# BAB I

# PENDAHULUAN

## Latar Belakang

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

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

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

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

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

## Rumusan Masalah

Berikut rincian permasalahan yang telah dituliskan pada latar belakang:

1. Bagaimana penerapan dari sistem rekomendasi menggunakan metode *Deep Collaborative Filtering* pada perpustakaan digital?
2. Bagaimana metode *Deep Collaborative Filtering* mengatasi *sparsity* matriks *rating* pada perpustakaan digital?
3. Bagaimana *user* perpustakaan digital mendapatkan *item* yang sesuai dengan kriteria dan keinginannya?

## Tujuan

Tujuan tugas akhir ini yaitu sebagai berikut:

1. Menerapkan *Deep* *Collaborative Filtering* pada perpustakaan digital.
2. Mengatasi *sparsity* matriks *rating* pada perpustakaan digital.
3. Mengembangkan aplikasi 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 perpustakaan digital yang bersifat *easy to use*, sehingga mempermudah dalam melakukan pencarian dan menampilkan hasil sesuai keinginan *user*.
2. Aplikasi ini dapat membantu masyarakat untuk terus membaca buku dimana saja dan kapan saja.

## Batasan Masalah

Batasan masalah pada penelitian implementasi *Deep Collaborative Filtering* untuk menyelesaikan rekomendasi pada perpustakaan digital dalam mendapatkan *item* sesuai dengan kebutuhan dan keinginan user sesuai dengan identifikasi dan rumusan masalah. Pembatasan masalah yang digunakan pada penelitian ini adalah:

1. *Rating user* digunakan sebagai parameter dalam sistem rekomendasi *Deep Collaborative Filtering* dengan menggunakan *history* pencarian dan peminjaman yang dilakukan oleh *user* dengan asumsi ketika *user* telah meminjam *item* perpustakaan berarti *user* telah memberikan *rating* pada *item* tersebut.
2. Hanya menggunakan teknik *Deep Collaborative Filtering* karena pada sistem rekomendasi ini *item* yang pernah dicari atau dipinjam *user* akan menjadi patokan untuk pemberian rekomendasi.
3. *Mobile application* sebagai produk dari Tugas Akhir ini hanya dapat dipasang pada *smartphone* dengan *operating system* Android versi Lollipop 5.0 dan belum tersedia dalam sistem operasi IOS.
4. Jumlah aktor dalam sistem ini yaitu dari 3(tiga), terdiri dari admin, member dan *user* (aktor yang belum mendaftar ke dalam sistem) dengan masing-masing aktor yaitu sebagai berikut:

Admin:

1. Mengelola data item digital.
2. Mengelola pinjaman *item* digital.
3. Melakukan verifikasi member terdaftar.
4. Mengelola laporan.

Member:

1. Mendapat rekomendasi *item* digital dari algoritma *Deep Collaborative Filtering*.
2. Mencari, meminjam, membaca, mengembalikan dan memperpanjang *item* digital.
3. Memberi *rating* dan *review* *item* digital yang telah dipinjam.

*User*:

1. Mendapat rekomendasi *item* digital dari algoritma *Deep Collaborative Filtering*.
2. Mencari dan melihat rincian *item* digital.
3. Melakukan pendaftaran pada sistem.

## Metodologi Pengembangan Sistem

Metodologi yang digunakan pada pengembangan sistem ini yaitu dengan model *waterfall*. Metode *waterfall* merupakan metodologi pengembangan sistem yang bersifat sistematis sehingga dapat menghasilkan software yang baik dan penataan perpustakaan yang terurut dengan rapi karena tahap demi tahap yang dilalui harus menunggu selesainya tahap sebelumnya dan berjalan berurutan.

1. Analisa Proses

Pada tahapan ini perhitungan untuk algoritma *Deep Collaborative Filtering* yang akan digunakan dengan contoh kasus atau angka sederhana disajikan secara tahap demi tahap serta menggunakan *flowchart* untuk menganalisis proses algoritma yang digunakan*.*

1. Studi Literatur

Pada tahapan ini adalah mencari literatur yang berhubungan dengan topik yang diangkat. Tujuan dari tahapan ini adalah penulis memahami konsep, metode dan teknologi Native.

1. Data *Processing*

Pada tahap ini akan dilakukan pengolahan data dengan baik, dimana akan melakukan transformasi data ke suatu format yang prosesnya lebih mudah dan efektif sebagai kebutuhan pengguna. Tahapan *processing* ini terdiri dari pemilihan atribut, penggabungan data, *data cleaning*, dan standarisasi data.

1. Perancangan

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

1. Implementasi

Pada tahapan ini dilakukan penulisan kode program, untuk *website* dengan bahasa *Javascript* dan *Framework Laravel,* untuk aplikasi *mobile* menggunakan bahasa Kotlin dan dataset dengan [https://www.kaggle.com/](https://www.kaggle.com/jealousleopard/goodreadsbooks)

1. Pengujian

Pengujian software diperlukan untuk memastikan aplikasi yang dibangun dapat berjalan sesuai fungsionalitasnya.

1. Pengujian sistem rekomendasi *Deep Collaborative Filtering* untuk mengukur keakuratan nilai rekomendasi dimana nilai dari hasil pengujian pada sistem rekomendasi akan menjadi nilai parameter akurat atau tidaknya rekomendasi yang diberikan kepada *user*.
2. Pengujian fungsionalitas dari aplikasi perpustakaan, menggunakan metode *Black Box Testing* untuk menguji perangkat lunak apakah hasilnya sesuai dengan yang diharapkan.
3. Kesimpulan hasil pengujian

Penarikan Kesimpulan yang diambil berdasarkan hasil pengujian yang dilakukan dengan metode *Deep Collaborative Filtering.*

# BAB II

# TINJAUAN PUSTAKA

## Perpustakaan Digital

Perpustakaan Digital didefinisikan sebagai kumpulan informasi elektronik yang berisi repository objek digital yang besar dan beragam, dan dapat diakses oleh sejumlah besar pengguna yang tersebar secara geografis. Objek digital yang dimaksud seperti teks, gambar, peta, suara, video, katalog, dan jurnal ilmiah (Khiste, et al., 2018). Setiap *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)

Pada umumnya, Perpustakaan Digital melayani mesin, manajer informasi, dan pemakai informasi. Semuanya ini demi mendukung manajemen koleksi, menyimpan, pelayanan bantuan, dan penelusuran informasi. Perpustakaan Digital dengan konsep *resource* c*enter* atau teknologi penyimpanan elektronik dapat mengurangi ruang penyimpanan buku, jurnal, dan majalah yang masih berbasis kertas. Konsep r*esource* *center* juga mendukung fasilitas pencarian pada database sehingga memberikan kemudahan dalam pencarian (Aithal, 2016).

## Sistem Rekomendasi

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



Gambar 2. 1 Contoh *Matrix factorization-based*

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

### Input data dalam Sistem Rekomendasi

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

1. Content Based System

Dalam metode content-based, rekomendasi dibuat berdasarkan kemiripan atribut-atribut dari suatu *item* dengan preferensi pengguna tersebut (Candillier, et al., 2009) .Misalnya seseorang mempunyai kecenderungan suka dengan membaca buku algoritma, maka sistem akan merekomendasikan buku tentang pemrograman.

1. Collaborative Filtering System

*Collaborative filtering* menghasilkan rekomendasi *item* berdasarkan histori pengguna lain yang memiliki selera sama dengan pengguna tersebut (Candillier, et al., 2009). Sistem akan memprediksi ketertarikan pengguna tersebut terhadap *item* yang akan direkomendasikan berdasarkan data pengguna lain yang memiliki ketertarikan yang sama.

1. Demographic System

Jenis sistem rekomendasi ini akan merekomendasikan *item* berdasarkan demografi profil pengguna. Misalnya sistem akan merekomendasikan *item* berdasarkan negara/daerah asal dari pengguna, sehingga *item* yang direkomendasikan akan berbeda tergantung dari demografi pengguna yang dituju.

1. Knowledge-based System

Sistem akan merekomendasikan *item* berdasarkan ranah pengetahuan yang spesifik tentang bagaimana fitur *item* tertentu memenuhi kebutuhan dan preferensi pengguna, dan bagaimana *item* tersebut berguna bagi pengguna. Sebagai contoh sistem akan merekomendasikan komputer dengan spesifikasi tinggi bagi pengguna dengan preferensi gamer, dan akan merekomendasikan komputer dengan spesifikasi rendah bagi pengguna dengan kebutuhan biasa saja.

1. Community-based System

Jenis sistem rekomendasi ini memodelkan dan memperoleh informasi tentang hubungan sosial dari pengguna dan preferensi pengguna lain yang berhubungan dengan pengguna tersebut. Rekomendasi yang diberikan berdasarkan rating yang diberikan oleh pengguna lain tersebut.

1. Hybrid System

Jenis sistem rekomendasi hybrid merupakan gabungan dari metode-metode yang telah disebutkan. Sistem rekomendasi *hybrid* akan menggunakan lebih dari satu metode untuk mentolerir kekurangan dari metode lainnya.

Dari beberapa metode atau teknik yang digunakan dalam sistem rekomendasi. Setiap metode disesuaikan dengan permasalahan dalam menghasilkan sebuah informasi yang sesuai. Penggabungan algoritma *Collaborative Filtering* dengan *Deep Learning* dapat memberikan pendekatan yang efektif (Li, et al., 2015). Oleh karena itu, pada tugas akhir ini kami menggunakan metode *Collaborative Filtering System*.

## Collaborative Filtering

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

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

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

## Matrix Factorization

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

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

Table 2.1 - Contoh matriks *rating user* terhadap suatu *item*

Pada tabel 2.1 akan terdapat nilai *rating* yang kosong. Hal ini dikarenakan *user* tidak memberikan *rating* kepada *item* yang ada. Banyaknya *user* yang hanya menilai sejumlah *item* saja menyebabkan terjadinya *sparsity* matriks *rating* (Zhang & Chow, 2016). Oleh karena itu diperlukan suatu metode untuk mengatasi masalah tersebut yaitu *Matrix Factorization.*

*Matrix Factorization* adalah pendekatan *Collaborative Filtering* yang paling efektif. Ini memungkinkan kita untuk menemukan *latent factor* interaksi *user-item* dengan memfaktorkan matriks interaksi ke dalam *latent space* fitur *user-item*. Salah satu metode *Matrix Factorization* klasik adalah *Probabilistic Matrix Factorization* (PMF). Banyak algoritma yang telah dikembangkan untuk meningkatkan kinerja PMF, dengan memasukkan *side information* seperti hubungan sosial. Tetapi metode *Matrix Factorization* mengalami masalah *cold-start*, yaitu rekomendasi apa yang harus dibuat ketika *user* / *item* baru tiba di sistem. Masalah lain yang sering muncul di banyak aplikasi dunia nyata adalah ketersebaran data atau cakupan yang berkurang. Memasukkan *side information* telah menunjukkan kinerja yang menjanjikan dalam *Collaborative Filtering*. Tetapi akan bermasalah jika *side information* tidak lengkap. Oleh karena itu, fitur pembelajaran untuk *Matrix Factorization* sangat penting untuk diterapkan (Li, et al., 2015).

## Deep Collaborative Filtering

Algoritma *Deep Collaborative Filtering* adalah algoritma yang dibuat untuk mengatasi masalah *sparsity* matriks *rating* pada algoritma *Collaborative Filtering*. *Deep Collaborative Filtering* merupakan algoritma yang dikenal sebagai penggabungan *matrix factorization* pada *Based Collaborative Filtering* dengan *Deep Learning*. *Deep Collaborative Filtering* mengintegrasikan faktorisasi matriks dan pembelajaran *deep feature* yang memodelkan pemetaan antara faktor laten yang digunakan dalam *Collaborative Filtering* dan lapisan laten dalam *deep models* (Li, et al., 2015).

### Integrates matrix factorization and deep feature learning

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

Rumus mempelajari faktor laten dari rating dan side information

Dimana:

1. *R* € merupakan matriks *rating*
2. *U* € merupakan faktor laten *user*
3. *V* € merupakan faktor laten *item*
4. *X* € merupakan *side information* *user*
5. *Y* € merupakan *side information* *item*

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

## Penyelesaian sparsity matriks rating menggunakan Deep Collaborative Filtering

Sparsity matriks rating merupakan keadaan dimana ketika user baru mendaftar dan belum memberikan peringkat terhadap item, sehingga sistem tidak dapat merekomendasikan item kepada user (Mishra, et al., 2017). Algoritma *Deep Collaborative Filtering* dapat menyelesaikan permasalahan sparsity matrix dengan informasi samping yang diperoleh dari profil pengguna / item seperti demografi user ,genre buku, dll. *marginalized Denoising Auto-encoders* (mDA) merupakan bagian dari *Stacked Denoising Auto-encoder* (SDA) yang bertujuan untuk mengurangi waktu komputasi yang tinggi, Pada kasus ini mDA yang digunakan berbasis *Collaborative Filtering* (CF) Seperti gambar dibawah berikut (Li, et al., 2015).

****

*Gambar 2.6.1 Kerangka Deep Collaboratif Filtering*

Secara matematis kerangka *Deep Collaborative Filtering* dapat dirumuskan sebagai berikut:

Dimana

## Android

Android adalah sebuah sistem operasi untuk *smartphone* dan *tablet*. Sistem operasi dapat diilustrasikan sebagai 'jembatan' antara perantara (*device*) dan penggunanya, sehingga pengguna dapat berinteraksi dengan *device*-nya dan menjalankan aplikasi-aplikasi yang tersedia pada *device*. Di dunia personal komputer, sistem operasi yang banyak dipakai adalah *Windows*, *Mac*, dan *Linux* (Satyaputra, 2016)*.*

Memilih sistem android untuk perangkat *mobile* merupakan keputusan yang tepat karena dibawah Google, android pasti dikembangkan terus menerus dan juga menjadi *developer* yang bisa bekerja bebas membuat aplikasi berbasis android. Kita juga bisa menjual produk, jasa, sebagai solusi permasalahan untuk penggunaanya, android juga menyediakan arsitektur development yang banyak, kita tidak perlu repot menghafal komponen arsitektur dari android, cukup mengetahui apa yang tersedia dalam sistem yang ingin dibangun dalam aplikasi kita (Yudhanto, 2018)  
Berikut cara kerja dari sistem operasi android (Yudhanto, 2018)

1. Aplikasi: Aplikasi berada pada tingkat ini bersama dengan aplikasi sistem inti untuk email, perpesanan SMS, kalender, penjelajahan Internet, atau kontak.
2. *Java API Framework*: Semua fitur android tersedia untuk developer melalui antarmuka pemrograman aplikasi.
3. Sistem tampilan digunakan untuk membangun *UI* aplikasi, termasuk daftar, tombol, dan menu.
4. Pengelola referensi digunakan untuk mengakses sumber daya *non-code* seperti *string*, grafik, dan *file layout*.
5. Pengelola notifikasi digunakan untuk menampilkan peringatan khusus di bilah status.
6. Pengelola aktivitas yang mengelola daur hidup aplikasi.
7. Penyedia materi yang memungkinkan aplikasi untuk mengakses data dari aplikasi lain.
8. Semua *API* kerangka kerja yang digunakan aplikasi sistem android.
9. *Library* dan android *runtime*: Setiap aplikasi berjalan dalam prosesnya sendiri dan dengan *instance* *Android Runtime* sendiri, yang memungkinkan beberapa mesin sekaligus *virtual* pada perangkat bermemori rendah. android juga menyertakan rangkaian *library* waktu proses inti yang menyediakan sebagian besar fungsionalitas bahasa pemrograman *Java*, termasuk beberapa fitur bahasa *Java 8* yang digunakan *framework Java API*. Banyak layanan dan komponen sistem android inti dibangun dari kode asli yang memerlukan pustaka asli yang ditulis dalam *C* dan *C++.* Pustaka asli tersebut tersedia untuk aplikasi melalui kerangka kerja *Java API*.
10. *Hardware Abstraction Layer* (HAL): Lapisan ini menyediakan antarmuka standar yang menunjukkan kemampuan perangkat keras di perangkat ke kerangka kerja *Java API* yang lebih tinggi. *HAL* terdiri atas beberapa modul pustaka, masing-masing mengimplementasikan antarmuka untuk komponen perangkat keras tertentu, seperti modul kamera atau *bluetooth*.
11. *Kernel Linux*: Fondasi *platform* android adalah *kernel Linux*. Lapisan di atas mengandalkan *kernel Linux* untuk fungsionalitas pokok seperti *threading* dan manajemen memori tingkat rendah. Menggunakan *kernel Linux* memungkinkan android memanfaatkan fitur keamanan utama dan memungkinkan produsen perangkat mengembangkan *driver* perangkat keras untuk *kernel* yang cukup dikenal.

## Website

Website adalah sekumpulan data digital seperti teks, gambar, suara, video, yang diletakkan dalam web server yang umumnya diakses melalui internet (Steven Hendrawan, 2020) sebuah website terdiri dari beberapa halaman web yang saling berhubungan dengan web lainnya yang disebut hyperlink, sedangkan teks yang dijadikan media penghubung disebut hypertext. Domain merupakan nama unik yang dimiliki dalam sebuah institusi sehingga bisa diakses melalui internet, misalnya yahoo.com, google.com, friendster.com, dll. Istilah lain yang sehubungan dengan website adalah homepage. Homepage adalah halaman awal dari domain, misalnya, membuka website www.yahoo.com, halaman pertama yang muncul disebut dengan homepage, jika mengklik menu-menu yang ada dan meloncat ke lokasi lainnya disebut dengan web page, sedangkan keseluruhan isi atau konten domain disebut website.

Dalam pengelompokan jenis *web*, lebih diarahkan pada:

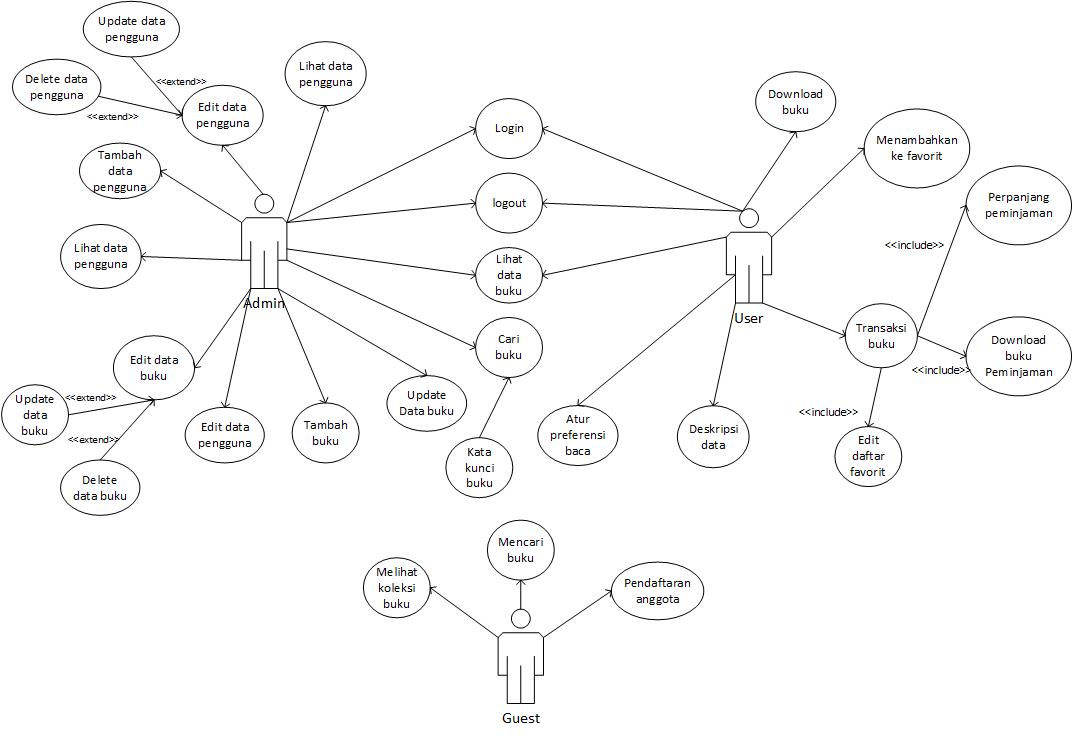
1. Jenis-jenis *web* berdasarkan sifatnya adalah:
2. *Website* dinamis, merupakan sebuah *website* yang menyediakan content atau isi yang selalu berubah-ubah setiap saat. Misalnya *website* berita, seperti, www.republika.co.id, [www.google.com](http://www.google.com).
3. *Website* statis, merupakan *website* yang kontennya sangat jarang diubah. Misalnya *web* profil organisasi, seperti, <http://kaskus.co.id/>
4. Ditinjau dari segi bahasa pemrograman, *website* terbagi atas:
5. Server side, merupakan *website* yang menggunakan bahasa pemrograman yang tergantung pada tersedianya *server*. Seperti, PHP, ASP dan sebagainya. Jika tidak ada server, *website* yang dibangun menggunakan bahasa pemrograman di atas tidak dapat berfungsi sebagaimana mestinya.
6. *Client side*, adalah *website* yang tidak membutuhkan *server* dalam menjalankannya, cukup diakses melalui browser saja. Misalnya, html.
7. Berdasarkan tujuannya*, website* dibagi atas:
8. *Personal web*, *website* yang berisi informasi pribadi seseorang.
9. *Corporate web*, *website* yang dimiliki oleh sebuah perusahaan.
10. *Portal web*, *website* yang mempunyai banyak layanan, mulai dari layanan berita, email, dan jasa-jasa lainnya.
11. *Forum web*, sebuah *web* yang bertujuan sebagai media diskusi

# BAB III

# ANALISIS DAN PERANCANGAN

3.1 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional dilakukan dengan menggunakan *use case diagram* untuk memperlihatkan interaksi antar user dan sistem. Berikut *use case diagram* dari aplikasi implementasi *Deep Collaborative Filtering* dalam merekomendasikan item perpustakaan digitalberbasis *mobile* dan *web*.



Gambar 3. 1 Use case diagram pada *mobile* dan *website*

Berikut adalah aktor pada *Use Case Diagram* aplikasi implementasi *Deep Collaborative Filtering*  dalammerekomendasikan untuk item berbasis *mobile* dan *web.*

|  |  |  |
| --- | --- | --- |
| No. | Aktor | Deskripsi |
| 1 | *Admin* | Pengguna sistem yang bertugas dan memiliki hak akses untuk melakukan operasi pengelolaan terhadap user dan guest yang ingin memakai sistem dan juga memiliki hak akses terhadap *user.* |
| 2 | *User* | Pengguna sistem yang telah mendaftar pada perusahaan yang memakai sistem dan telah ter-verifikasi dalam perusahaan yang sudah terdaftar dalam sistem. |
| 3 | *Guest* | Pengguna sistem yang tidak terdaftar pada sistem yang memiliki akses melihat lokasi *user* yang ditujukan kepada guest. |

Berikut adalah narasi *Use Case Diagram* pada *website*:

Tabel 3. 1 Narasi *Use Case* *Admin* sistem melihat daftar perusahan

Tabel 3. 8 Narasi *Use Case* *admin* sistem melihat *user* dari sistem.

|  |  |  |
| --- | --- | --- |
| Nama *use case* | Melihat Daftar *User* | |
| Aktor | *Admin* | |
| Deskripsi | *Use case* ini menjelaskan proses *admin* untuk melihat daftar para *user* dari perusahaan tertentu yang terdaftar di sistem. | |
|  | Aksi Aktor | Respon Sistem |
| *Normal Flow Event* | 1. Pergi ke halaman *Home*. |  |
|  | 1. Menampilkan halaman *Home* 2. Menampilkan *list* data perusahaan. |
| 1. Mengetuk atau mengklik pada *card* perusahaan yang ingin di lihat para *user* nya. |  |
|  |  |
|  |  |
|  | 1. Melakukan pengecekan dari data perusahaan yang diklik. 2. Menampilkan list data para *user*. |
| *Alternative Flow Event* |  | 1. Menampilkan pesan informasi tidak ada perusahaan. 2. Menampilkan pesan informasi tidak ada *user*. |

# DAFTAR PUSTAKA

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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