(2), pp. 12-13.

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

Ricci, F., Shapira, B. & Rokach, L., 2015. *Recommender Systems Handbook, Second Edition.* 2 ed. New York: Springer US.

Saeed, M. & Mansoori, E. G., 2017. A Novel Fuzzy-Based Similarity Measure For Collaborative Filtering To Alleviate The Sparsity Problem. *School of Electrical and Computer Engineering,* Volume 14, p. 2.

Sammut, C. & Webb, G. I., 2011. Latent Factor Models and Matrix Factorizations. p. 1.

Satija, M., 2008. DESIDOC Journal of Library & Information Technology. *Universal Decimal Classification: Past and Present,* 28(6), pp. 3-5.

Singh, U., 2017. Digitization Of Library Resources And The Formation Of Digital Libraries: A Practical Approach. pp. 11-13.

Suresha, G. & Narayanaswamy, B., 2016. International Research: Journal of Library & Information Science. *The Scheme of Library Classifications: Concerning the Structural changes of 23rd Dewey Decimal Classification (DDC),* 6(4), pp. 645-646.

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,* Volume 1.

Wenige, L. & Ruhland, J., 2018. Retrieval By Recommendation: Using Lod Technologies To Improve Digital Library Search. *Int J Digit Libr,* p. 2.

Yang, B., Lei, Y., Liu, J. & Li, W., 2017. Social Collaborative Filtering by Trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 39(8), p. 1.

Zeng, Y., Bai, X., Xu, J. & He, C. G. H., 2016. The Influence Of E-book Format And Reading Device On Users’ Reading Experience A Case Study Of Graduate Students. *Publishing Research Quarterly,* 32(3), pp. 2-3.

Zhang, J.-D., Chow, C.-Y. & Xu, J., 2016. Enabling Kernel-Based Attribute-Aware Matrix Factorization For Rating Prediction. *IEEE Transactions on Knowledge and Data Engineering.*

Zhang, M. & Yang, Z., 2019. GACOforRec: Session-Based Graph Convolutional Neural Networks Recommendation Model. *IEEE Access,* Volume 7, p. 114077.

Zhou, W. et al., 2018. Deep Learning Modeling for Top-N Recommendation With Interests Exploring. Volume 6, p. 51442.