



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

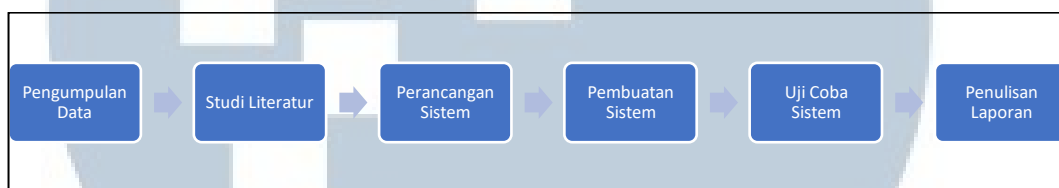
This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam rancang bangun sistem rekomendasi lagu menggunakan metode *user-item based collaborative filtering* adalah sebagai berikut.



Gambar 3.1 Metodologi Penelitian

1. Pengumpulan Data

Pengumpulan data dilakukan agar mendapatkan data yang akan digunakan dalam penelitian. Penelitian ini membutuhkan data lagu yang diambil dari situs [spotify.com](https://open.spotify.com) melalui API yang disediakan oleh Spotify.

2. Studi Literatur

Studi literatur dilakukan sebagai pembelajaran terhadap berbagai teori yang memiliki relevansi terhadap perancangan dan pembuatan sistem. Teori yang dimaksud antara lain Sistem Rekomendasi, *Collaborative Filtering*, *User-Item Based Collaborative Filtering*, *Mean Absolute Error (MAE)*, REST API, dan API Spotify.

3. Perancangan Sistem

Tahap awal perancangan sistem dimulai dengan perancangan model sistem untuk memperlihatkan gambaran umum sistem. Lalu dilanjutkan dengan membuat *Data Flow Diagram (DFD)* untuk menggambarkan alur perpindahan data yang

terjadi pada sistem yang dibuat. Kemudian dibuat *sitemap* untuk menggambarkan struktur halaman aplikasi. Kemudian dibuat *flowchart* yang menggambarkan alur kerja dari sistem ini. Kemudian dibuat *entity relationship diagram* untuk menggambarkan hubungan antara entitas. Selanjutnya, dibuat *database schema* untuk menggambarkan hubungan antar tabel. Kemudian dibuat struktur tabel dan rancangan antarmuka.

4. Pembuatan Sistem

Pembuatan sistem rekomendasi lagu menggunakan metode *user-item based collaborative filtering* ini berbasis Android dengan menggunakan bahasa pemrograman Java dan menggunakan API yang tersedia pada [spotify.com](https://open.spotify.com/).

5. Uji Coba Sistem

Proses uji coba sistem diperlukan agar dapat menguji sistem yang telah dibuat serta memperbaiki jika ada *error/bug* di dalamnya. Uji coba dilakukan dengan cara membandingkan nilai prediksi *rating item* dengan *rating* yang sebenarnya dari *user* dan diuji tingkat keakuratannya dengan pengukuran *error* menggunakan *Mean Absolute Error* (MAE) terhadap dua jenis daftar rekomendasi yang diberikan, yaitu berdasarkan *genre* yang dipilih *user* dan berdasarkan keseluruhan *user* yang menggunakan aplikasi. Semakin kecil nilai MAE maka semakin akurat sistem dalam memberikan rekomendasi.

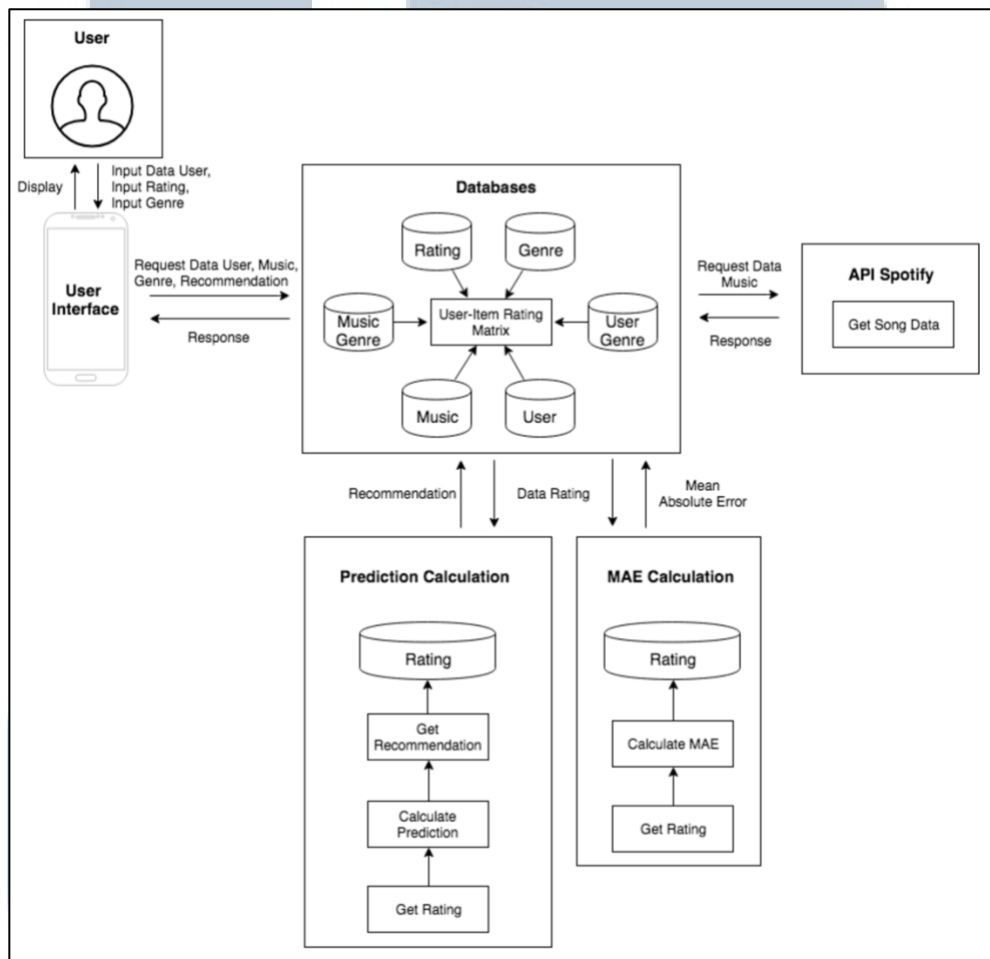
6. Penulisan Laporan

Penulisan laporan mencakup seluruh penelitian dari awal pengumpulan data hingga hasil, kesimpulan, dan saran dari penelitian yang telah dilakukan.

3.2 Perancangan Sistem

Sistem dirancang dengan menggunakan model sistem, *data flow diagram*, *sitemap*, *flowchart*, *Entity Relationship Diagram*, *database schema*, struktur tabel dan desain antarmuka.

3.2.1 Model Sistem



Gambar 3.2 Model Sistem Rekomendasi Lagu

Gambar 3.2 merupakan model Sistem Rekomendasi Lagu. Komponen pada Gambar 3.2 dapat dijabarkan sebagai berikut.

1. User

User dapat melihat lagu dari sistem dan memberikan data berupa *rating* dan *genre*.

2. User Interface

User interface bertujuan untuk menampilkan tampilan antarmuka sistem rekomendasi lagu. Pada halaman ini, *user* dapat melihat lagu-lagu yang terdapat pada sistem.

3. Database

Database bertujuan untuk menyimpan data lagu, *rating*, *genre*, *user genre*, *music genre*, dan *user*. Kemudian akan dibuat *user-item rating matrix* untuk perhitungan rekomendasi.

4. Prediction Calculation

Prediction calculation bertujuan untuk memperhitungkan prediksi rekomendasi lagu dengan menggunakan algoritma MVA. Hasil dari nilai prediksi diurutkan dari yang nilainya tertinggi hingga terendah kemudian ditampilkan ke *user*. Hasil dari nilai prediksi akan disimpan di dalam *database*.

5. MAE Calculation

MAE Calculation berfungsi untuk menghitung tingkat keakuratan dari sistem rekomendasi dengan membandingkan nilai prediksi *rating* dengan nilai *rating* yang diberikan oleh *user*. Semakin kecil nilai MAE maka semakin akurat sistem dalam memberikan rekomendasi.

6. API Spotify

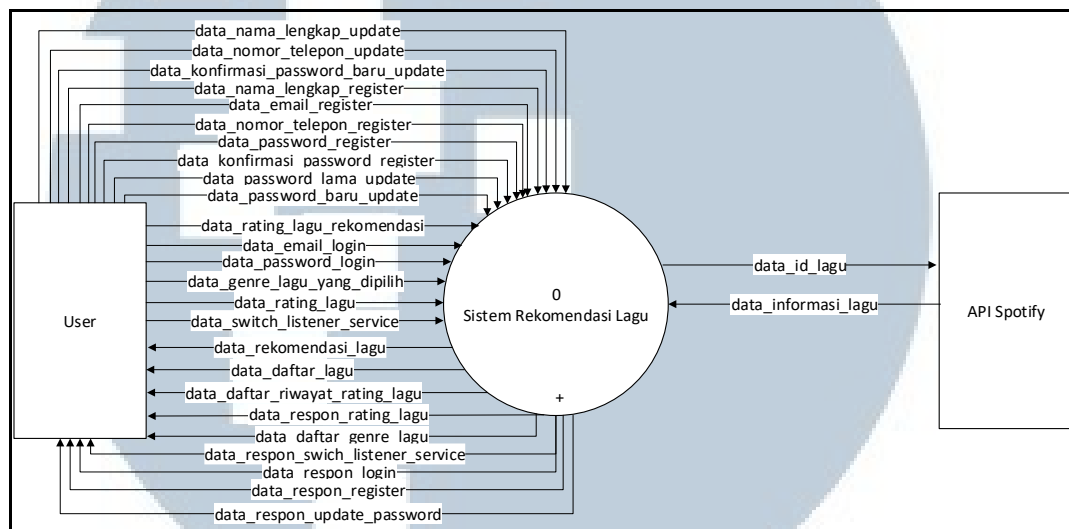
API Spotify digunakan untuk mengambil data informasi lagu.

3.2.2 Data Flow Diagram

Berikut merupakan *Data Flow Diagram* dari aplikasi sistem rekomendasi.

A. Data Flow Diagram Level 0

Pada bagian ini, digambarkan alur perpindahan data yang terjadi pada aplikasi sistem rekomendasi lagu yang dibuat. Berikut adalah *Data Flow Diagram* dari aplikasi sistem rekomendasi lagu.

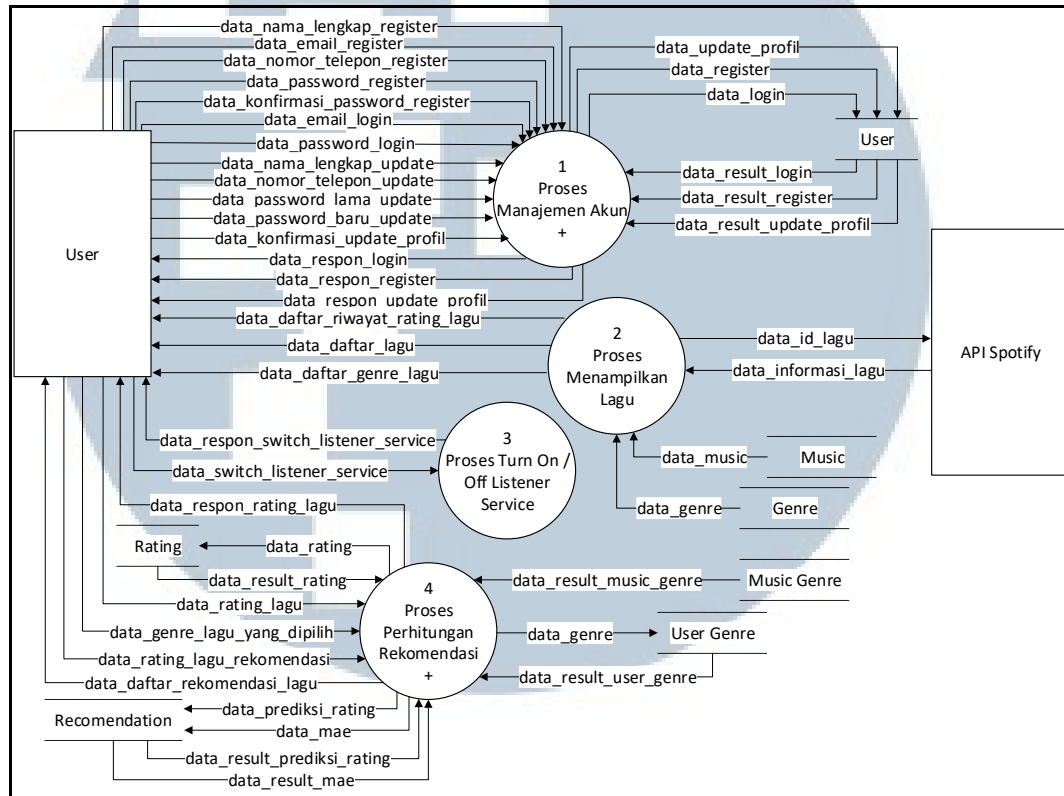


Gambar 3.3 Data Flow Diagram Level 0

Gambar 3.3 menunjukkan *Data Flow Diagram* Level 0 dari aplikasi sistem rekomendasi lagu. Terdapat dua buah entitas, yaitu *user* dan API Spotify dan satu sistem utama yaitu Sistem Rekomendasi Lagu. Dapat dilihat bahwa *user* dapat mengirimkan sejumlah data ke sistem, yang terdiri dari *data_nama_lengkap_register*, *data_email_register*, *data_nomor_telepon_register*, *data_password_register*, *data_password_lama_update*, *data_email_login*, *data_nomor_telepon_update*, *data_rating_lagu*, *data_konfirmasi_update_profil*, *data_konfirmasi_password_register*, *data_genre_lagu_yang_dipilih*, *data_password_login*, *data_nama_lengkap_update*, *data_password_baru_update*, *data_rating_lagu_rekomendasi*, dan *data_switch_listener_service*. Di samping itu, sistem dapat mengirimkan *data_id_lagu* ke API Spotify. API Spotify juga dapat mengirimkan data ke sistem, yaitu *data_informasi_lagu*. Setiap aliran data yang ada

pada *Data Flow Diagram Level 0* memiliki lebih dari satu bagian dan akan dijelaskan pada *Data Flow Diagram Level 1*.

B. Data Flow Diagram Level 1



Gambar 3.4 Data Flow Diagram Level 1

Gambar 3.4 menunjukkan *Data Flow Diagram Level 1* dari aplikasi sistem rekomendasi lagu dengan membagi data ke dalam proses-proses yang lebih spesifik, seperti Proses Manajemen Akun, Proses Menampilkan Lagu, Proses Turn On/Off Listener Service, dan Proses Perhitungan Rekomendasi. *User* dapat mengirimkan sejumlah data ke Proses Manajemen Akun, yang terdiri dari `data_nama_lengkap_register`, `data_email_register`, `data_nomor_telepon_register`, `data_password_register`, `data_konfirmasi_password_register`, `data_email_login`, `data_password_login`, `data_nama_lengkap_update`, `data_password_lama_update`, `data_nomor_telepon_update`, `data_password_baru_update` dan

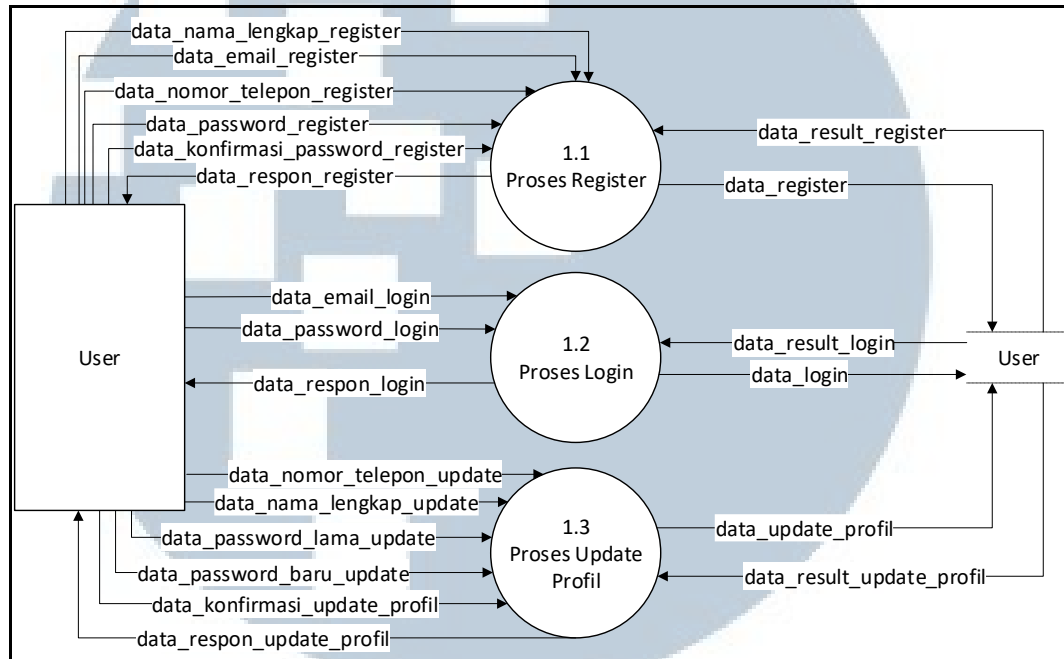
data_konfirmasi_update_profil. Proses Manajemen Akun dapat mengirimkan sejumlah data ke tabel User, yang terdiri dari data_update_profil, data_register, dan data_login. Proses Manajemen Akun mendapatkan respon dari tabel User yang terdiri dari data_result_update_profil, data_result_register, dan data_result_login. Kemudian Proses Manajemen Akun dapat mengirimkan sejumlah data ke *user*, yang terdiri dari data_respon_register, data_respon_login, dan data_respon_update_profil.

Pada Proses Menampilkan Lagu terdapat dua entitas, yaitu *user* dan API Spotify. Proses Menampilkan Lagu dapat mengirimkan data data_id_lagu ke API Spotify dan menerima data_informasi_lagu dari API Spotify. Proses Menampilkan Lagu juga dapat menerima data data_music dari tabel Music dan data data_genre dari tabel Genre. *User* dapat mengirimkan data data_switch_listener_service ke Proses Turn On/Off Listener Service dan mendapat respon dari Proses yaitu data_respon_switch_listener_service.

Pada Proses Perhitungan Rekomendasi, *user* dapat mengirimkan sejumlah data ke Proses Perhitungan Rekomendasi, yang terdiri dari data_rating_lagu, data_genre_lagu_yang_dipilih, dan data_rating_lagu_rekomendasi. Proses Perhitungan Rekomendasi dapat mengirimkan data data_genre ke tabel User Genre dan mendapatkan respon data_result_user_genre, data_prediksi_rating dan data_mae ke tabel Rating dan mendapatkan respon data_result_mae, data_result_prediksi_rating, data_rating ke tabel Rating dan mendapatkan respon data_result_rating. Proses Perhitungan Rekomendasi juga dapat menerima data data_result_music_genre dari tabel Music Genre. *User* juga dapat menerima sejumlah data dari proses yang terdiri dari data_respon_rating_lagu, dan

data_daftar_rekomendasi_lagu. Detail dari Proses Manajemen Akun dan Proses Perhitungan Rekomendasi akan dijabarkan pada *Data Flow Diagram Level 2*.

C. Data Flow Diagram Level 2 Proses Manajemen Akun



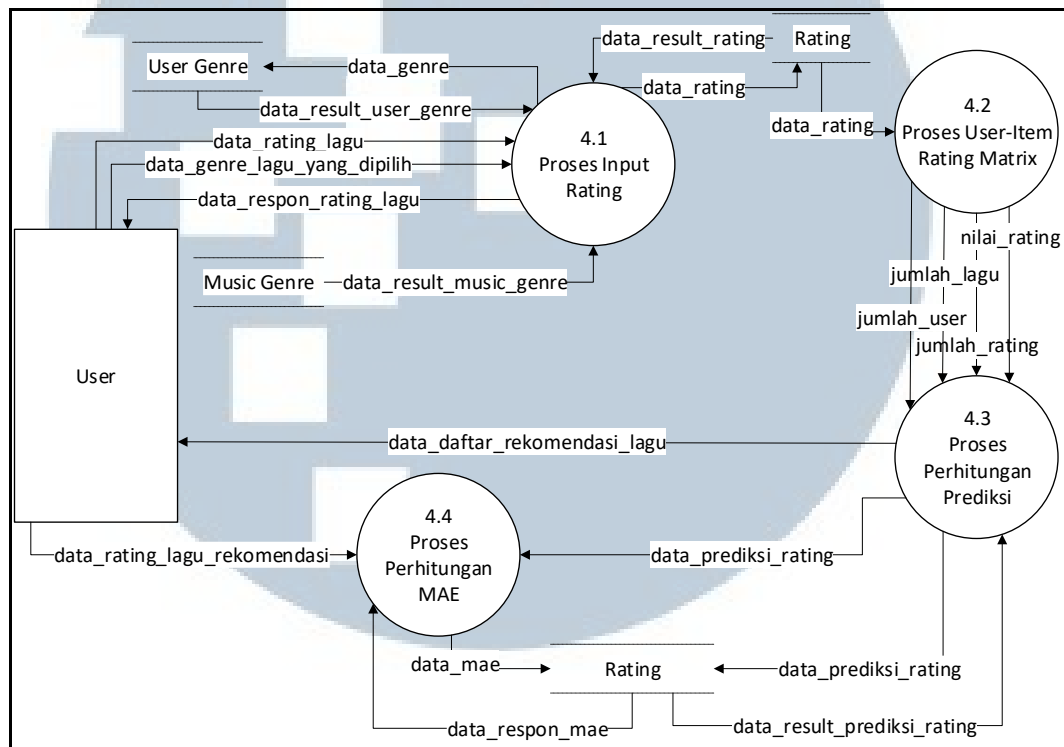
Gambar 3.5 Data Flow Diagram Level 2 Proses Manajemen Akun

Gambar 3.5 menunjukkan *Data Flow Diagram Level 2* Proses Manajemen Akun. Terdapat tiga proses yaitu Proses Register, Proses Login, dan Proses Update Profil. Pada Proses Register, *user* perlu memasukkan data berupa nama lengkap, *email*, nomor telepon, *password*, dan konfirmasi *password*. Pada Proses Login, *user* perlu memasukkan data berupa *email* dan *password*. Pada Proses Update Profil, *user* dapat memasukkan *password* lama, nama lengkap, nomor telepon, *password* baru, dan konfirmasi *password* baru.

D. Data Flow Diagram Level 2 Proses Perhitungan Rekomendasi

Gambar 3.6 menunjukkan *Data Flow Diagram Level 2* Proses Perhitungan Rekomendasi. Terdapat empat proses yang terdiri dari Proses Input Rating, Proses User-Item Rating Matrix, Proses Perhitungan Prediksi, dan Proses Perhitungan

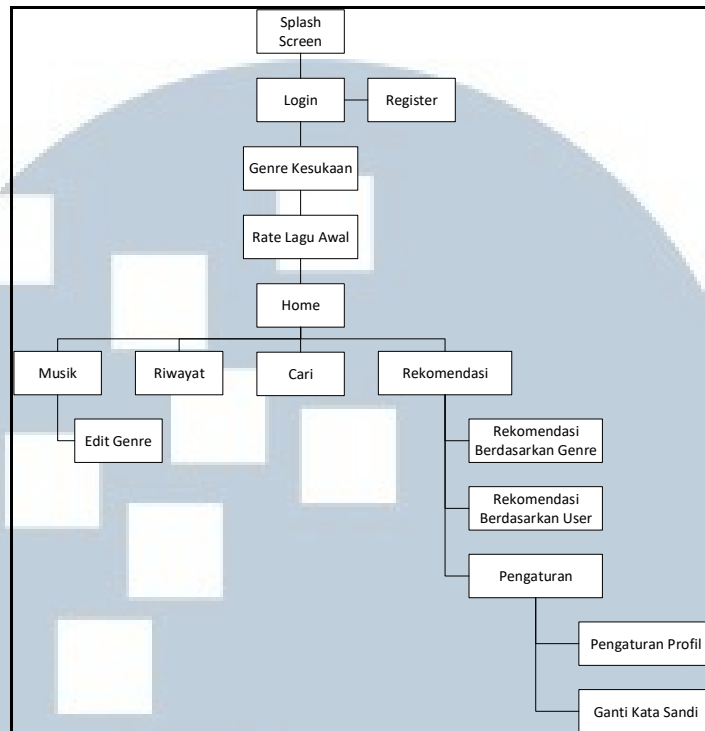
MAE. Untuk mendapatkan rekomendasi lagu, *user* perlu memasukkan data berupa minimal 3 *genre* dan *rating* terhadap lagu sebanyak minimal 10 lagu. Untuk mengetahui MAE dari sistem rekomendasi, *user* perlu melakukan *rating* terhadap lagu yang direkomendasikan oleh sistem.



Gambar 3.6 Data Flow Diagram Level 2 Proses Perhitungan Rekomendasi

3.2.3 Sitemap

Gambar 3.7 menunjukkan *sitemap* sistem rekomendasi lagu. Terdapat halaman Splash Screen, Login, Register, Genre Kesukaan, Rate Lagu Awal, dan Home yang terdiri dari Musik, Riwayat, Cari, dan Rekomendasi. Pada halaman Musik terdapat halaman Edit Genre untuk mengubah *genre* kesukaan, pada halaman Rekomendasi terdapat Rekomendasi Berdasarkan Genre, Rekomendasi Berdasarkan User, dan Pengaturan yang terdiri dari Pengaturan Profil dan Ganti Kata Sandi.



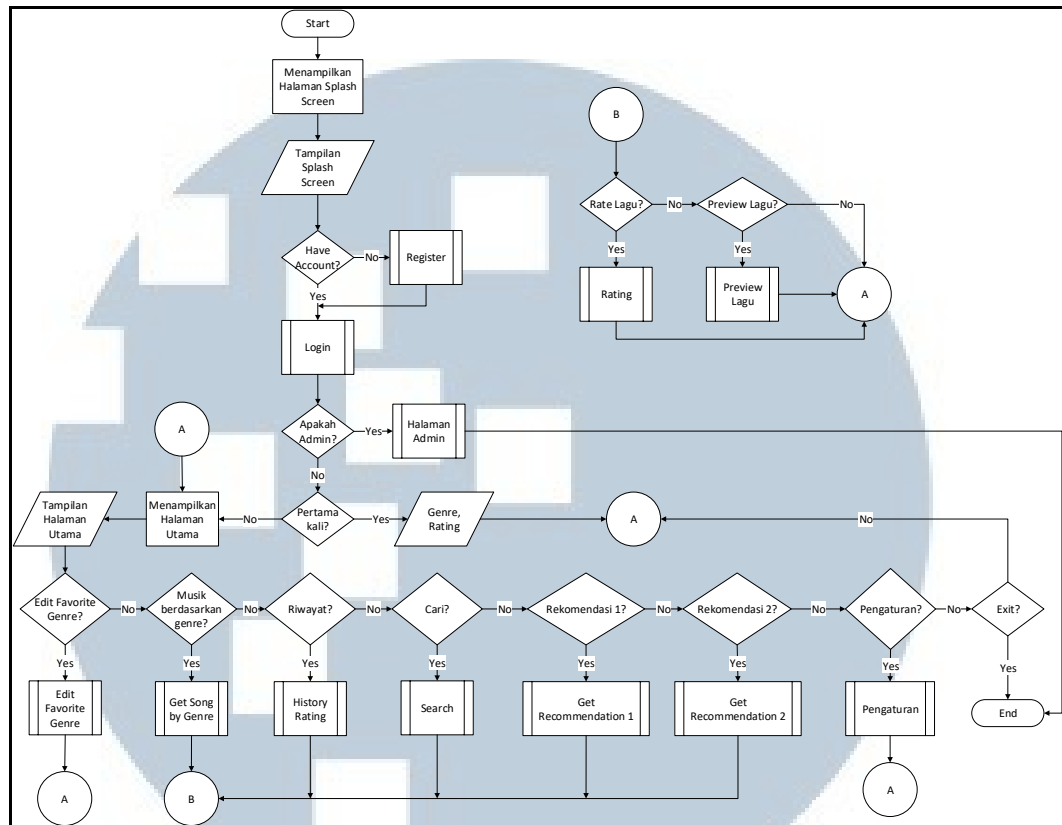
Gambar 3.7 Sitemap Sistem Rekomendasi Lagu

3.2.4 Flowchart

Berikut merupakan *flowchart* dari aplikasi sistem rekomendasi.

A. Flowchart Sistem

Gambar 3.8 adalah *flowchart* dari sistem rekomendasi ini. Awalnya akan ada *splash screen* dari logo aplikasi. Kemudian *user* akan diarahkan ke laman *login*. Apabila *user* telah memiliki akun, *user* dapat masuk ke sistem. Apabila belum, *user* dapat membuat akun terlebih dahulu. *User* yang login akan dicek terlebih dahulu, apakah *user* tersebut *admin* atau bukan, apabila *user* tersebut *admin* maka akan masuk ke halaman Admin. Apabila bukan *admin* maka akan lanjut sebagai *user* ke halaman utama, jika pertama kali menggunakan aplikasi, maka *user* diminta untuk memilih minimal tiga *genre* lagu, dan memberikan *rating* terhadap 10 lagu.



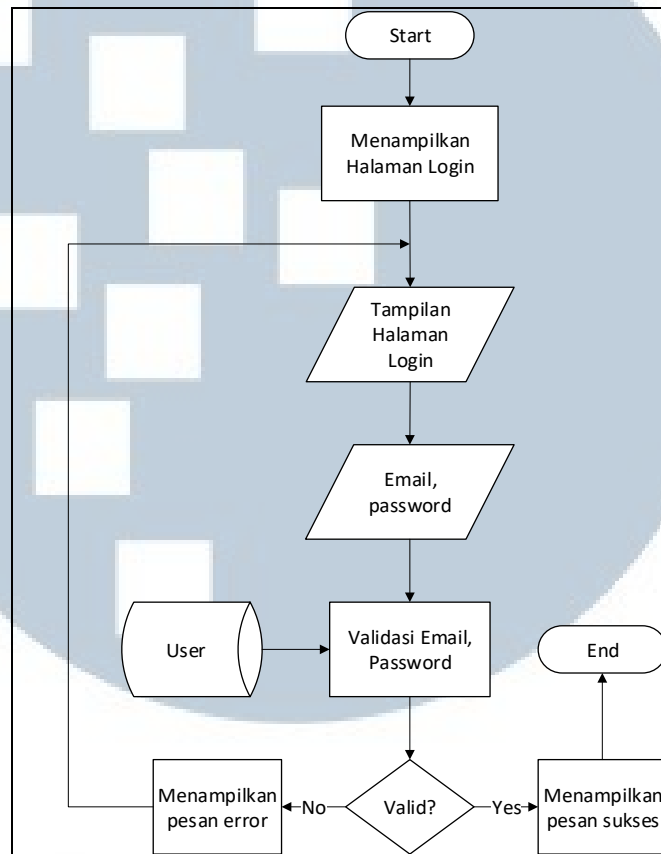
Gambar 3.8 Flowchart Sistem

Pada halaman utama terdapat beberapa halaman, yaitu Musik, Riwayat, Cari dan Rekomendasi. *User* dapat mengubah *genre* yang disukai pada halaman Edit Genre Favorit, *user* juga dapat melihat riwayat rating yang telah diberikan, serta melakukan pencarian terhadap lagu atau artis. *User* dapat memutar *preview* lagu yang dipilih dan memberikan penilaian berupa *rating*. Nilai *rating* yang diberikan *user* akan dimasukkan ke dalam *database* aplikasi. *Rating* ini berfungsi untuk menghitung *Mean Absolute Error*. Terdapat halaman Pengaturan bagi *user* untuk dapat melakukan perubahan nama lengkap, nomor telepon dan kata sandi.

B. Flowchart Login

Gambar 3.9 merupakan *flowchart* Login. Akan ditampilkan halaman Login berupa *form*. *User* dapat melakukan *login* dengan memasukkan *email* dan *password*. Setelah itu akan dilakukan pengecekan validasi *email* dan *password* ke

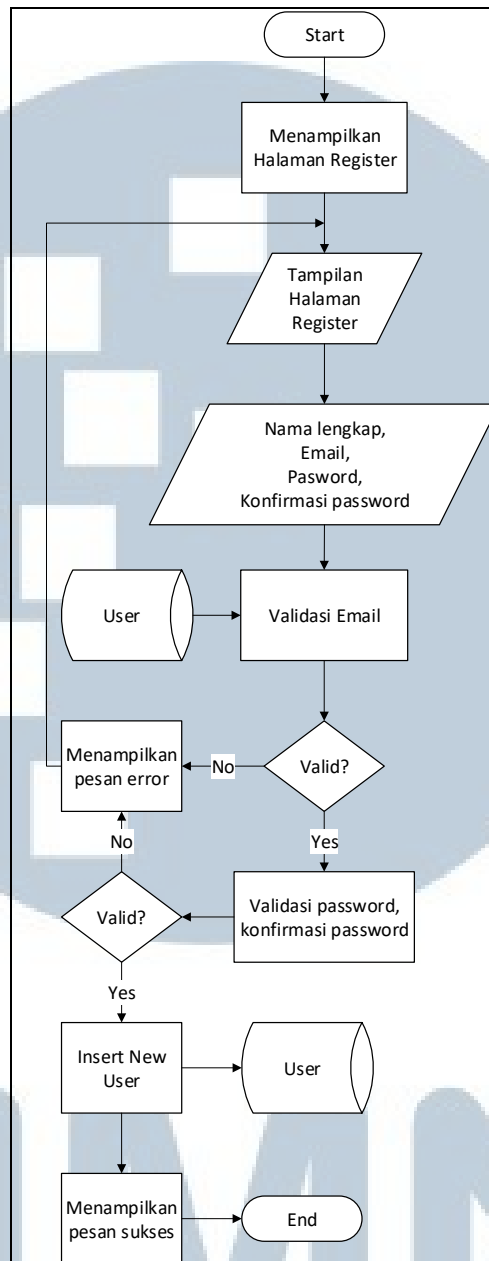
database. Apabila tidak valid, akan ditampilkan pesan *error* dan *user* dapat melakukan login kembali dengan memasukkan data yang benar. Jika data yang dimasukkan valid, akan ditampilkan pesan sukses dan berhasil masuk ke sistem.



Gambar 3.9 Flowchart Login

C. Flowchart Register

Gambar 3.10 merupakan *flowchart* Register. Akan ditampilkan halaman Register berupa *form*. *User* akan diminta untuk memasukkan data berupa nama lengkap, *email*, *password*, dan konfirmasi *password*. Setelah itu, akan ada validasi, yaitu pengecekan apakah *email* sudah ada di dalam *database* atau belum.



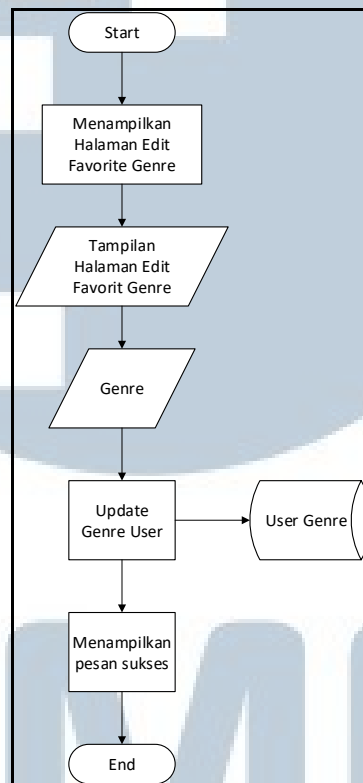
Gambar 3.10 Flowchart Register

Apabila tidak valid, maka akan muncul pesan *error* dan *user* dapat mendaftar ulang dengan memasukkan data yang benar. Apabila valid, akan dilakukan validasi password dengan konfirmasi password. Apabila tidak valid, akan ditampilkan pesan error, dan *user* dapat mendaftar ulang dengan memasukkan data yang benar. Apabila valid, data *user* akan dimasukkan ke dalam tabel User pada *database* dan

menampilkan pesan sukses. *User* yang telah berhasil melakukan registrasi dapat masuk ke sistem dengan akun yang telah dibuat.

D. Flowchart Edit Favorite Genre

Gambar 3.11 merupakan *flowchart* Edit Favorite Genre. Akan ditampilkan halaman Edit Favorit Genre, *user* akan diminta untuk memasukkan minimal 3 *genre* kesukaan yang akan disimpan.



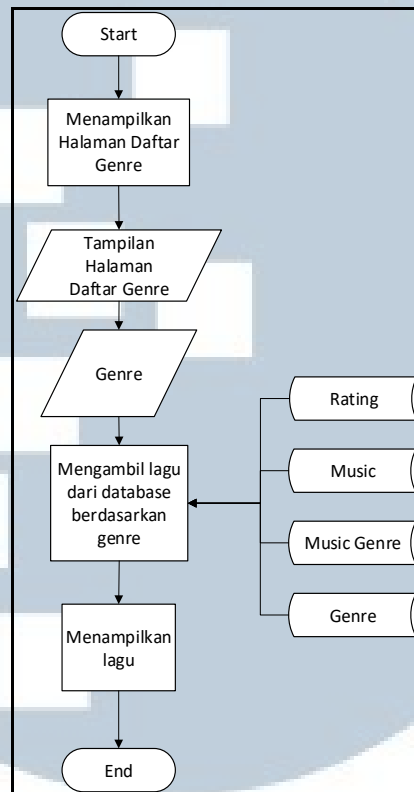
Gambar 3.11 Flowchart Edit Favorite Genre

Setelah itu data *genre* yang dipilih *user* akan dimasukkan ke dalam *database* pada tabel User Genre, dan ditampilkan pesan sukses.

E. Flowchart Get Song by Genre

Gambar 3.12 merupakan *flowchart* Get Song by Genre. Akan ditampilkan halaman yang berisi daftar *genre* musik. *User* dapat memilih *genre* dari daftar yang

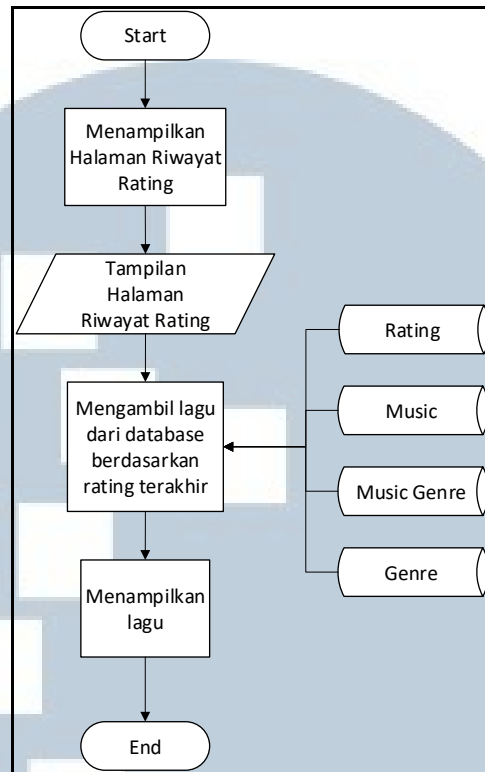
ditampilkan lalu sistem akan mengambil lagu dari database berdasarkan *genre* yang dipilih. Setelah mengambil data, maka akan ditampilkan daftar lagu.



Gambar 3.12 Flowchart Get Song by Genre

F. Flowchart History Rating

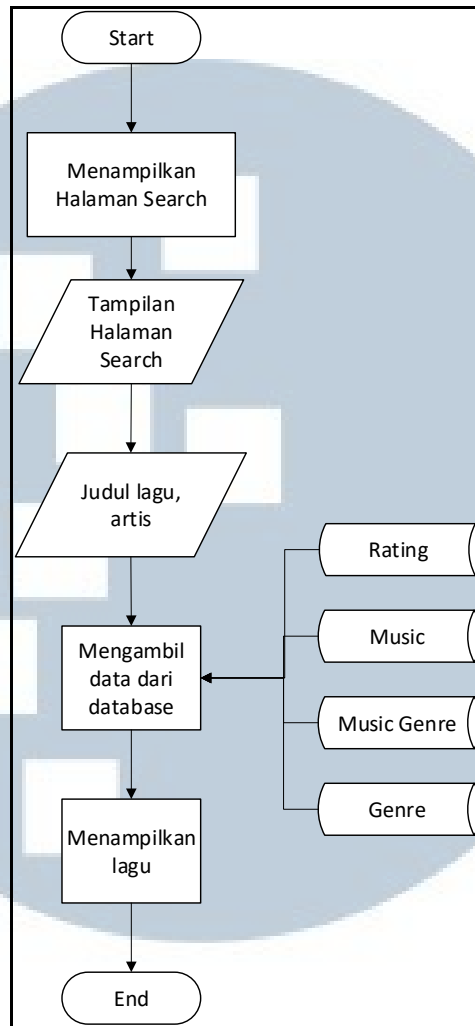
Gambar 3.13 merupakan *flowchart* History Rating. Akan ditampilkan halaman Riwayat Rating yang telah diberikan *user*. Sistem akan mengambil data lagu dan diurutkan berdasarkan waktu *user* memberikan *rating* dari yang terbaru hingga terlama. Setelah mengambil data, akan ditampilkan daftar lagu.



Gambar 3.13 Flowchart History Rating

G. Flowchart Search

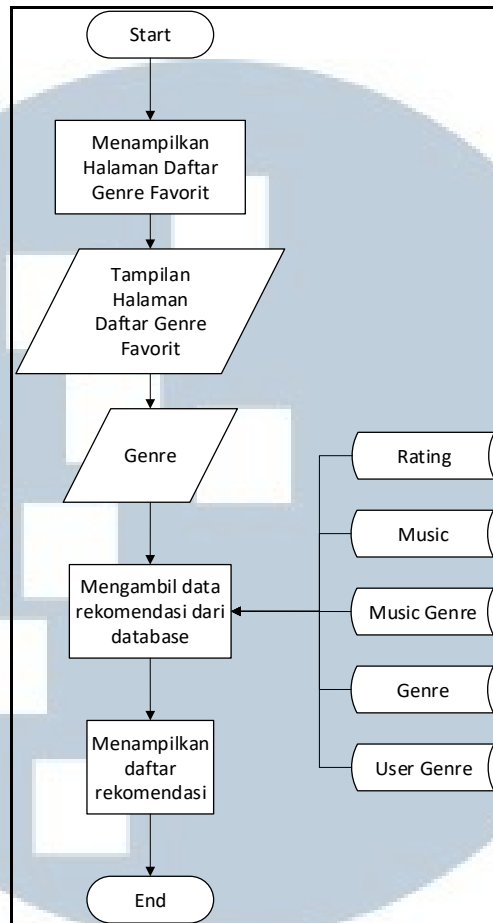
Gambar 3.14 merupakan *flowchart* Search. User dapat melakukan pencarian musik pada aplikasi dengan memasukkan data input berupa nama artis atau judul lagu. Sistem kemudian mengambil data musik dari database, lalu ditampilkan.



Gambar 3.14 Flowchart Search

F. Flowchart Get Recommendation 1

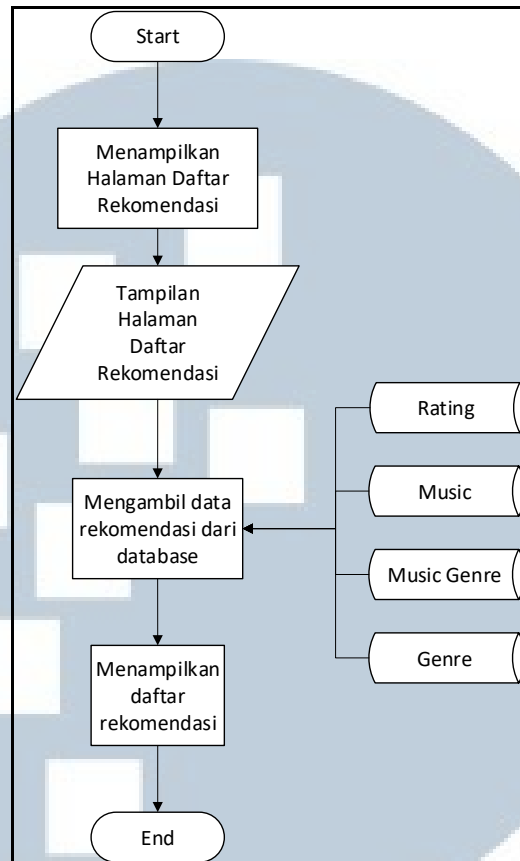
Gambar 3.15 merupakan *flowchart* Get Recommendation 1. Akan ditampilkan halaman yang berisi daftar *genre* yang disukai *user*. *User* dapat melihat rekomendasi dengan memilih *genre* yang tersedia. Sistem akan mengambil data rekomendasi dari *database* dan diurutkan berdasarkan nilai prediksi yang tertinggi hingga terendah berdasarkan *genre* yang dipilih. Setelah mengambil data dari *database*, sistem akan menampilkan daftar rekomendasi untuk *user*.



Gambar 3.15 Flowchart Get Recommendation 1

G. Flowchart Get Recommendation 2

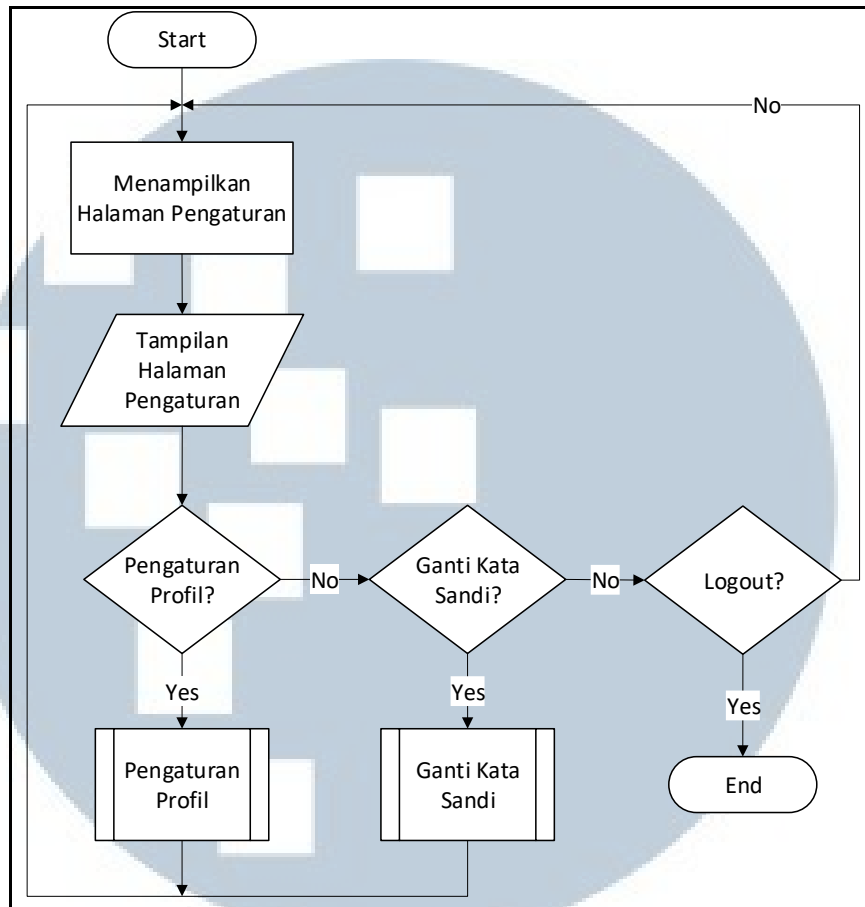
Gambar 3.16 merupakan *flowchart* Get Recommendation 2. Akan ditampilkan halaman yang berisi daftar rekomendasi untuk *user*. Sistem akan mengambil data rekomendasi dari *database* dan diurutkan berdasarkan nilai prediksi yang tertinggi hingga terendah. Setelah mengambil data dari *database*, sistem akan menampilkan daftar rekomendasi untuk *user*



Gambar 3.16 Flowchart Get Recommendation 2

H. Flowchart Pengaturan

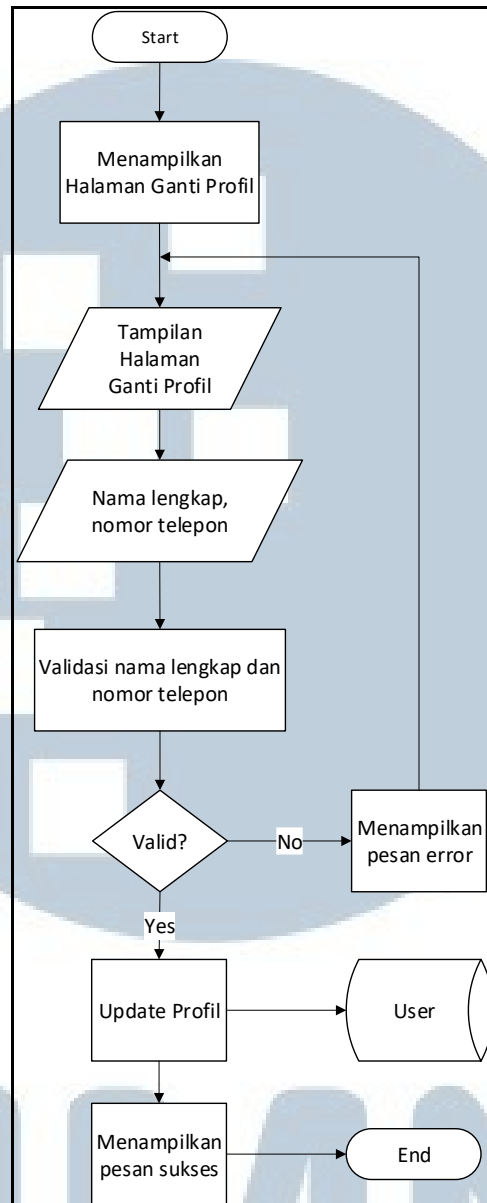
Gambar 3.17 merupakan *flowchart* Pengaturan. Akan ditampilkan halaman Pengaturan, pada halaman ini terdapat tiga pilihan, yaitu pengaturan profil, ganti kata sandi dan *logout*. Jika *user* memilih pengaturan profil, maka akan masuk ke halaman Pengaturan Profil, jika *user* memilih ganti kata sandi maka akan masuk ke halaman Ganti Kata Sandi, jika *user* memilih *logout* maka akan keluar dari sistem.



Gambar 3.17 Flowchart Pengaturan

I. Flowchart Pengaturan Profil

Gambar 3.18 merupakan *flowchart* Pengaturan Profil. Akan ditampilkan halaman untuk melakukan pergantian profil. *User* dapat mengganti nama lengkap dan nomor telepon akun dengan cara memasukkan data berupa nama lengkap dan nomor telepon baru. Setelah itu akan dilakukan validasi apakah data yang dimasukkan benar dan sesuai format. Jika tidak valid, maka akan ditampilkan pesan *error* dan *user* dapat melakukan pergantian profil kembali dengan memasukkan data dengan benar. Apabila valid, maka akan ditampilkan pesan sukses, profil berhasil diganti dan di-*update* ke *database*.

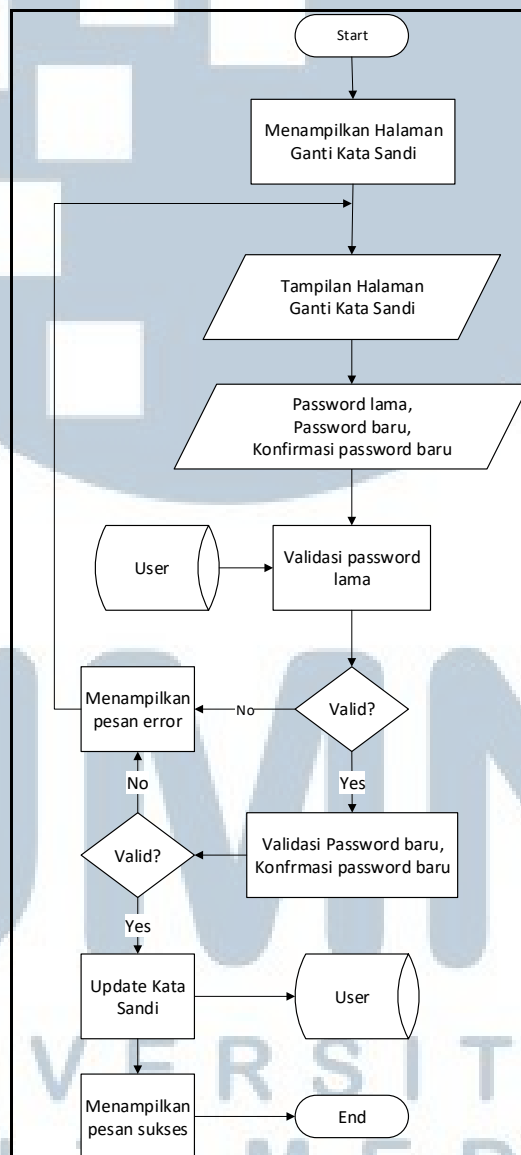


Gambar 3.18 Flowchart Pengaturan Profil

J. Flowchart Ganti Kata Sandi

Gambar 3.19 merupakan *flowchat* Ganti Kata Sandi. Akan ditampilkan halaman untuk melakukan pergantian *password*. *User* dapat mengganti *password* akun dengan cara memasukkan data berupa *password* lama, *password* baru, dan konfirmasi *password* baru. Setelah itu akan dilakukan validasi apakah *password* lama sama dengan *password* yang ada di dalam *database*. Jika tidak valid, maka akan ditampilkan pesan *error* dan *user* dapat melakukan pergantian *password*

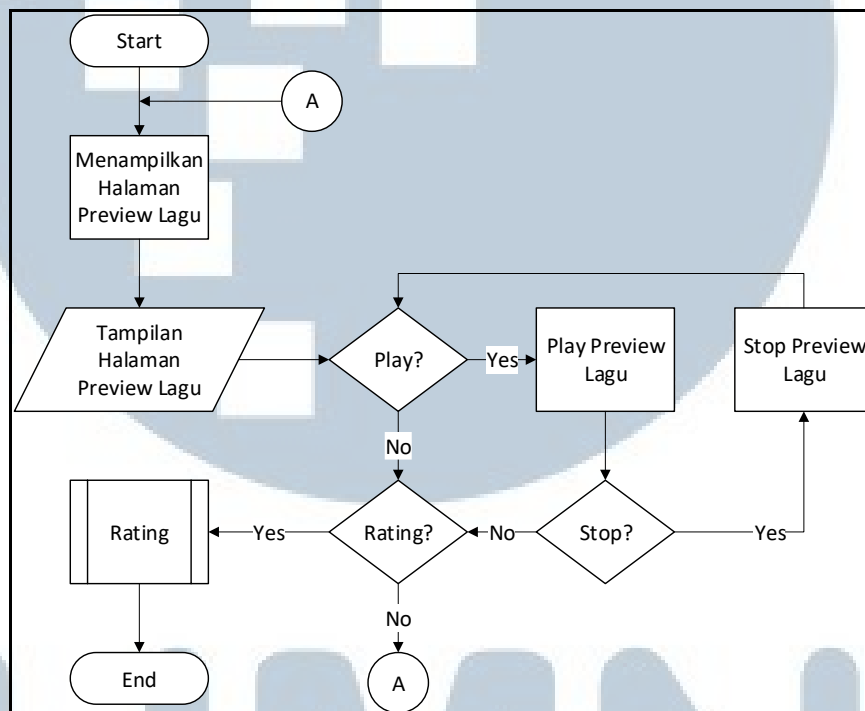
kembali dengan memasukkan data dengan benar. Apabila valid, maka akan dilakukan validasi apakah *password* baru sama dengan konfirmasi *password* baru. Jika tidak *valid*, maka akan ditampilkan pesan *error* dan *user* dapat melakukan pergantian *password* kembali dengan memasukkan data yang benar. Jika *valid*, akan ditampilkan pesan sukses, *password* berhasil diganti dan di-*update* ke *database*.



Gambar 3.19 Flowchart Ganti Kata Sandi

K. Flowchart Preview Lagu

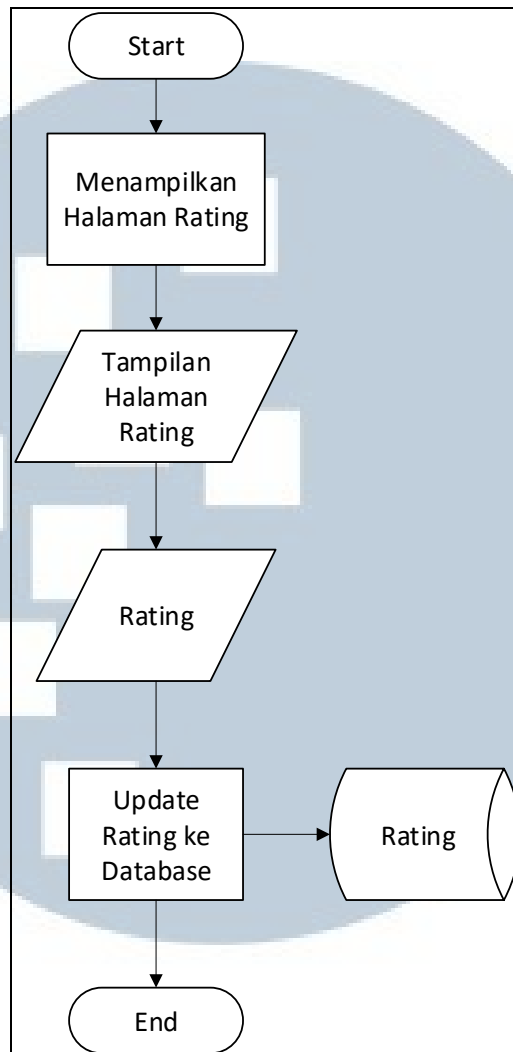
Gambar 3.20 merupakan *flowchart* Preview Lagu. Akan ditampilkan halaman Preview Lagu, pada halaman ini terdapat dua pilihan, yaitu *preview* lagu dan *rating*. Apabila *user* menekan tombol *play* maka *preview* lagu akan dijalankan, apabila menekan tombol *stop* maka *preview* lagu akan dimatikan. Apabila *user* menekan *rating* maka akan ditampilkan *dialog rating*.



Gambar 3.20 Flowchart Preview Lagu

L. Flowchart Rating

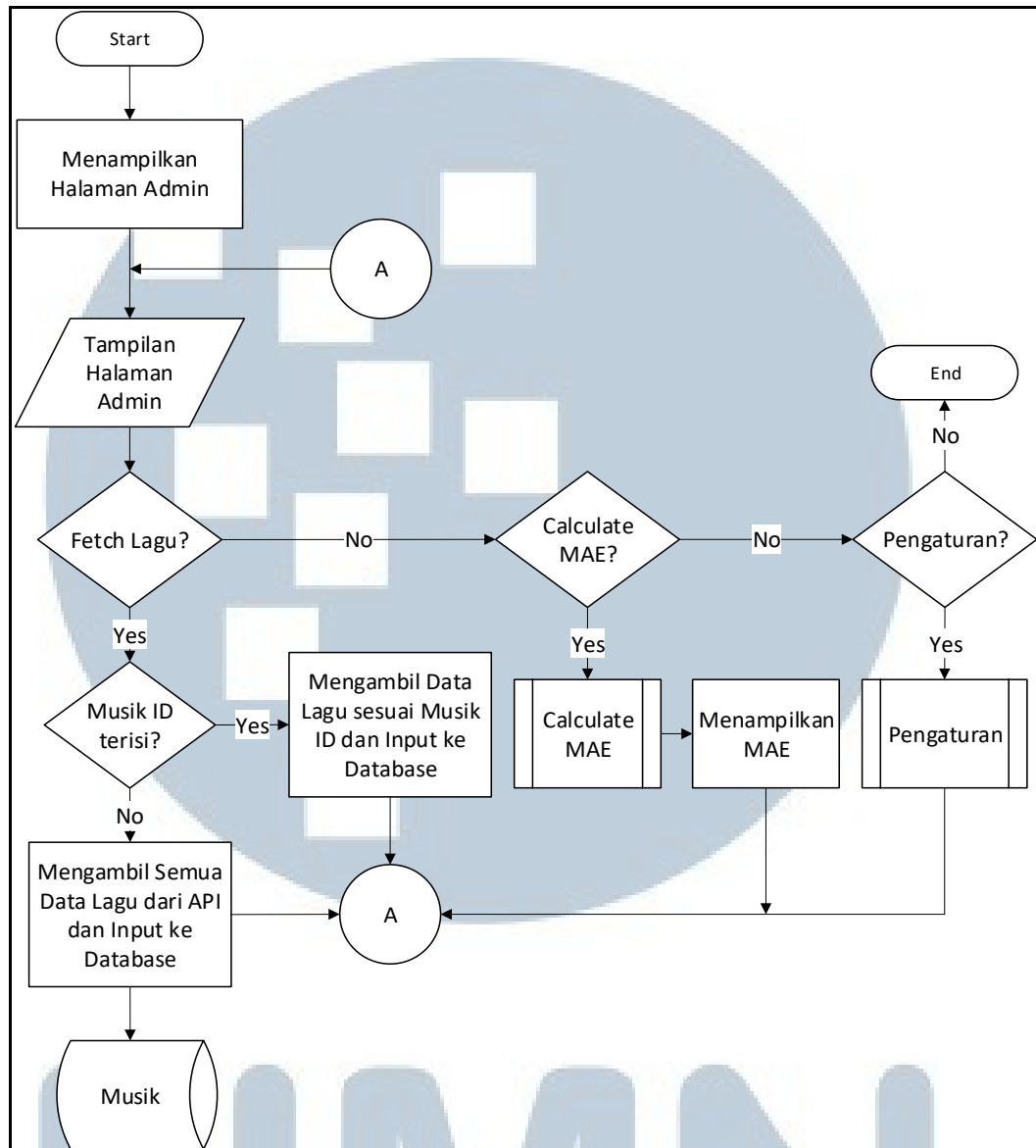
Gambar 3.21 merupakan *flowchart* Rating. *User* diminta untuk memberikan *rating* terhadap lagu yang dipilih dengan memilih skala 1 – 5. Data *rating* yang diberikan *user* akan dimasukkan ke dalam *database* pada tabel Rating.



Gambar 3.21 Flowchart Rating

L. Flowchart Halaman Admin

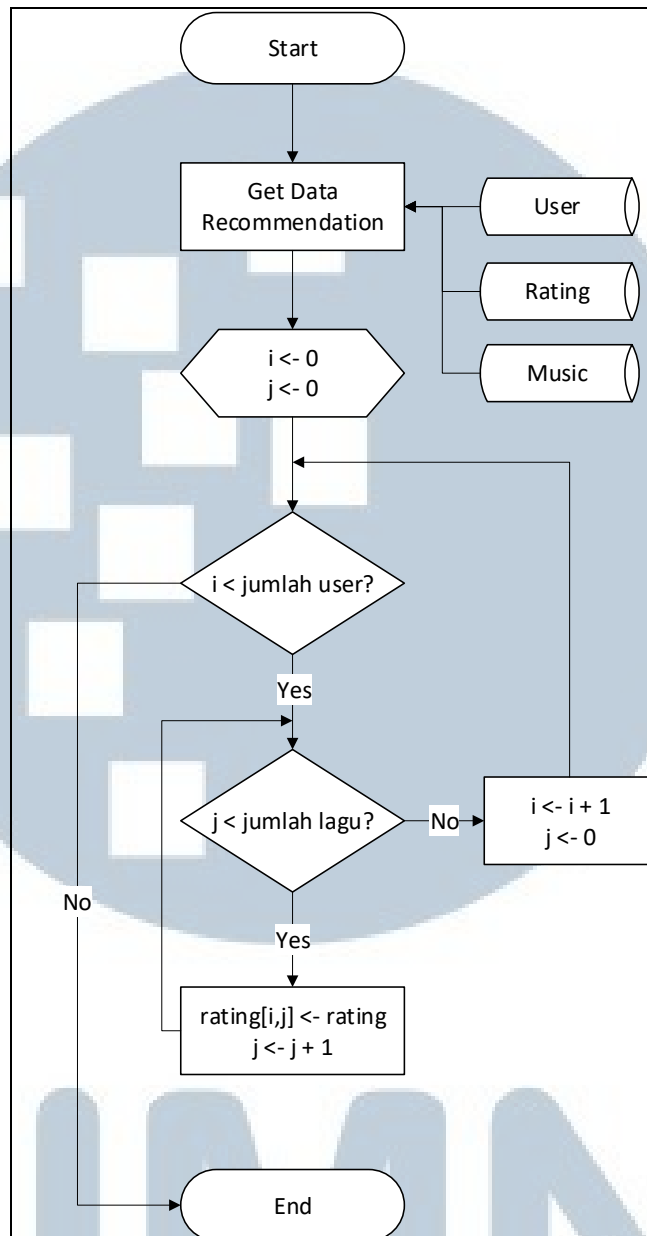
Gambar 3.22 merupakan *flowchart* Halaman Admin. Admin yang masuk ke dalam sistem dapat melakukan *fetch* data ke *database* yang bertujuan untuk mengambil data lagu seperti judul lagu, nama artis, gambar, dan *preview url* untuk disimpan ke dalam *database* sesuai dengan *music_id* nya. Admin juga dapat menghitung MAE dan ditampilkan di halaman Admin. Pada halaman Admin juga terdapat pengaturan yang dapat mengganti profil, kata sandi serta keluar dari sistem.



Gambar 3.22 Flowchart Halaman Admin

M. Flowchart User-Item Rating Matrix

Gambar 3.23 merupakan *flowchart* User-Item Rating Matrix. Sistem akan mengambil data *rating* dari *user* pada tabel Rating, data lagu pada tabel Music, dan data *user* pada tabel User. Kemudian akan dibuat matrix-nya dengan melakukan pengulangan sebanyak jumlah *user* dan jumlah *lagu*.



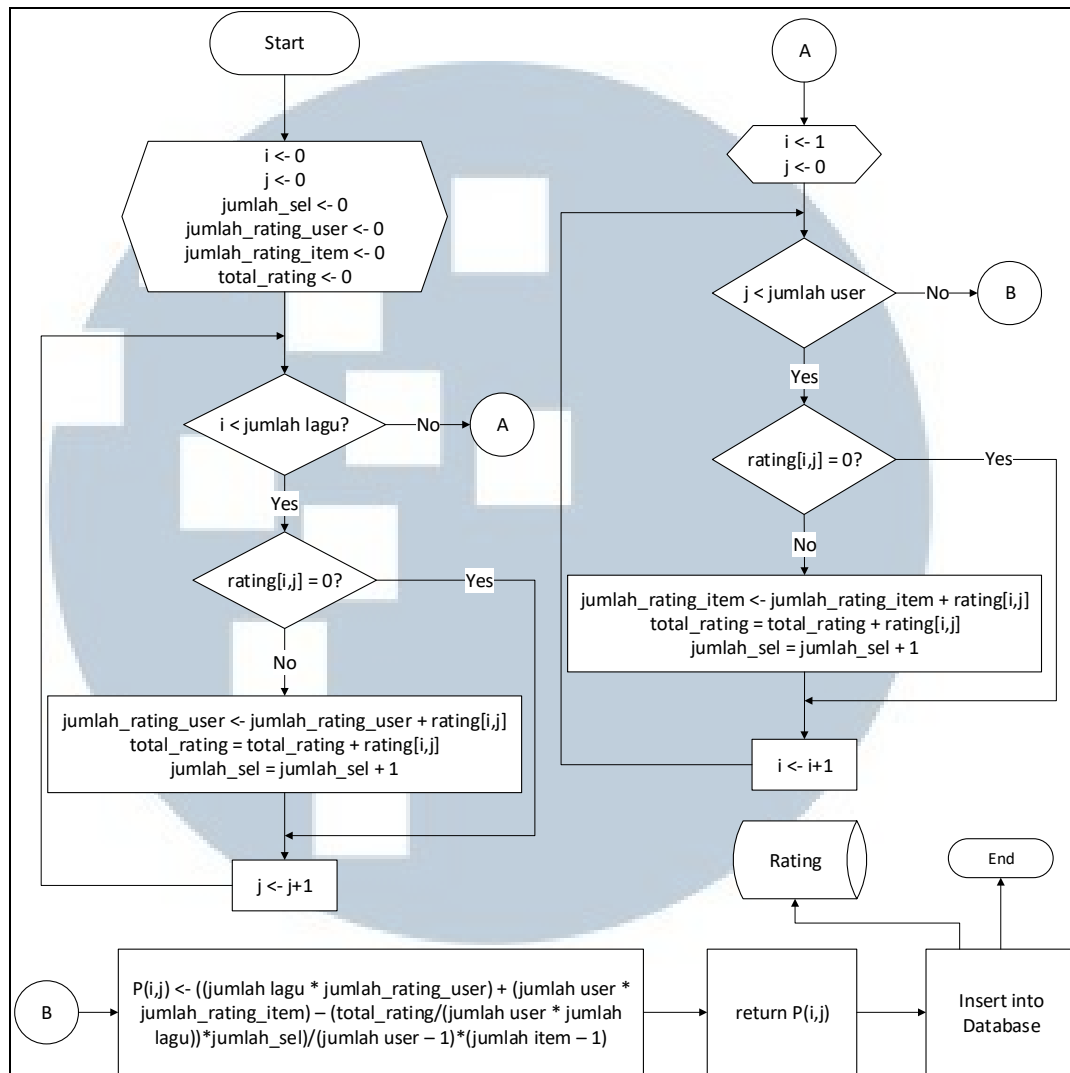
Gambar 3.23 Flowchart User-Item Rating Matrix

N. Flowchart Calculate Recommendation

Gambar 3.24 merupakan *flowchart* Calculate Recommendation. Aplikasi akan membuat *rating user-item matrix*. Akan dilakukan perhitungan prediksi menggunakan algoritma *adjusted MVA*. Pertama akan dilakukan inisialisasi untuk variable seperti *i*, *j*, *x*, *y*, *jumlah_sel*, *jumlah_rating_user*, *jumlah_rating_item*, dan *total_rating*. Untuk mendapatkan prediksi *rating* maka harus dilakukan perhitungan jumlah *rating* pada *user* dari lagu pertama hingga lagu terakhir. Akan dilakukan

pengecekan apakah pada kolom tersebut terdapat *rating* atau tidak. Apabila terdapat *rating* maka akan dilakukan penjumlahan *rating* ke variabel *jumlah_rating_user*, total *rating* ke variabel *total_rating*, dan jumlah selnya. Apabila pada kolom tersebut tidak terdapat *rating*, maka proses penjumlahan dilewati.

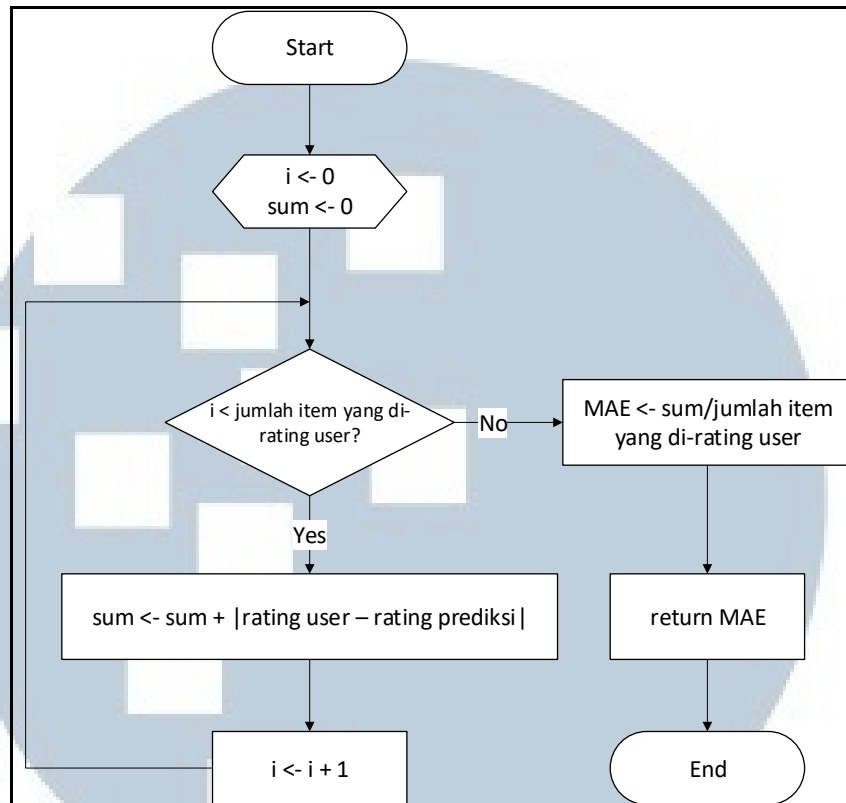
Setelah selesai mendapatkan jumlah *rating* pada *user*, selanjutnya akan dilakukan perhitungan jumlah *rating* pada *item* dari *user* pertama hingga *user* terakhir. Akan dilakukan pengecekan apakah kolom tersebut terdapat *rating* atau tidak. Apabila terdapat *rating* maka akan dilakukan penjumlahan *rating* ke variabel *jumlah_rating_item*, total *rating* ke variabel *total_rating*, dan jumlah selnya. Apabila pada kolom tersebut tidak terdapat *rating*, maka proses penjumlahan dilewati. Setelah mendapatkan nilai dari variabel *jumlah_sel*, *jumlah_rating_user*, *jumlah_rating_item*, dan *total_rating*, dilakukan perhitungan untuk mendapatkan prediksi *rating* dengan variabel $P(i,j)$ dengan mengalikan jumlah lagu dengan *jumlah_rating_user*, kemudian dijumlahkan dengan perkalian antara jumlah *user* dengan *jumlah_rating_item*, lalu dikurangi dengan perhitungan dari *total_rating* dibagi perkalian antara jumlah *user* dan jumlah lagu dikalikan dengan *jumlah_sel*, kemudian dibagi dengan perkalian antara $(user-1)$ dan $(item-1)$. Hasil prediksi akan dimasukkan ke *database* pada tabel *Rating*.



Gambar 3.24 Flowchart Calculate Recommendation

M. Flowchart Calculate MAE

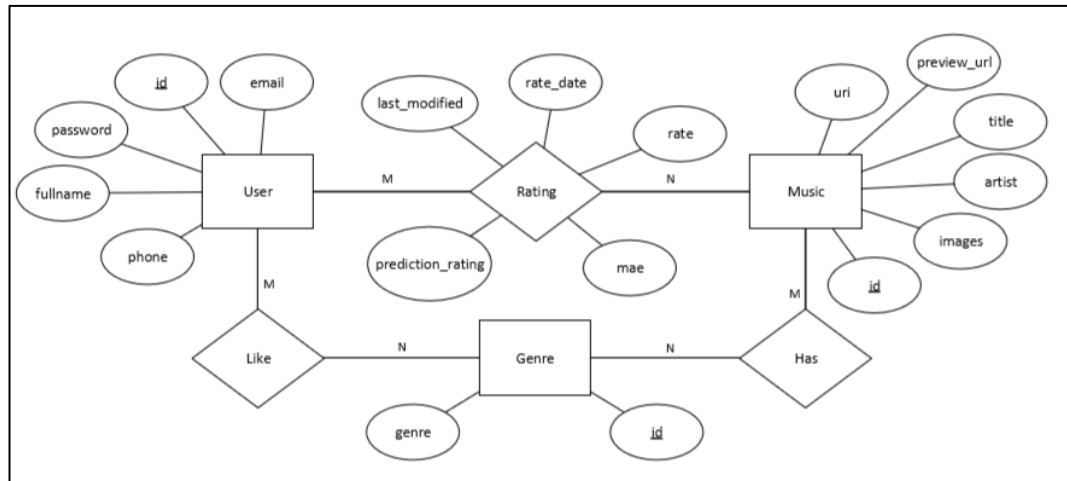
Gambar 3.25 merupakan *flowchart* Calculate MAE. Setelah *user* melakukan *rating* terhadap lagu-lagu yang direkomendasikan, maka sistem dapat menghitung tingkat keakuratan dari sistem rekomendasi tersebut dengan melakukan penjumlahan dari perbandingan nilai *rating* dari *user* dengan nilai *rating* prediksi rekomendasi, kemudian dibagi dengan jumlah lagu yang di-*rating*. Nilai pengurangan antara nilai *rating* dari *user* dengan nilai *rating* prediksi berupa nilai mutlak (positif).



Gambar 3.25 Flowchart Calculate MAE

3.2.5 Entity Relationship Diagram

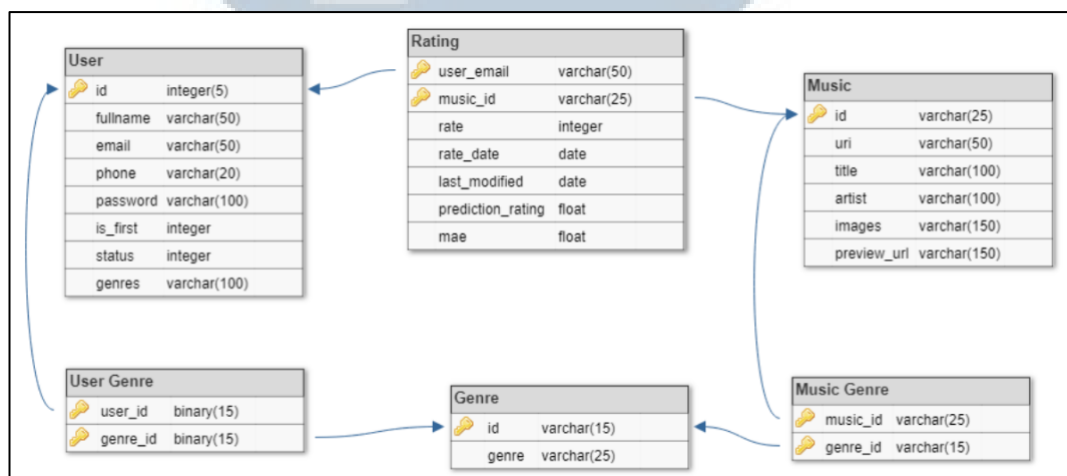
Gambar 3.26 adalah *Entity Relationship Diagram* dari sistem rekomendasi ini. Seorang *user* bisa me-*rating* banyak lagu, dan sebuah lagu bisa di-*rating* oleh banyak *user*. Seorang *user* bisa menyukai banyak *genre*, dan sebuah *genre* bisa disukai oleh banyak *user*. Sebuah lagu bisa memiliki banyak *genre*, dan sebuah *genre* bisa dimiliki banyak lagu. Seorang *user* bisa mendapatkan banyak rekomendasi lagu, dan sebuah lagu bisa direkomendasikan ke banyak *user*.



Gambar 3.26 Entity Relationship Diagram

3.2.6 Database Schema

Gambar 3.27 adalah *database schema* dari sistem rekomendasi ini. Sistem ini memiliki 7 tabel yang saling berelasi satu dengan yang lainnya, yang terdiri dari Rating, User, Music, Music Genre, Genre, dan User Genre.



Gambar 3.27 Database Schema

3.2.7 Struktur Tabel

Sistem ini memiliki 6 tabel yang memiliki fungsi masing-masing. Berikut merupakan struktur tabelnya.

A. Tabel User

Tabel 3.1 adalah struktur tabel User yang berfungsi untuk menyimpan seluruh akun *user* yang berada di dalam sistem.

Tabel 3.1 Struktur Tabel User

No.	Nama Kolom	Tipe	Panjang	Keterangan
1	id	int	5	<i>Primary key</i>
2	fullname	varchar	50	Nama lengkap
3	email	varchar	50	<i>Email user</i>
4	phone	varchar	20	Nomor telepon <i>user</i>
5	password	varchar	100	<i>Password User</i>
6	is_first	int		Menandakan <i>user</i> sudah memilih 3 genre awal
7	status	int		Menandakan <i>user</i> sudah <i>rating</i> minimal 10 lagu
8	genres	varchar	100	<i>Genre</i> awal yang disukai <i>user</i>

B. Tabel Music

Tabel 3.2 adalah struktur tabel Music yang berfungsi untuk menyimpan seluruh URI lagu yang berada di dalam sistem.

Tabel 3.2 Struktur Tabel Music

No.	Nama Kolom	Tipe	Panjang	Keterangan
1	id	varchar	25	<i>Primary key</i>
2	uri	varchar	50	ID <i>track</i> Spotify
3	title	varchar	100	Judul lagu
3	artist	varchar	100	Nama artis
4	images	varchar	150	Url gambar lagu
5	preview_url	varchar	150	Url <i>preview</i> lagu

C. Tabel Rating

Tabel 3.3 adalah struktur tabel Rating yang berfungsi untuk menyimpan seluruh *rating*.

Tabel 3.3 Struktur Tabel Rating

No.	Nama Kolom	Tipe	Panjang	Keterangan
1	user_email	varchar	50	Primary key dan foreign key ke tabel User
2	music_id	varchar	25	Primary key dan foreign key ke tabel Music
3	rate	int	-	Nilai rating 1-5
4	rate_date	date	-	Tanggal melakukan rating
5	last_modified	date	-	Perubahan terakhir rating
6	prediction_rating	float	-	Berisi prediksi rating dari perhitungan adjusted MVA
7	mae	Float	-	Nilai perbandingan prediksi dan rating dari user

D. Tabel User Genre

Tabel 3.4 adalah struktur tabel User Genre yang berfungsi untuk menyimpan seluruh *genre* yang disukai *user*.

Tabel 3.4 Struktur Tabel User Genre

No.	Nama Kolom	Tipe	Panjang	Keterangan
1	user_email	varchar	50	Primary key dan foreign key ke tabel User
2	genre_id	varchar	15	Primary key dan foreign key ke tabel Genre

E. Tabel Genre

Tabel 3.5 adalah struktur tabel Genre yang berfungsi untuk menyimpan seluruh *genre* yang terdapat dalam sistem rekomendasi ini.

Tabel 3.5 Struktur Tabel Genre

No.	Nama Kolom	Tipe	Panjang	Keterangan
1	id	varchar	15	Primary key
2	Genre	varchar	25	Genre lagu

F. Tabel Music Genre

Tabel 3.6 adalah struktur tabel Music Genre yang berfungsi untuk menyimpan seluruh *genre* yang dimiliki *music*.

Tabel 3.6 Struktur Tabel Music Genre

No.	Nama Kolom	Tipe	Panjang	Keterangan
1	genre_id	varchar	15	Primary key dan foreign key ke tabel Genre
2	music_id	varchar	25	Primary key dan foreign key ke tabel Music

3.2.8 Rancangan Antarmuka Aplikasi

Dalam mengimplementasikan sistem rekomendasi ini, dilakukan desain antarmuka terlebih dahulu. Gambar 3.28 merupakan rancangan antarmuka *splash screen* dengan menampilkan logo aplikasi saat aplikasi dibuka.



Gambar 3.28 Rancangan Antarmuka Splash Screen

Gambar 3.29 merupakan rancangan antarmuka *login*. User dapat melakukan *login* pada aplikasi ini jika sudah memiliki akun dengan memasukkan data seperti alamat *email* dan *password*. Tombol masuk digunakan untuk melakukan proses *login*. Apabila user belum memiliki akun, user dapat menekan tombol daftar untuk bernavigasi ke halaman Register dan melakukan registrasi.



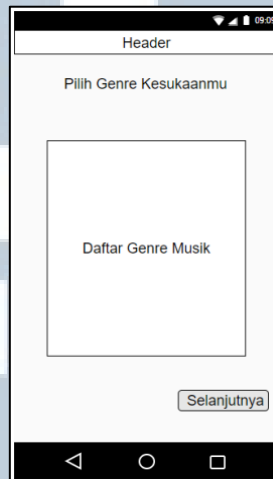
Gambar 3.29 Rancangan Antarmuka Login



Gambar 3.30 Rancangan Antarmuka Register

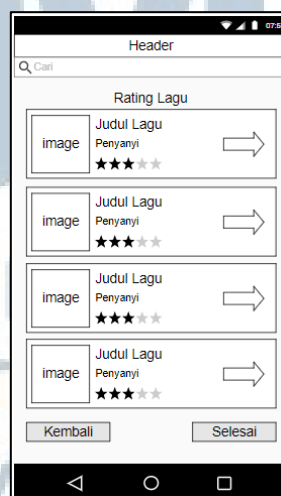
Gambar 3.30 merupakan rancangan antarmuka *register*. *User* dapat melakukan registrasi pada aplikasi ini jika belum memiliki akun. Terdapat *field* yang harus diisi oleh *user*, yaitu nama lengkap, alamat *email*, *password* dan konfirmasi *password*. Tombol daftar digunakan untuk melakukan pendaftaran akun. Apabila *user* sudah memiliki akun, *user* dapat menekan tombol masuk untuk bernavigasi ke halaman Login dan melakukan *login*.

Jika *user* pertama kali menggunakan aplikasi, maka akan diminta untuk memilih minimal tiga *genre* musik. Gambar 3.31 merupakan tampilan ketika *user* akan memilih *genre* musik.



Gambar 3.31 Rancangan Antarmuka Pilih Genre

Gambar 3.32 merupakan rancangan antarmuka halaman Rating Lagu Awal. Jika sudah memilih minimal 3 *genre* musik, *user* akan diminta untuk menekan tombol “Selanjutnya”. Ketika tombol ini ditekan maka akan terdapat daftar lagu yang harus di-*rating* oleh *user*. Setelah sudah memberikan *rating* terhadap minimal 10 lagu, tombol “Selesai” akan muncul. *User* juga dapat kembali ke halaman sebelumnya untuk memilih *genre* kembali.



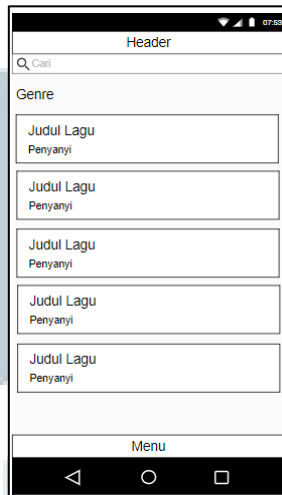
Gambar 3.32 Rancangan Antarmuka Rating Lagu Awal

Gambar 3.33 merupakan rancangan antarmuka halaman Musik. Halaman ini akan menampilkan daftar-daftar kelompok *genre* lagu dalam bentuk *card-view*.

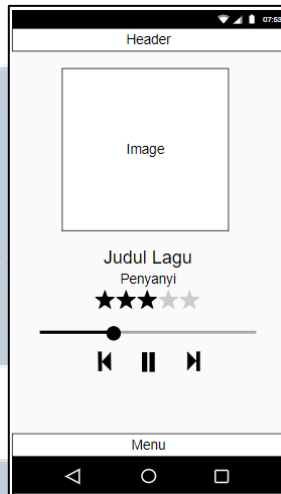


Gambar 3.33 Rancangan Antarmuka Musik

Gambar 3.34 merupakan rancangan antarmuka daftar lagu. Apabila *user* menekan salah satu dari daftar *genre* di atas, maka akan muncul tampilan daftar lagu yang terdiri dari nama lagu dan nama artis. Jika salah satu daftar lagu ditekan maka akan masuk ke halaman Preview seperti pada Gambar 3.35.

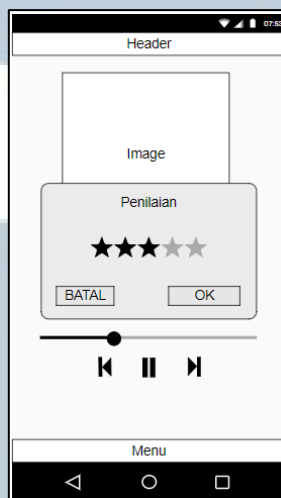


Gambar 3.34 Rancangan Antarmuka Daftar Lagu



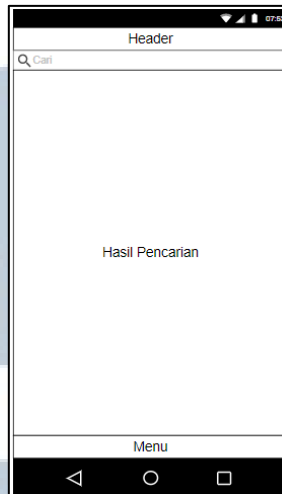
Gambar 3.35 Rancangan Antarmuka Preview

User dapat melakukan penilaian atau *rating* terhadap lagu yang sedang diputar dengan mengklik *icon* bintang pada Gambar 3.35. Ketika user ingin memberikan *rating* maka akan muncul dialog untuk memberikan *rating* terhadap lagu yang sedang dimainkan seperti pada Gambar 3.36.



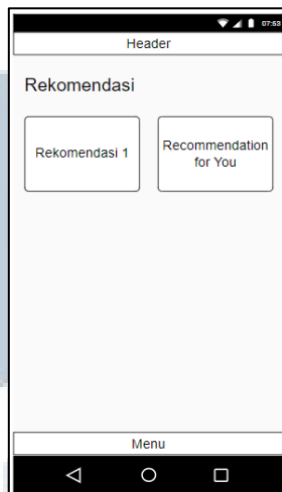
Gambar 3.36 Rancangan Antarmuka Rating

Gambar 3.37 merupakan rancangan antarmuka Cari. User dapat melakukan pencarian lagu maupun artis pada halaman ini dengan memasukkan *keyword* pada *text-field search*. Hasil dari pencarian berupa daftar lagu ataupun nama penyanyi, user juga dapat memutar lagu dari hasil pencarian.



Gambar 3.37 Rancangan Antarmuka Cari

Gambar 3.38 merupakan rancangan antarmuka rekomendasi. Pada halaman ini terdapat dua *card-view* yang menampilkan rekomendasi untuk *user*, yaitu Rekomendasi 1 yang menampilkan rekomendasi berdasarkan *genre* yang disukai oleh *user* dan Rekomendasi 2 yang menampilkan rekomendasi berdasarkan *user* lain. Jika *card-view* ditekan maka akan muncul daftar lagu seperti pada Gambar 3.34.



Gambar 3.38 Rancangan Antarmuka Rekomendasi

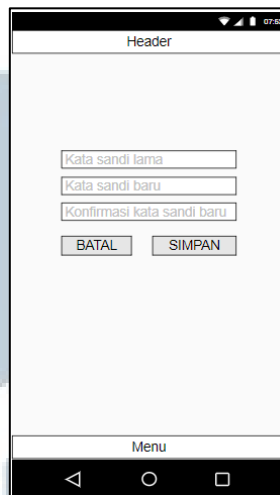
Gambar 3.39 merupakan rancangan antarmuka pengaturan profil. Pada halaman ini, *user* dapat melakukan pergantian nama lengkap atau nomor telepon dengan cara memasukkan nama lengkap atau nomor telepon yang baru. Tombol

simpan untuk menyimpan perubahan, tombol batal untuk kembali ke menu sebelumnya.



Gambar 3.39 Rancangan Antarmuka Pengaturan Profil

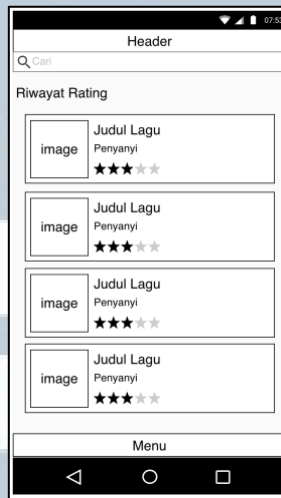
Gambar 3.40 merupakan rancangan antarmuka ganti kata sandi. Pada halaman ini, *user* dapat melakukan pergantian kata sandi dengan cara memasukkan kata sandi lama, kata sandi baru, dan konfirmasi kata sandi baru. Tombol simpan untuk menyimpan perubahan, tombol batal untuk kembali ke menu sebelumnya.



Gambar 3.40 Rancangan Antarmuka Ganti Kata Sandi

Gambar 3.41 merupakan rancangan antarmuka riwayat *rating*. Pada halaman ini, *user* dapat melihat lagu-lagu yang sudah diberikan *rating* oleh *user* dan diurutkan dari waktu terakhir melakukan *rating*. Daftar lagu yang ditampilkan

lengkap dengan *detail* lagu seperti judul lagu, nama artis, gambar, dan nilai *rating* yang diberikan oleh *user*.



Gambar 3.41 Rancangan Antarmuka Riwayat Rating

Gambar 3.42 merupakan rancangan antarmuka halaman *admin*. Pada halaman ini, *admin* dapat melakukan *fetch* lagu untuk megambil data lagu seperti nama artis, judul lagu, gambar, dan *link preview* lagu dari API Spotify lalu dimasukkan ke dalam *database*. Terdapat tombol untuk menghitung nilai MAE dari sistem dan menampilkannya pada halaman *admin*.



Gambar 3.42 Rancangan Antarmuka Halaman Admin