

**DPPL-001**

## **DESKRIPSI PERANCANGAN PERANGKAT LUNAK**

### **MUSIC FLOW**

untuk:

Penikmat Musik

Dipersiapkan oleh:

Kelompok 2 IF-44-03

Raden Fasya Mazaya Putri	1301200390
Ahmad Fasya Adila	1301204231
Muhammad Satrio Tri Nugraha	1301204045
Karuna Dewa Satyananda	1301204295

Program Studi S1 Informatika

Fakultas Informatika, Universitas Telkom

Jl. Telekomunikasi 1, Dayeuhkolot Bandung

Telkom University	Prodi S1- Informatika Universitas Telkom	Nomor Dokumen	Halaman
		DPPL-01 <xx:no grp>	<1>/<jml #
		Revisi	<nomor revisi>
			Tgl: 28/10/2022

## DAFTAR PERUBAHAN

<b>Revisi</b>	<b>Deskripsi</b>
<b>A</b>	
<b>B</b>	
<b>C</b>	
<b>D</b>	
<b>E</b>	
<b>F</b>	
<b>G</b>	

INDEX TGL	-	A	B	C	D	E	F	G
Ditulis oleh								
Diperiksa oleh								
Disetujui oleh								

## **Daftar Halaman Perubahan**

<b>Halaman</b>	<b>Revisi</b>	<b>Halaman</b>	<b>Revisi</b>

## Daftar Isi

Daftar Isi	3
1. Pendahuluan	6
1.1 Tujuan Penulisan Dokumen	6
1.2 Lingkup Masalah	6
1.3 Definisi dan Istilah	7
1.4 Referensi	8
2 Perancangan Global	8
2.1 Rencana Lingkungan Implementasi	8
2.2 Deskripsi Arsitektur Perangkat Lunak	9
3 Perancangan Rinci	10
3.1 Realisasi Use Case	10
3.1.1 Use Case #1 Login	11
3.1.1.1 Use Case Scenario #1 Login	11
3.1.1.2 UI Design dan Deskripsi Objek UI #1 Login	12
3.1.1.3 Identifikasi Object dan Tipe #1 Login	13
3.1.1.4 Sequence Diagram #1 Login	13
3.1.2 Use Case #2 Registrasi	14
3.1.2.1 Use Case Scenario #2 Registrasi	14
3.1.2.2 UI Design dan Deskripsi Objek UI #2 Registrasi	15
3.1.2.3 Identifikasi Object Baru & Tipe Kelas #2 Registrasi	16
3.1.2.4 Sequence Diagram #2 Registrasi	16
3.1.3 Use Case #3 Pencarian Lagu	16
3.1.3.1 Use Case Scenario #3 Pencarian Lagu	16
3.1.3.2 UI Design dan Deskripsi Objek UI #3 Pencarian Lagu	17
3.1.3.3 Identifikasi Object Baru & Tipe Kelas #3 Pencarian Lagu	18
3.1.3.4 Sequence Diagram #3 Pencarian Lagu	19
3.1.4 Use Case #4 Memutar Lagu	20
3.1.4.1 Use Case Scenario #4 Memutar Lagu	20
3.1.4.2 UI Design dan Deskripsi Objek UI #4 Memutar Lagu	20
3.1.4.3 Identifikasi Object Baru & Tipe Kelas #4 Memutar Lagu	21
3.1.4.4 Sequence Diagram #4 Memutar Lagu	21
3.1.5 Use Case #5 Kelola Favourites	23
3.1.5.1 Use Case Scenario #5 Kelola Favourites	23
3.1.5.2 UI Design dan Deskripsi Objek UI #5 Kelola Favourites	23
3.1.5.3 Identifikasi Object Baru & Tipe Kelas #5 Kelola Favourites	24
3.1.5.4 Sequence Diagram #5 Kelola Favourites	25

3.1.6 Use Case #6 Kelola Queue	26
3.1.6.1 Use Case Scenario #6 Kelola Queue	26
3.1.6.2 UI Design dan Deskripsi Objek UI #6 Kelola Queue	26
3.1.6.3 Identifikasi Object Baru & Tipe Kelas #6 Kelola Queue	27
3.1.6.4 Sequence Diagram #6 Kelola Queue	27
3.1.7 Use Case #7 Kelola User	28
3.1.7.1 Use Case Scenario #7 Kelola User	28
3.1.7.2 UI Design dan Deskripsi Objek UI #7 Kelola User	28
3.1.7.3 Identifikasi Object Baru & Tipe Kelas #7 Kelola User	29
3.1.7.4 Sequence Diagram #7 Kelola User	29
3.1.8 Use Case #8 Kelola Lagu	30
3.1.8.1 Use Case Scenario #8 Kelola Lagu	30
3.1.8.2 UI Design dan Deskripsi Objek UI #8 Kelola Lagu	31
3.1.8.3 Identifikasi Object Baru & Tipe Kelas #8 Kelola Lagu	33
3.1.8.4 Sequence Diagram #8 Kelola Lagu	33
3.1.9 Use Case #9 Kelola Musisi	34
3.1.9.1 Use Case Scenario #9 Kelola Musisi	34
3.1.9.2 UI Design dan Deskripsi Objek UI #9 Kelola Penyanyi	35
3.1.9.3 Identifikasi Object Baru & Tipe Kelas #9 Kelola Penyanyi	37
3.1.9.4 Sequence Diagram #9 Kelola Penyanyi	37
3.2 Diagram Kelas Keseluruhan	39
3.3 Perancangan Data / Basis Data	40
3.4 Perancangan Algoritma dan/atau Query	41
<b>4. Matriks Kerunutan (Requirement Traceability Matrix)</b>	<b>51</b>

Setelah Daftar Isi Boleh ada Daftar Tabel dan Daftar Gambar

# 1. Pendahuluan

## 1.1 Tujuan Penulisan Dokumen

Dokumen Deskripsi Perancangan Perangkat Lunak (DPPL) adalah dokumen yang berisi deskripsi dari perangkat lunak yang akan dikembangkan. Dokumen ini ditulis untuk menjelaskan deskripsi kebutuhan perangkat lunak dari aplikasi Music Flow.

.Musik merupakan bagian yang tidak terpisahkan dalam kehidupan sehari-hari. Terdapat kiasan dalam bahasa Inggris yang menyebut "*music makes the world goes round*" atau musik membuat dunia berputar. Sebab, musik-lah yang seringkali menjadi teman ketika bekerja, ketika merasa senang, dan merasa sedih. Bayangkan apabila dunia tidak mengenal musik. Kejemuhan akan melanda, menjadikan kita tidak bersemangat melakukan aktivitas. Ketika sedih, tidak akan ada alunan nada menenangkan yang dapat memperbaiki suasana hati.

Hal tersebut merupakan indikasi betapa pentingnya musik dalam kehidupan manusia. Alan P. Merriam dalam buku The Anthropology Of Music menjelaskan beberapa fungsi musik, yaitu fungsi musik sebagai pengungkapan emosional, fungsi musik untuk penghayatan estetis, fungsi komunikasi, fungsi kesinambungan budaya, fungsi ekonomi, dan yang terpenting adalah fungsi hiburan. Oleh karena itu, dengan adanya website pemutar musik online Music Flow diharapkan dapat menjadi salah satu solusi dari fungsi-fungsi tersebut.

## 1.2 Lingkup Masalah

Dokumen ini menjelaskan pengembangan perangkat lunak tentang Music Flow yaitu perangkat lunak berbasis website yang digunakan untuk memainkan musik dan juga tempat untuk mengunggah karya-karya musik. Perangkat lunak pemutar musik online berbasis website ini dikembangkan dengan tujuan :

- Login user dan admin
- Register user
- Fitur pencarian lagu
- Memutar lagu
- Menghentikan lagu (Pause)
- Mengatur volume
- Next lagu
- Previous lagu
- Pengelolaan queue lagu
- Pengelolaan favorite lagu

- Pengelolaan data user
- Pengelolaan data penyanyi
- Pengelolaan data lagu

Penerapan perangkat lunak yang dispesifikasikan pada dokumen ini bermanfaat untuk mewadahi orang-orang yang berkecimpung di dunia musik dan juga untuk masyarakat umum penikmat musik. Sasaran dari pembuatan perangkat lunak Music Flow adalah masyarakat umum terutama generasi milenial.

### **1.3 Definisi dan Istilah**

- *Music App* : Istilah umum untuk mengacu kepada perangkat lunak komputer yang dapat memainkan berkas multimedia.
- *Admin* : Orang yang punya tugas untuk melakukan tata kelola administrasi website
- *User* : Pengguna yang dapat menggunakan website serta melalui proses login
- *Guest User* : Pengguna yang dapat menggunakan website tanpa melalui proses login dengan batasan tertentu
- *Play* : Perintah untuk memulai
- *Pause* : Perintah untuk berhenti
- *Skip* : Perintah untuk melewati lagu yang sedang diputar
- *Previous* : Perintah untuk memutar lagu sebelumnya
- *Loop* : Perintah untuk memutar lagu kembali
- *Queue* : Dalam ilmu komputer, antrian adalah koleksi dari data-data yang memiliki urutan dan hanya bisa diubah dengan menambahkan data di satu ujung dan mengeluarkan data di ujung lainnya.
- *Usecase Diagram* : Gambaran atau representasi dari interaksi yang terjadi antara sistem dan lingkungannya
- *Class Diagram* : Diagram UML yang menggambarkan kelas-kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi.
- *Warning* : Sebuah pesan bahwa terdapat peringatan tentang apa yang sedang dikerjakan

## **1.4 Referensi**

Beberapa textbook, panduan, atau dokumentasi lain yang digunakan sebagai acuan dalam pengembangan perangkat lunak ini adalah sebagai berikut.

1. Materi Perkuliahan RPL: Transisi - Analisis ke Perancangan
2. SKPL RPL : Implementasi dan Pengujian PL IF-44-03 Kelompok 2  
[https://bit.ly/SRS\\_Kelompok2\\_RPL](https://bit.ly/SRS_Kelompok2_RPL)
3. [www.spotify.com](http://www.spotify.com)

## **2 Perancangan Global**

### **2.1 Rencana Lingkungan Implementasi**

Perangkat lunak yang kami bangun berupa sebuah website yang dapat diakses menggunakan beberapa web browser seperti Google Chrome, Mozilla Firefox, Internet Explorer, dan web browser lainnya. Website Music Flow juga dapat diakses menggunakan berbagai macam sistem operasi seperti Windows, Ios, Linux, dan sebagainya.

Adapun spesifikasi minimal dari perangkat keras user yang diperlukan untuk mengakses website Music Flow sebagai berikut :

- Sistem Operasi : Windows 8/10
- RAM : 4 GB
- Processor : Intel Core i3 8th Gen
- Free disk space : 500mb
- Penyimpanan : SSD, HDD

Perangkat lunak dari website Music Flow juga akan mendapatkan waktu maintenance yang dimana akan meningkatkan performa dan juga kualitas atau fitur yang dapat diperbarui. Untuk menunjang pembuatan website ini diperlukan perangkat keras dari sisi server sebagai berikut :

- Sistem Operasi : Windows 7/8/10
- RAM : 4 GB
- Processor : Intel Core i5 8th Gen
- Frontend : Bootstrap
- Backend : PHP
- DBMS : MySQL

## **2.2 Deskripsi Arsitektur Perangkat Lunak**

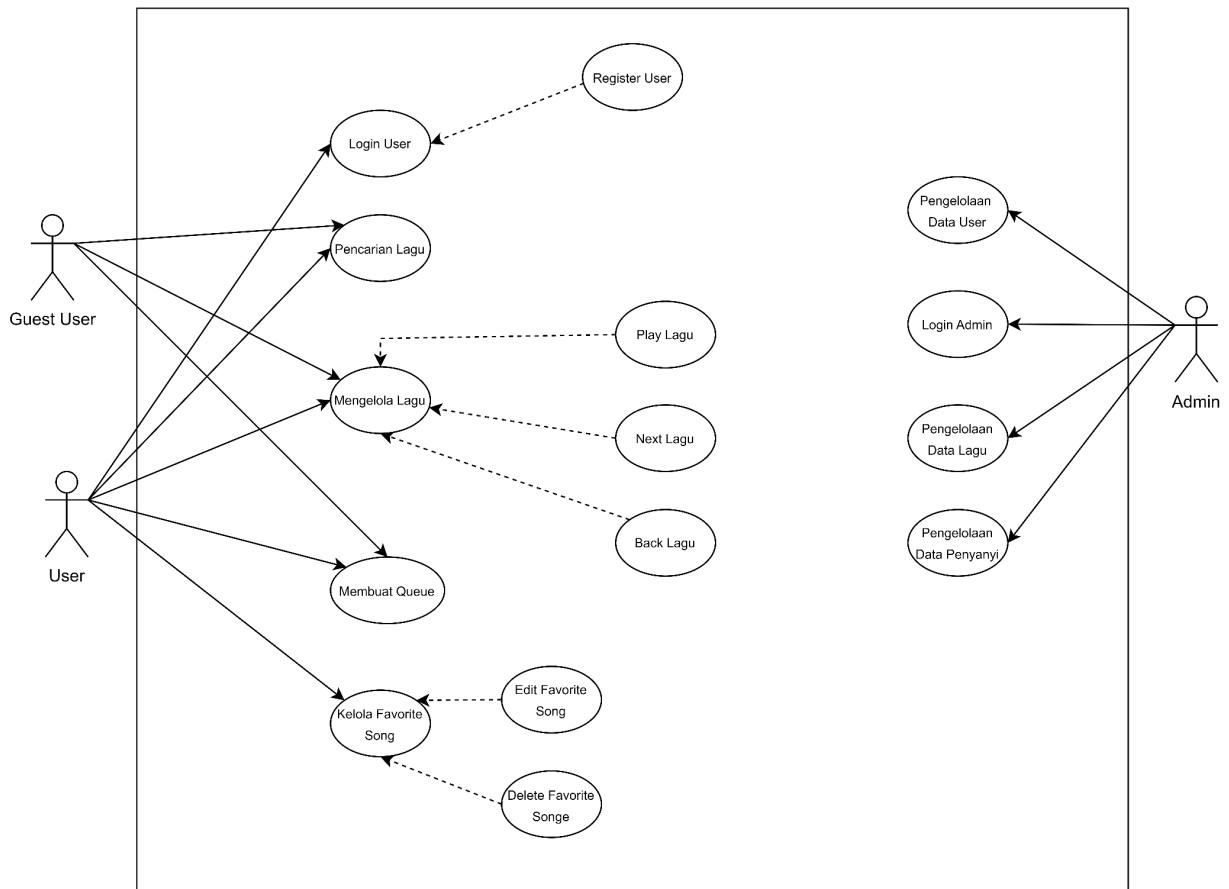
Dilisik dengan gambar komponen diagram atau daftar modul, dapat dituliskan dalam bentuk tabel berikut:

No	Nama Komponen	Keterangan
1	Login	User dan Admin melakukan login untuk masuk ke dalam aplikasi
2	Register	User melakukan Register untuk masuk ke dalam aplikasi
3	Pencarian Lagu	Fitur ini menyediakan layanan untuk mencari suatu lagu yang diinginkan oleh user
4	Memutar Lagu	Fitur ini menyediakan layanan untuk memutar lagu yang dipilih oleh user
5	Kelola Favourite	Fitur ini menyediakan layanan untuk membuat playlist lagu favourite yang diinginkan user
6	Kelola Queue	Fitur ini menyediakan layanan mengelola antrian yang telah dibuat oleh user
7	Kelola User	Fitur menyediakan layanan mengelola user oleh admin
8	Kelola Lagu	Fitur menyediakan layanan mengelola lagu oleh admin
9	Kelola Penyanyi/Artis	Fitur menyediakan layanan mengelola penyanyi/artis oleh admin

### 3 Perancangan Rinci

#### 3.1 Realisasi Use Case

Kami membuat Use Case Diagram yang menggabungkan dan menggambarkan seluruh usecase website Music Flow. Use Case Diagram yang telah kami buat juga bisa digunakan untuk gambaran apa yang akan dibangun dalam website Music Flow.



Berisi TABEL USE CASE sebagai berikut :

No	Nama Use Case	Deskripsi Use Case
#1	Login	Use Case Login User berfungsi untuk mengauthentifikasi user ketika akan memasuki website dengan menggunakan username dan password yang diperoleh dari admin. Username dan password tersebut diperoleh setelah user mengisi data-data yang dibutuhkan pada saat registrasi. Tujuan dari use case Login ini adalah agar hanya user yang telah melakukan registrasi yang dapat memasuki website ini
#2	Register	Use Case Register berfungsi untuk mendaftarkan data data user yang akan disimpan di tabel database User. Tujuan dari use case Register ini adalah agar user dapat melakukan proses Login sebelum memasuki halaman utama website.

#3	Pencarian Lagu	Use Case Pencarian Lagu berfungsi untuk mencari suatu lagu yang diinginkan yang sudah terdata dalam database Lagu. Tujuan dari use case Pencarian lagu adalah agar user dapat melakukan proses pencarian lagu dengan hanya mengetik nama judul lagunya saja.
#4	Membuat Queue	Use Case Membuat Queue berfungsi untuk membuat antrian lagu yang diinginkan oleh User. Tujuan dari use case Membuat Queue adalah agar user dapat membuat antrian baru.
#5	Kelola Favourite	Use Case Kelola Favourite berfungsi untuk menampilkan daftar lagu favourite yang sudah dibuat oleh User. Tujuan dari use case Kelola Favorite adalah agar user dapat melihat dan mengelola daftar lagu favourite yang sudah dibuat oleh user.
#6	Mengelola Data Artis/Penyanyi	Use Case Mengelola Data Artis/Penyanyi berfungsi untuk menampilkan pengelolaan data Artis/Penyanyi yang terdaftar. Tujuan dari Use Case Mengelola Data Artis/Penyanyi adalah agar admin dapat mendata dan mengelola daftar artis/penyanyi yang sudah dibuat terdaftar di sistem.
#7	Mengelola Data User	Use Case Mengelola Data User berfungsi untuk menampilkan pengelolaan data User yang terdaftar. Tujuan dari Use Case Mengelola Data User adalah agar admin dapat mendata dan mengelola daftar user yang sudah dibuat terdaftar di sistem.
#8	Mengelola Data Lagu	Use Case Mengelola Data Lagu berfungsi untuk menampilkan pengelolaan data Lagu yang terdaftar. Tujuan dari Use Case Mengelola Data Lagu adalah agar admin dapat mendata dan mengelola daftar lagu yang sudah dibuat terdaftar di sistem.
#9	Memutar Lagu	Use Case Memutar Lagu berfungsi untuk menampilkan lagu yang ingin diputar. Tujuan dari Use Case Memutar Lagu adalah menampilkan lagu yang mirip dengan yang diinginkan user dan terdapat pada website.

### 3.1.1 Use Case #1 Login

#### 3.1.1.1 Use Case Scenario #1 Login

Skenario Use Case #1 :

- i. Pre-Condition : User belum login ke dalam sistem
- ii. Use Case Description
  - a. Primary Flow

Aktor	Sistem
Memasukkan username dan password yang telah diperoleh dari admin, klik tombol “Login”	
	Mengecek username dan password yang dimasukkan oleh user (Alternative Flow)

	Menampilkan halaman home dari website
--	---------------------------------------

b. Alternative Flow

User salah memasukkan username dan password, maka

- Sistem akan memberikan warning yang menyatakan "Incorrect username or password!"
- Sistem akan meminta user memasukkan kembali username dan password (primary flow langkah 1)

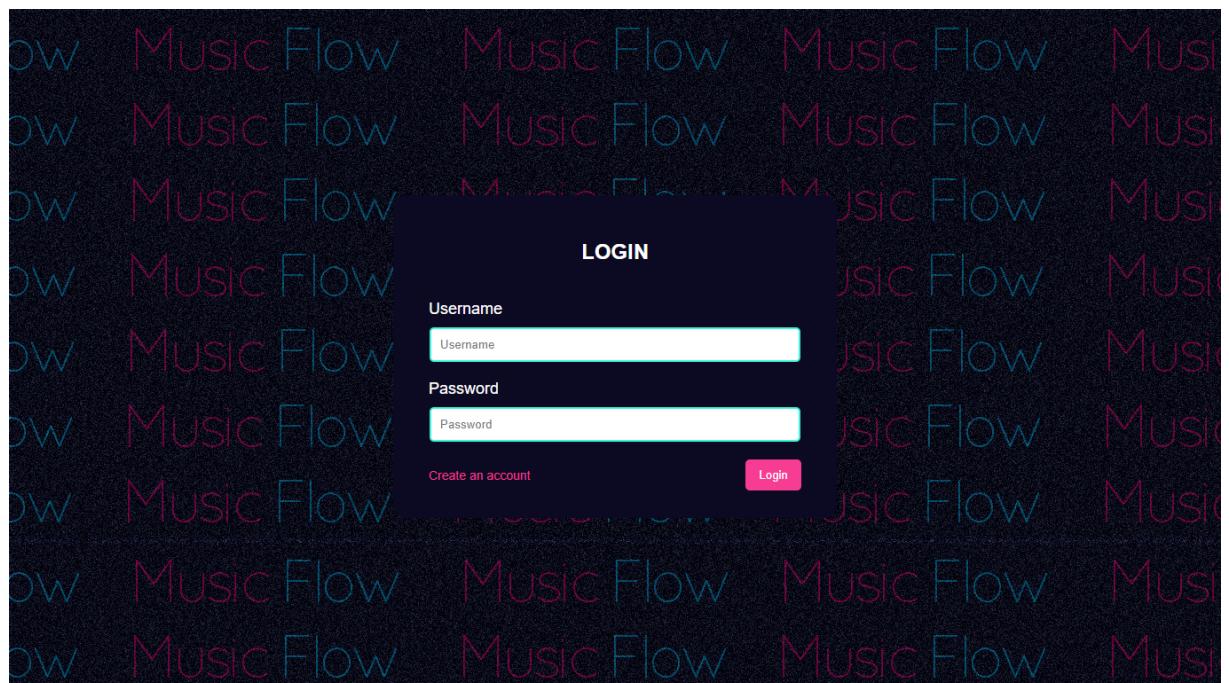
User tidak memasukkan password, maka

- Sistem akan memberikan warning yang menyatakan "Password cant be empty!"
- Sistem akan meminta user memasukkan kembali username dan password (primary flow langkah 1)

- iii. Post-Condition : User telah melakukan login dan masuk ke halaman home dari website ini.

### 3.1.1.2 UI Design dan Deskripsi Objek UI #1 Login

*Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detailnya.*



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
001	<i>Login Page</i>	<i>Halaman Login akun untuk pengguna</i>

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detail  
*Antarmuka 001: {diisi dengan no. layar atau no gambar rancangan antarmuka}*

#### *Page Login Page*

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>btn_login</i>	<i>Button</i>	<i>Login</i>	<i>Jika di klik, akan mengaktifkan Function Login</i>
<i>btn_create_account</i>	<i>Button</i>	<i>Create an account</i>	<i>Jika diklik akan mengaktifkan link ke halaman Register</i>
<i>txt_username</i>	<i>Text Field editable</i>	<i>Username</i>	<i>Text Field yang digunakan untuk mengisi username pengguna</i>
<i>txt_password</i>	<i>Text Field editable</i>	<i>Password</i>	<i>Text Field yang digunakan untuk mengisi password pengguna</i>

#### **3.1.1.3 Identifikasi Object dan Tipe #1 Login**

*Identifikasi object yang terkait dengan use case tersebut.*

*Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.*

*Gunakan tabel di bawah:*

**TABEL OBJECT PERANCANGAN**

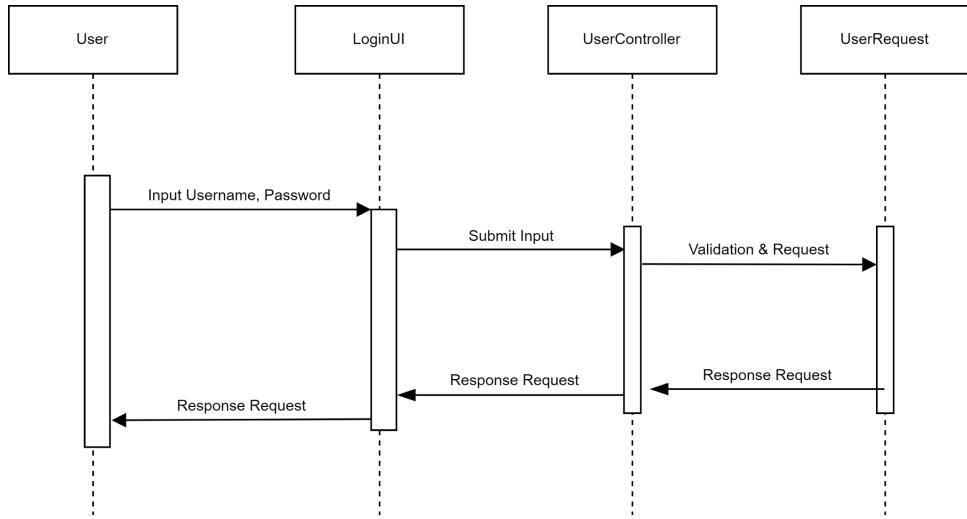
<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	<i>Database</i>	<i>Entity</i>
2.	<i>User</i>	<i>Entity</i>
3.	<i>LoginUI</i>	<i>Boundary</i>
4.	<i>User Controller</i>	<i>Controller</i>
5.	<i>User Request</i>	<i>Controller</i>

*\*Tipe kelas diisi dengan Boundary(Interface), Entity(Database), Controller*

#### **3.1.1.4 Sequence Diagram #1 Login**

*Buatlah diagram sequence untuk masing - masing use case sesuai skenario.*

*Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.*



### 3.1.2 Use Case #2 Registrasi

#### 3.1.2.1 Use Case Scenario #2 Registrasi

Skenario Use Case #2 :

- Pre-Condition : User belum pernah register ke dalam sistem
- Use Case Description
  - Primary Flow

Aktor	Sistem
Klik tombol “Create an account” pada halaman Login	
	Menampilkan halaman Sign Up
Memasukkan <i>username</i> , <i>password</i> dan <i>confirm password</i> , klik tombol “Sign Up”	
	Mengecek username yang dimasukkan oleh user (Alternative Flow)
	Mengecek kecocokan password yang dimasukkan oleh user (Alternative Flow)
	Menampilkan halaman home dari website

#### b. Alternative Flow

Username sudah dipakai sebelumnya, maka

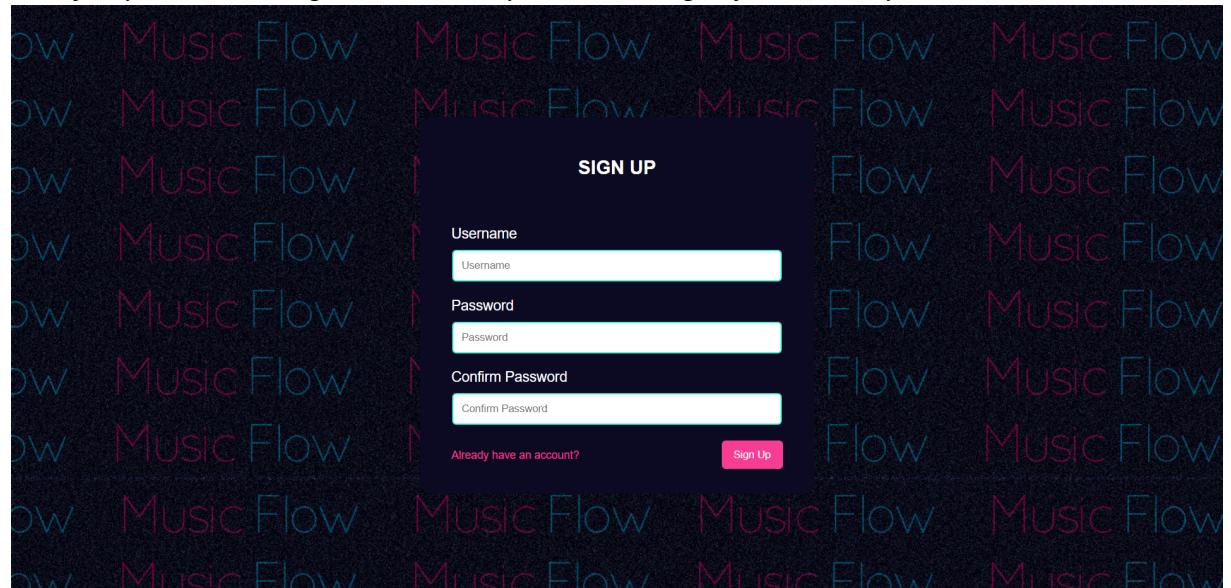
- Sistem akan memberikan warning yang menyatakan "User is already exist"
- Sistem akan meminta user memasukkan kembali username, password dan confirmation password (basic flow langkah 1)

Password yang dimasukkan tidak cocok, maka

- Sistem akan memberikan warning yang menyatakan "The confirmation password does not match !"
  - Sistem akan meminta user memasukkan kembali username, password dan confirmation password (basic flow langkah 1)
- iii. Post-Condition : User telah melakukan register dan dapat melakukan login untuk masuk ke server

### 3.1.2.2 UI Design dan Deskripsi Objek UI #2 Registrasi

*Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya.*



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
002	Register Page	Halaman Register akun untuk pengguna

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detail Antarmuka 002: {diisi dengan no. layar atau no gambar rancangan antarmuka}

Page Register Page

ID_OBJEK	JENIS	LABEL*	KETERANGAN**
btn_signup	Button	Sign Up	Jika di klik, akan mengaktifkan Function Register
btn_has_account	Button	Already have an account?	Jika diklik akan mengaktifkan link ke halaman Login
txt_username	Text Field editable	Username	Text Field yang digunakan untuk mengisi username pengguna
txt_password	Text Field editable	Password	Text Field yang digunakan untuk mengisi password pengguna

PRODI S1 INFORMATIKA TEL-U	DPPL-001	HALAMAN 15 DARI 53
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>txt_confirm_password</i>	<i>Text Field editable</i>	<i>Confirm Password</i>	<i>Text Field yang digunakan untuk mengisi confirmation password pengguna</i>

### 3.1.2.3 Identifikasi Object Baru & Tipe Kelas #2 Registrasi

*Identifikasi object yang terkait dengan use case tersebut.*

*Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.*

*Gunakan tabel di bawah:*

TABEL OBJECT PERANCANGAN

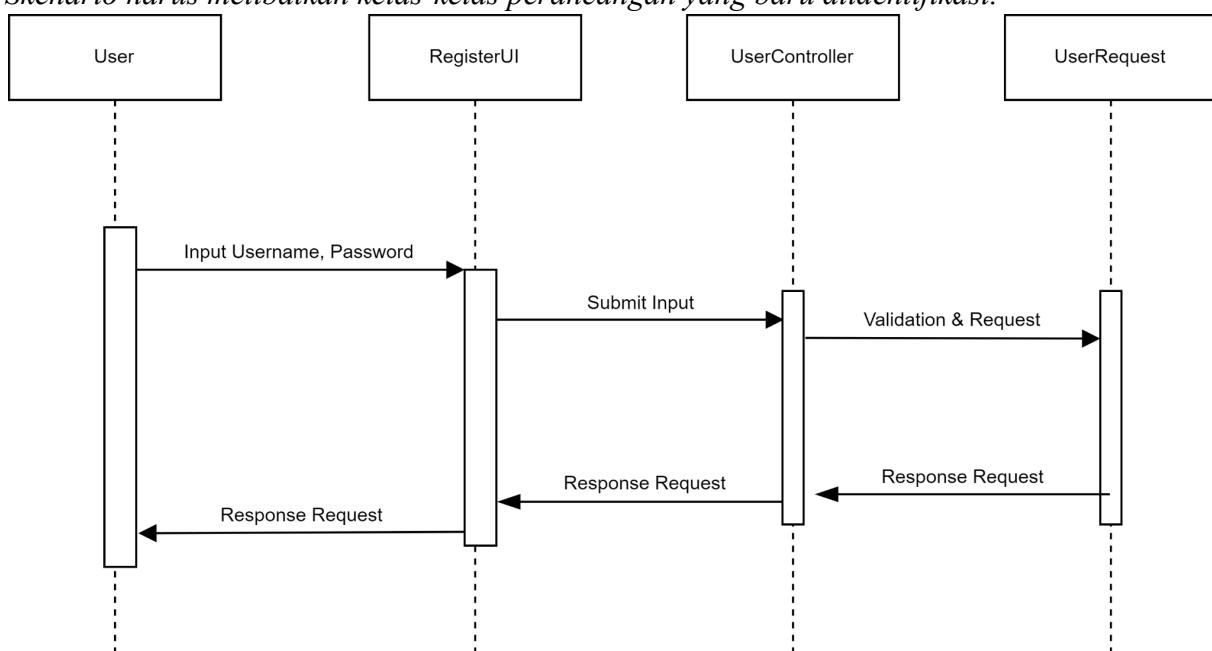
<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1.	<i>Database</i>	<i>Entity</i>
2.	<i>User</i>	<i>Entity</i>
3.	<i>RegisterUI</i>	<i>Boundary</i>
4.	<i>User Controller</i>	<i>Controller</i>
5.	<i>User Request</i>	<i>Controller</i>

*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

### 3.1.2.4 Sequence Diagram #2 Registrasi

*Buatlah diagram sequence untuk masing - masing use case sesuai skenario.*

*Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.*



### 3.1.3 Use Case #3 Pencarian Lagu

#### 3.1.3.1 Use Case Scenario #3 Pencarian Lagu

Skenario Use Case #3 :

Prodi S1 Informatika Tel-U	DPPL-001	Halaman 16 dari 53
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

- i. Pre-Condition : Sistem menampilkan semua Lagu yang tersedia
- ii. Use Case Description
  - a. Primary Flow

Aktor	Sistem
Isi nama judul lagu atau nama musisi yang ingin dicari pada kolom Pencarian.	
	Menampilkan halaman kosong (Alternative Flow)
	Menampilkan lagu yang dicari

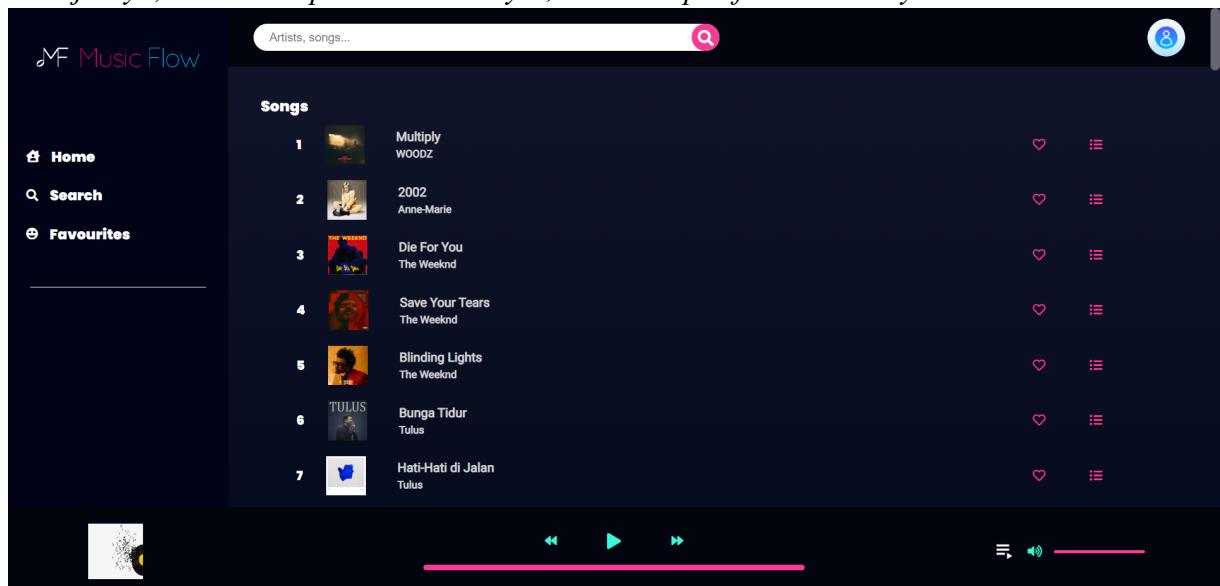
b. Alternative Flow

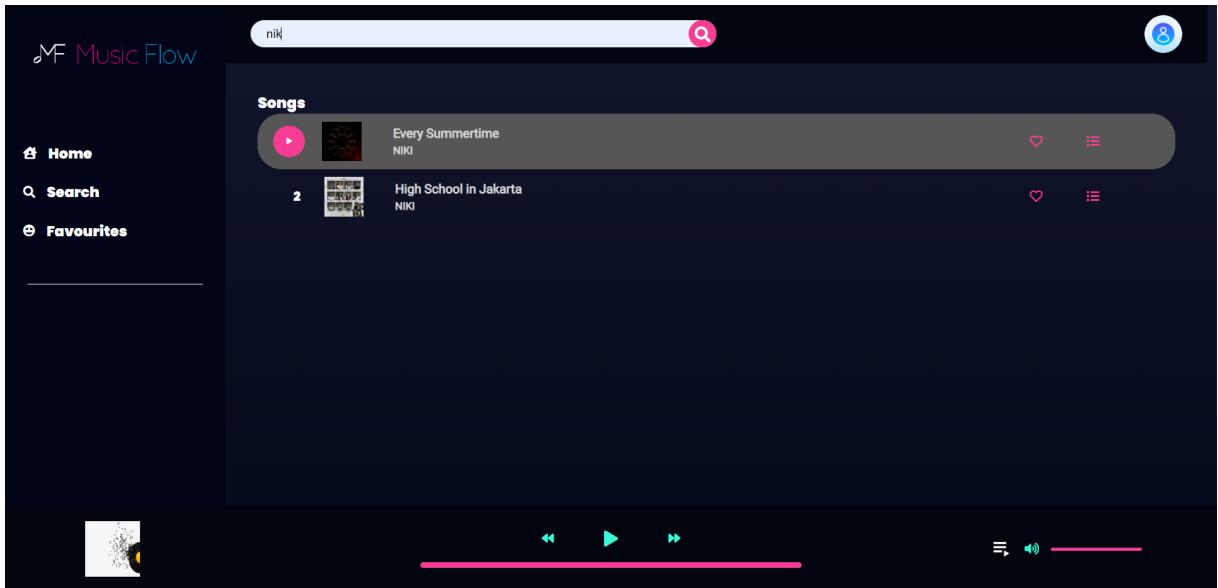
- Lagu atau musisi belum terdata / terdaftar pada website, maka
- Sistem akan menampilkan halaman kosong

- iii. Post-Condition : Sistem menampilkan Lagu yang dicari.

### 3.1.3.2 UI Design dan Deskripsi Objek UI #3 Pencarian Lagu

Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detailnya.





Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
003	Search Page	Halaman Search pada aplikasi

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil  
*Antarmuka 003: {diisi dengan no. layar atau no gambar rancangan antarmuka}*

Page Search Page

Id_Objek	JENIS	LABEL*	Keterangan**
txt_search	Text Field editable	Artists, songs...	Text Field yang digunakan untuk mengisi nama lagu atau musisi yang ingin dicari
btn_search	Button	Icon	Jika di klik, akan mengaktifkan Function Search
btn_favourites	Button	Icon	Jika di klik, akan mengaktifkan Function Favourites
btn_queue	Button	Icon	Jika di klik, akan mengaktifkan Function Queue
btn_play	Button	Icon	Jika di klik, akan mengaktifkan Function Play
txt_number	Label	1 - dst (number)	Label yang menampilkan informasi urutan dengan angka
txt_artists	Label	Nama Musisi	Label yang menampilkan nama musisi
txt_songs	Label	Judul Lagu	Label yang menampilkan judul lagu
img_album	Image	-	Image dari cover album/lagu dari lagu

### 3.1.3.3 Identifikasi Object Baru & Tipe Kelas #3 Pencarian Lagu

Identifikasi object yang terkait dengan use case tersebut.

Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.

Gunakan tabel di bawah:

Prodi S1 Informatika Tel-U	DPPL-001	Halaman 18 dari 53
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

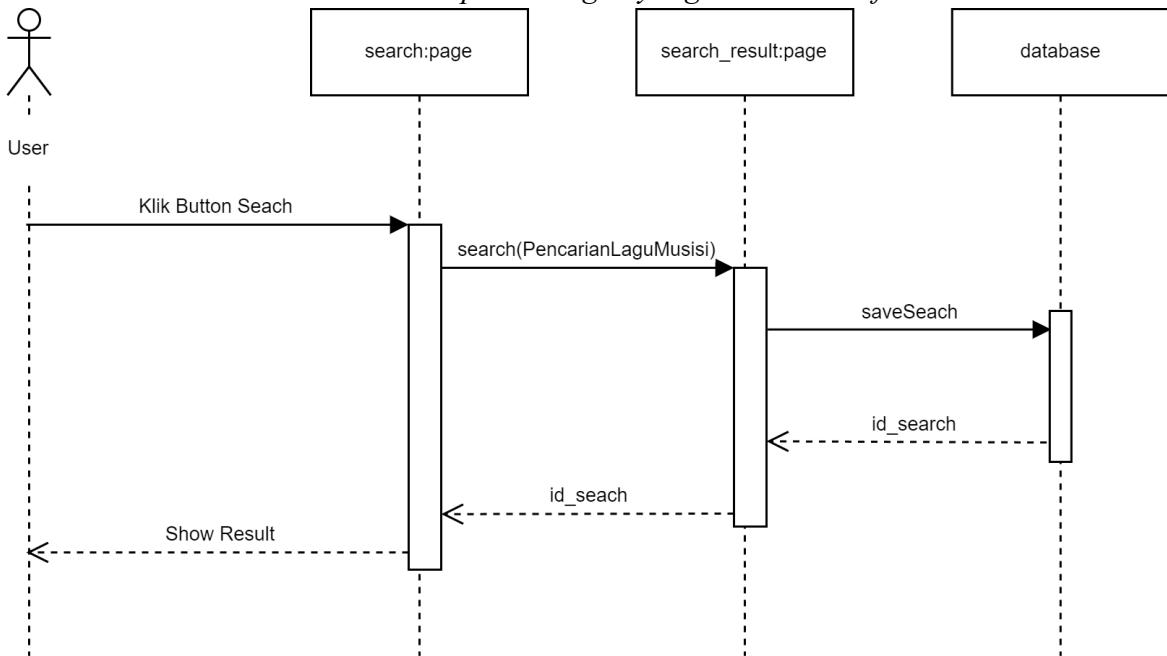
TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
1	searchpage	Boundary
2	searchresult	Controller
3	Database	Entity

\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

### 3.1.3.4 Sequence Diagram #3 Pencarian Lagu

Buatlah diagram sequence untuk masing - masing use case sesuai skenario.  
Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.



### 3.1.4 Use Case #4 Memutar Lagu

#### 3.1.4.1 Use Case Scenario #4 Memutar Lagu

Skenario Use Case #4 :

- i. Pre-Condition : User belum memilih lagu yang akan diputar
- ii. Use Case Description
  - a. Primary Flow

Aktor	Sistem
User memilih salah satu lagu dalam daftar lagu.	
	Sistem menampilkan lagu yang ingin diputar (Alternatif 2.1)

- b. Alternative Flow

2.1 Tidak ada Alternative Flow

- iii. Post-Condition : Sistem menampilkan dan memutar lagu yang dipilih

#### 3.1.4.2 UI Design dan Deskripsi Objek UI #4 Memutar Lagu

Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya.



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
004	Play Page	Halaman Play pada aplikasi

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil Antarmuka 004 {diisi dengan no. layar atau no gambar rancangan antarmuka}

Page Play Page

Id_Objek	JENIS	LABEL*	Keterangan**
txt_artists	Label	Nama Musisi	Label yang menampilkan nama musisi dari lagu yang ada pada halaman
txt_songs	Label	Judul Lagu	Label yang menampilkan judul lagu dari lagu yang ada pada halaman
btn_prev	Button	Icon	Jika di klik, akan mengaktifkan Function Previous
btn_play	Button	Icon	Jika di klik, akan mengaktifkan Function Play

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>btn_next</i>	<i>Button</i>	Icon	<i>Jika di klik, akan mengaktifkan Function Next</i>
<i>progressbar_lagu</i>	<i>Progress Bar</i>	-	<i>Progress Bar yang menampilkan informasi mengenai progress lagu pada lagu yang sudah terplay</i>
<i>btn_queue</i>	<i>Button</i>	Icon	<i>Jika di klik, akan mengaktifkan Function Queue</i>
<i>btn_volume</i>	<i>Button</i>	Icon	<i>Jika di klik, akan melakukan mute dan unmute pada lagu</i>
<i>slider_volume</i>	<i>Slider</i>	-	<i>Jika di slide, akan membesarkan dan mengecilkan lagu</i>

### 3.1.4.3 Identifikasi Object Baru & Tipe Kelas #4 Memutar Lagu

*Identifikasi object yang terkait dengan use case tersebut.*

*Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.*

*Gunakan tabel di bawah:*

TABEL OBJECT PERANCANGAN

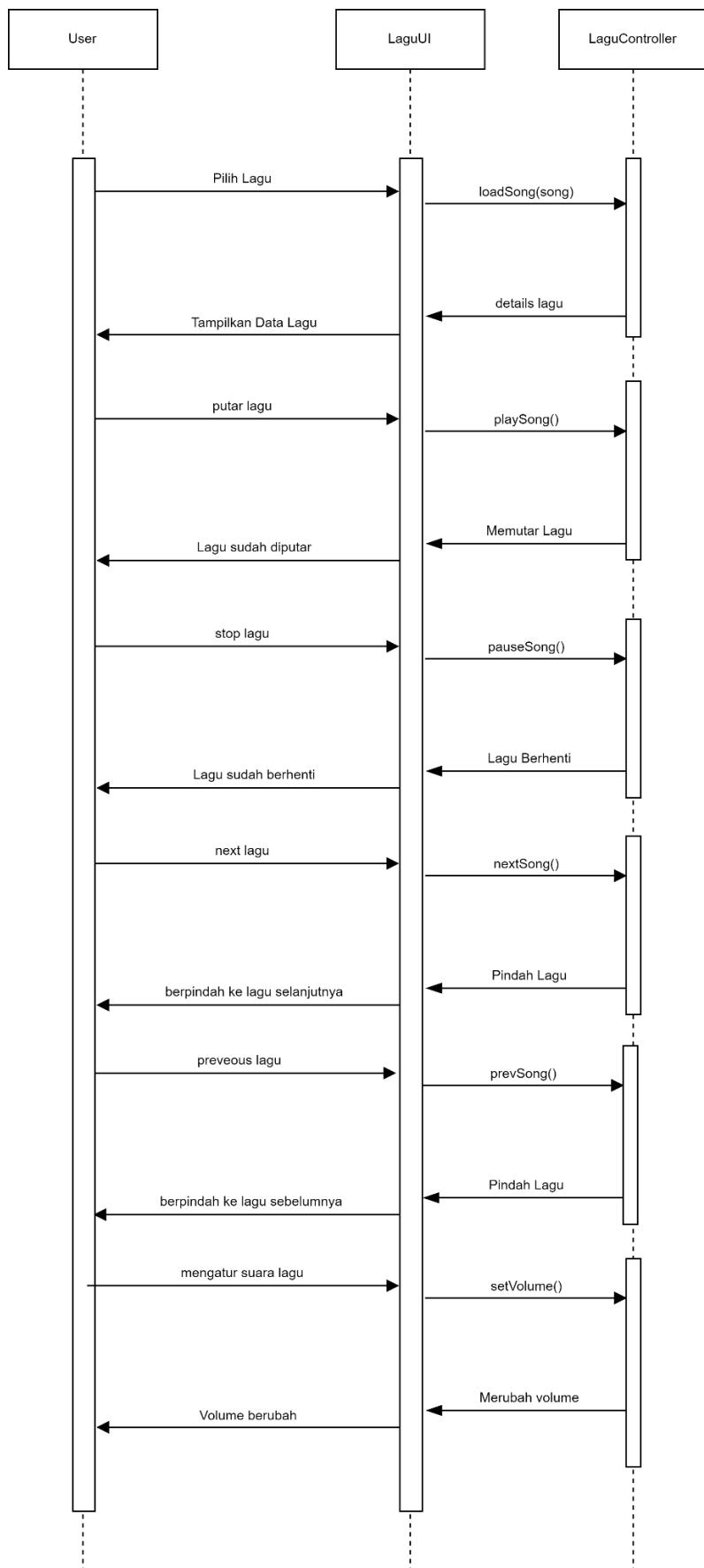
<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1	<i>User</i>	<i>Entity</i>
2	<i>Database</i>	<i>Entity</i>
3	<i>LaguUI</i>	<i>Boundary</i>
4	<i>LaguController</i>	<i>Controller</i>

*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

### 3.1.4.4 Sequence Diagram #4 Memutar Lagu

*Buatlah diagram sequence untuk masing - masing use case sesuai skenario.*

*Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.*



### 3.1.5 Use Case #5 Kelola Favourites

#### 3.1.5.1 Use Case Scenario #5 Kelola Favourites

Skenario Use Case #5 :

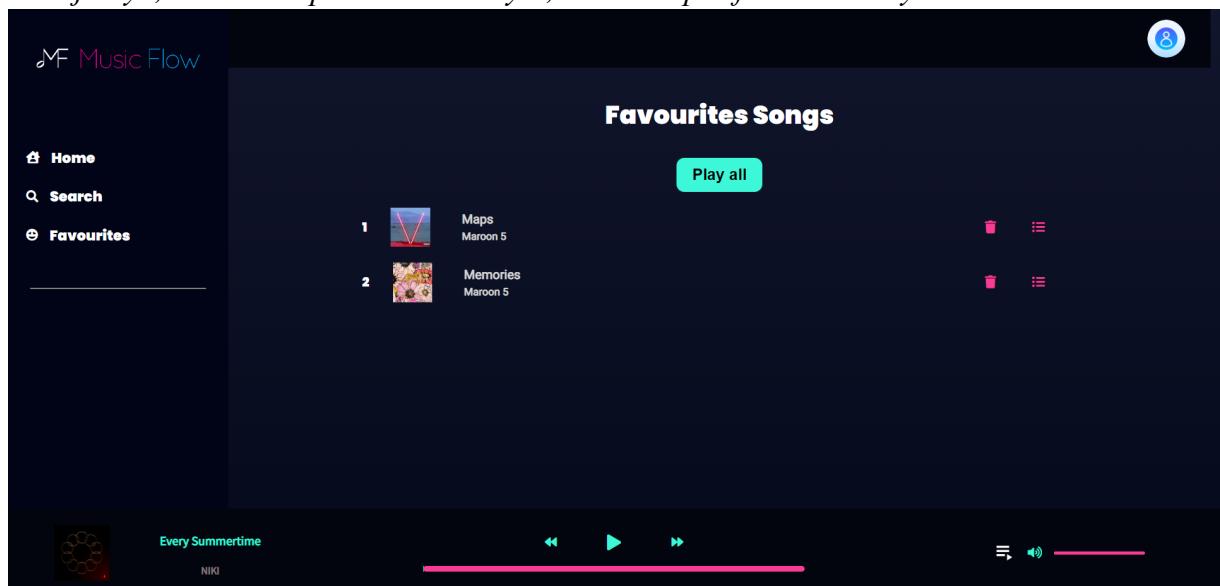
- i. Pre-Condition : User belum masuk ke halaman *Favourites*
- ii. Use Case Description
  - a. Primary Flow

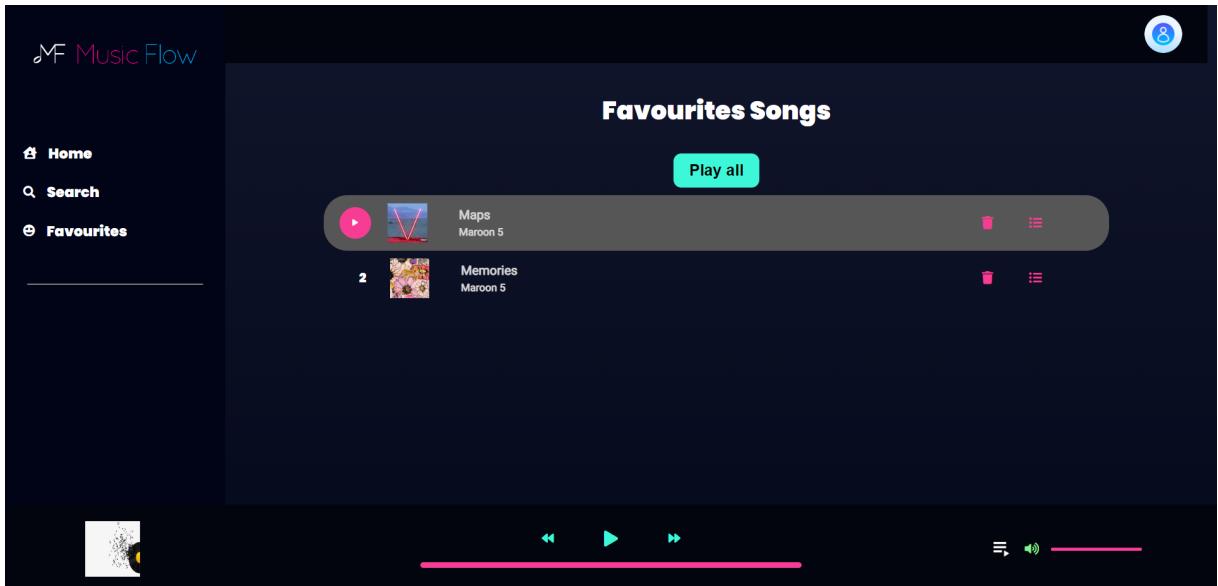
Aktor	Sistem
User memilih menu Favourites.	
	Sistem menampilkan halaman <i>Favourites</i> (Alternatif 2.1)

- b. Alternative Flow
  - 2.1 Tidak ada Alternative Flow
- iii. Post-Condition : User masuk ke dalam halaman *Favourites*

#### 3.1.5.2 UI Design dan Deskripsi Objek UI #5 Kelola Favourites

Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detailnya.





Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
005	Favourites Page	Halaman Favourites pada aplikasi

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detail  
*Antarmuka 005: {diisi dengan no. layar atau no gambar rancangan antarmuka}*

Page Favourites Page

Id_Objek	JENIS	LABEL*	Keterangan**
btn_play_all	Button	Play all	Jika di klik, akan memainkan seluruh lagu pada Favourites
btn_delete_fav_ourite	Button	Icon	Jika di klik, akan menghapus lagu dari Favourites
btn_queue	Button	Icon	Jika di klik, akan mengaktifkan Function Queue
btn_play	Button	Icon	Jika di klik, akan mengaktifkan Function Play
txt_number	Label	1 - dst (number)	Label yang menampilkan informasi urutan dengan angka
txt_artists	Label	Nama Musisi	Label yang menampilkan nama musisi
txt_songs	Label	Judul Lagu	Label yang menampilkan judul lagu
img_album	Image	-	Image dari cover album/lagu dari lagu

### 3.1.5.3 Identifikasi Object Baru & Tipe Kelas #5 Kelola Favourites

Identifikasi object yang terkait dengan use case tersebut.

Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.

Gunakan tabel di bawah:

TABEL OBJECT PERANCANGAN

No	Nama Object Baru	Jenis / Tipe Kelas
----	------------------	--------------------

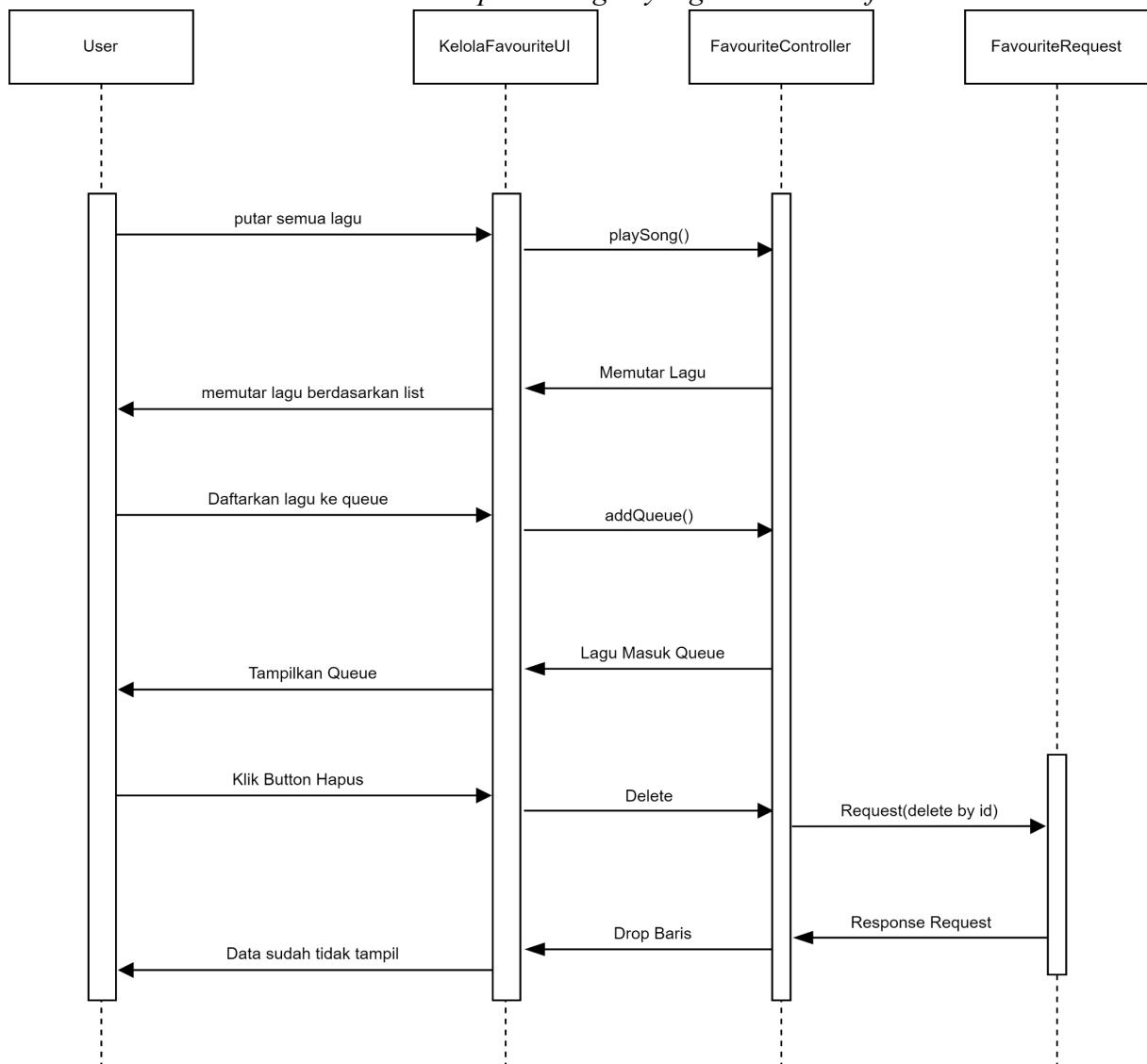
Prodi S1 Informatika Tel-U	DPPL-001	Halaman 24 dari 53
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

1	User	Entity
2	Database	Entity
3	KelolaFavouriteUI	Boundary
4	FavouriteController	Controller
5	FavouriteRequest	Controller

\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

### 3.1.5.4 Sequence Diagram #5 Kelola Favourites

Buatlah diagram sequence untuk masing - masing use case sesuai skenario.  
Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.



### 3.1.6 Use Case #6 Kelola Queue

#### 3.1.6.1 Use Case Scenario #6 Kelola Queue

Skenario Use Case #6 :

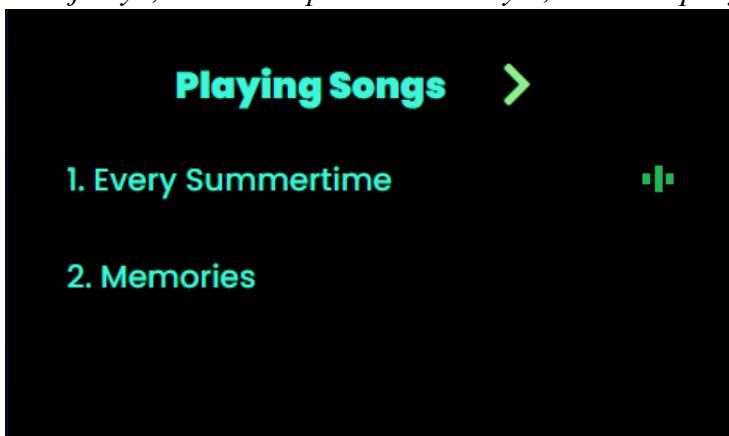
- iv. Pre-Condition : User belum masuk ke halaman *Queue*
- v. Use Case Description
  - a. Primary Flow

Aktor	Sistem
User memilih menu Queue.	
	Sistem menampilkan halaman <i>Queue</i> (Alternatif 2.1)

- b. Alternative Flow
  - 2.1 Tidak ada Alternative Flow
- vi. Post-Condition : User masuk ke dalam halaman *Queue*

#### 3.1.6.2 UI Design dan Deskripsi Objek UI #6 Kelola Queue

Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detailnya.



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
006	<i>Queue Page</i>	Halaman Queue pada aplikasi

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil  
*Antarmuka 006: {diisi dengan no. layar atau no gambar rancangan antarmuka}*

Page *Queue Page*

Prodi S1 Informatika Tel-U	DPPL-001	Halaman 26 dari 53
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>btn_play_queue</i>	<i>Button</i>	Urutan lagu pada Queue dan Judul Lagu	<i>Jika di klik, akan mengaktifkan Function Play</i>
<i>btn_back_queue</i>	<i>Button</i>	Icon	<i>Jika di klik, akan kembali ke halaman sebelumnya</i>

### 3.1.6.3 Identifikasi Object Baru & Tipe Kelas #6 Kelola Queue

*Identifikasi object yang terkait dengan use case tersebut.*

*Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.*

*Gunakan tabel di bawah:*

TABEL OBJECT PERANCANGAN

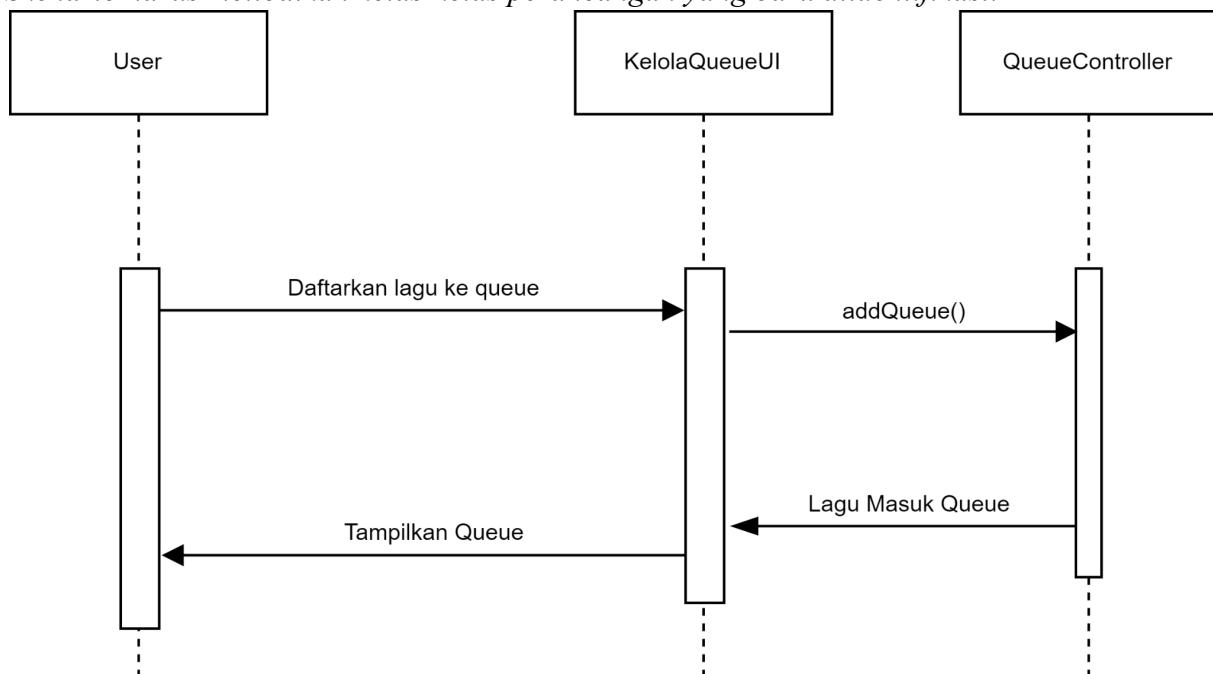
<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1	<i>User</i>	<i>Entity</i>
2	<i>Database</i>	<i>Entity</i>
3	<i>KelolaQueueUI</i>	<i>Boundary</i>
4	<i>QueueController</i>	<i>Controller</i>

*\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller*

### 3.1.6.4 Sequence Diagram #6 Kelola Queue

*Buatlah diagram sequence untuk masing - masing use case sesuai skenario.*

*Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.*



### 3.1.7 Use Case #7 Kelola User

#### 3.1.7.1 Use Case Scenario #7 Kelola User

Skenario Use Case #7 :

- i. Pre-Condition : Admin belum masuk ke halaman *Users Account Management*
- ii. Use Case Description
  - a. Primary Flow

Aktor	Sistem
Admin memilih menu kelola User	
	Sistem menampilkan halaman <i>Users Account Management</i> (Alternatif 2.1)

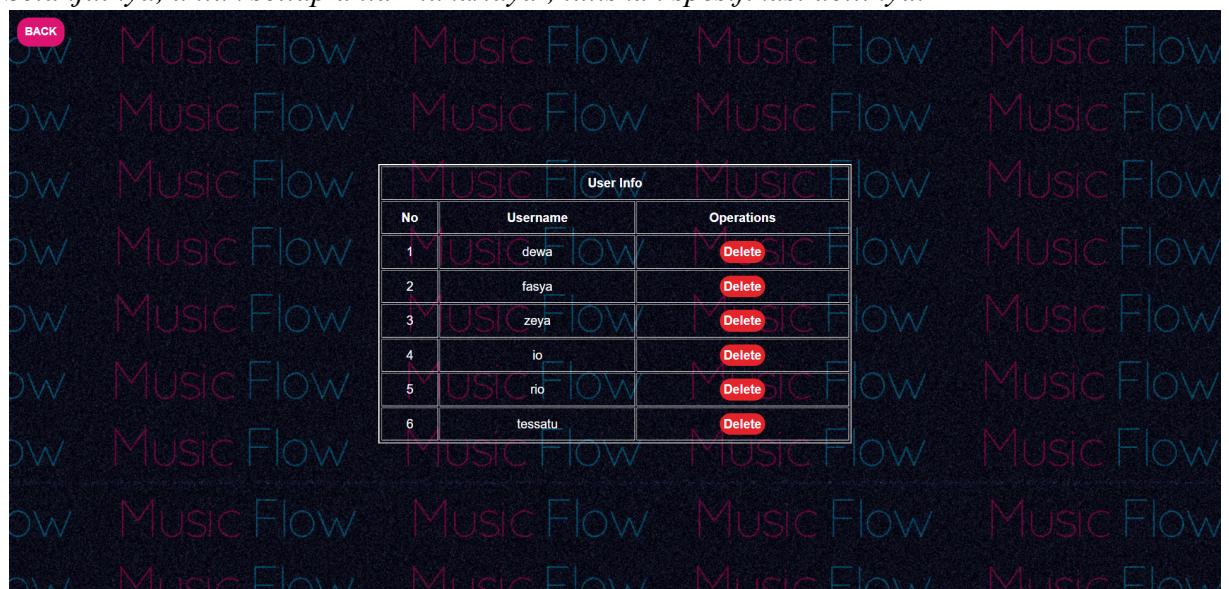
- b. Alternative Flow

2.1 Tidak ada Alternative Flow

- iii. Post-Condition : Admin masuk ke halaman *Users Account Management*

#### 3.1.7.2 UI Design dan Deskripsi Objek UI #7 Kelola User

Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya.



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
007	<i>Users Account Management Page</i>	Halaman Management Page ( <i>Users Account</i> ) pada aplikasi

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil

### Page Users Account Management Page

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>gridview_number</i>	<i>Grid View</i>	1 - dst (number)	<i>Grid View yang menampilkan informasi urutan dengan angka</i>
<i>gridview_username</i>	<i>Grid View</i>	Username	<i>Grid View yang menampilkan informasi berupa username</i>
<i>btn_delete</i>	<i>Button</i>	Delete	<i>Jika di klik, akan mengaktifkan Function Delete</i>
<i>btn_back</i>	<i>Button</i>	Back	<i>Jika di klik, akan mengaktifkan Function Back</i>

#### 3.1.7.3 Identifikasi Object Baru & Tipe Kelas #7 Kelola User

Identifikasi object yang terkait dengan use case tersebut.

Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.

Gunakan tabel di bawah:

TABEL OBJECT PERANCANGAN

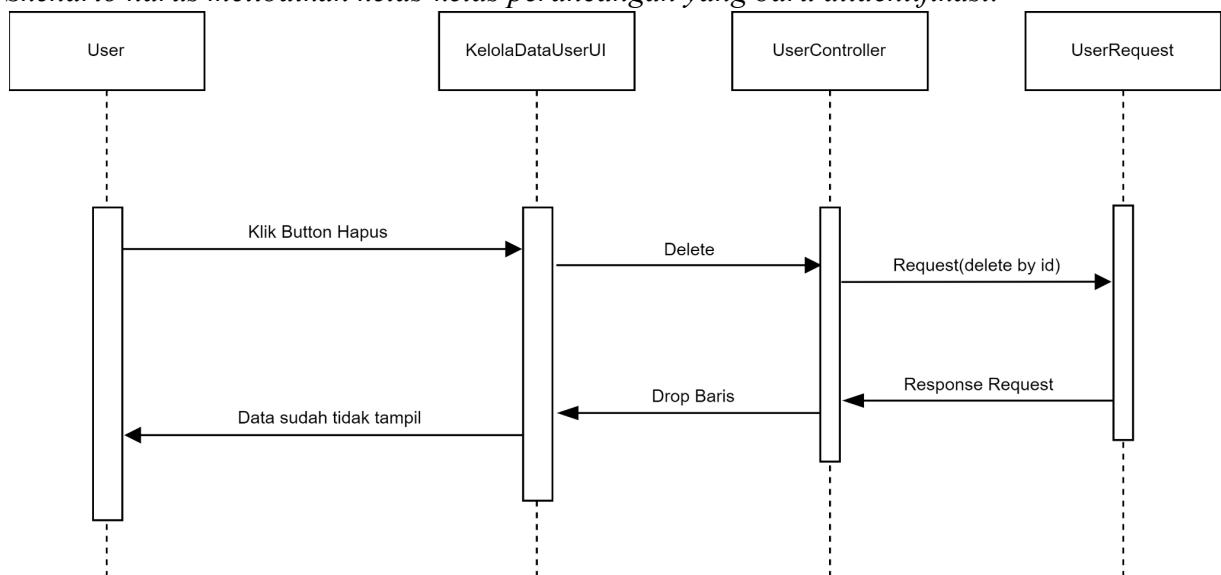
<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1	User	Entity
2	Database	Entity
3	KelolaDataUserUI	Boundary
4	UserController	Controller
5	UserRequest	Controller

\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

#### 3.1.7.4 Sequence Diagram #7 Kelola User

Buatlah diagram sequence untuk masing - masing use case sesuai skenario.

Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.



### 3.1.8 Use Case #8 Kelola Lagu

#### 3.1.8.1 Use Case Scenario #8 Kelola Lagu

Skenario Use Case #8 :

- i. Pre-Condition : Admin belum masuk ke halaman *Songs Management Page*
- ii. Use Case Description
  - a. Primary Flow

Aktor	Sistem
Admin memilih menu kelola Lagu.	
	Sistem menampilkan halaman <i>Songs Management Page</i>
Admin melakukan tambah lagu dengan button “INSERT SONG”	
	Sistem menampilkan halaman <i>Songs Update Page</i>
Admin mengisi form pengisian data lagu, lalu klik “SUBMIT”.	
	Sistem melakukan tambah data lagu dan menampilkan halaman <i>Songs Management Page</i> (Alternatif 2.1)
Admin melakukan hapus data lagu, lalu klik “DELETE”	
	Sistem melakukan delete data lagu
Admin melakukan edit data Lagu, lalu klik “UPDATE”	
	Sistem menampilkan halaman <i>Songs Update Page</i>
Admin mengisi form pengisian data lagu, lalu klik “SUBMIT”.	
	Sistem melakukan edit data lagu dan menampilkan halaman <i>Songs Management Page</i> (Alternatif 2.2)

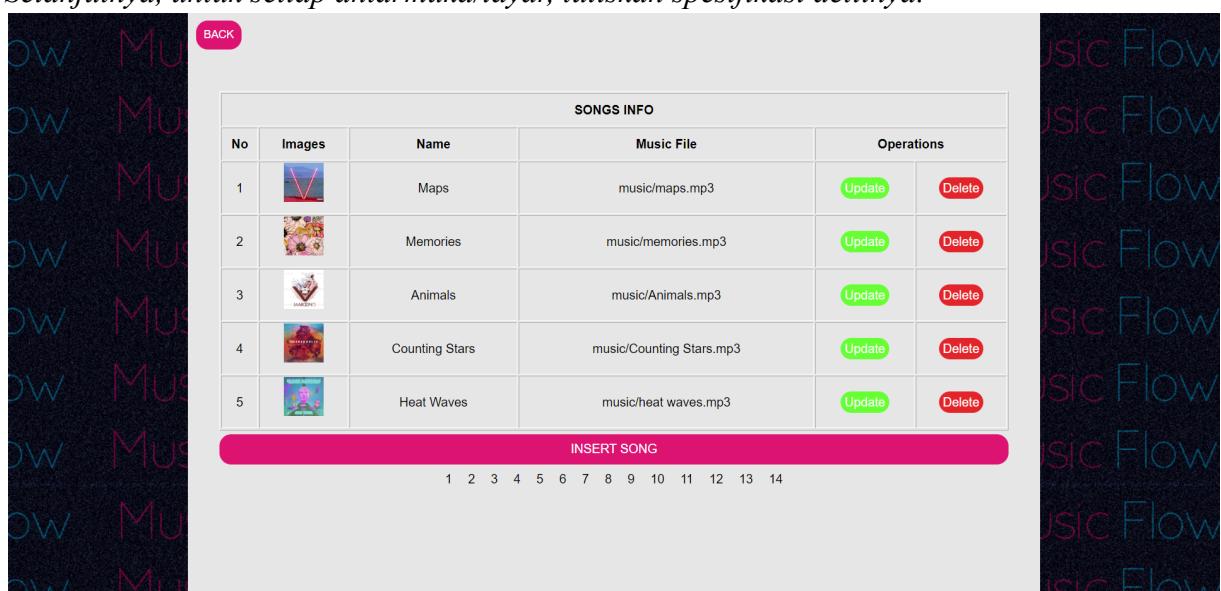
- b. Alternative Flow

- 2.1 Admin tidak mengisi form tambah data lagu, lalu melakukan klik “SUBMIT”, maka Sistem akan memberikan alert “Title cannot be empty” “Music File cannot be empty” “Image file cannot be empty”
- 2.2 Admin mengedit nama lagu atau atribut yang lain dengan cara mengosongkan isi data nya, maka Sistem akan memberikan alert “Music File cannot be empty” “Image file cannot be empty”

iii. Post-Condition : Admin masuk ke halaman *Songs Update Page*

### 3.1.8.2 UI Design dan Deskripsi Objek UI #8 Kelola Lagu

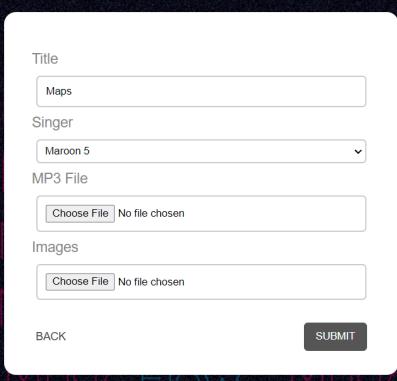
*Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detilnya.*



SONGS INFO					
No	Images	Name	Music File	Operations	
1		Maps	music/maps.mp3	<button>Update</button>	<button>Delete</button>
2		Memories	music/memories.mp3	<button>Update</button>	<button>Delete</button>
3		Animals	music/Animals.mp3	<button>Update</button>	<button>Delete</button>
4		Counting Stars	music/Counting Stars.mp3	<button>Update</button>	<button>Delete</button>
5		Heat Waves	music/heat waves.mp3	<button>Update</button>	<button>Delete</button>

**INSERT SONG**

1 2 3 4 5 6 7 8 9 10 11 12 13 14

Title

Singer

MP3 File

Images

[BACK](#) [SUBMIT](#)

Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
008	Songs Management Page	Halaman Songs Management Page pada aplikasi
009	Songs Update Page	Halaman Songs Update Page pada aplikasi

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil  
*Antarmuka 008 dan 009: {diisi dengan no. layar atau no gambar rancangan antarmuka}*

#### Page Songs Management Page

Id_Objek	JENIS	LABEL*	Keterangan**
gridview_number	Grid View	1 - dst (number)	Grid View yang menampilkan informasi urutan dengan angka
gridview_songs_image	Grid View	-	Grid View yang menampilkan informasi berupa gambar album dari lagu
gridview_songs_name	Grid View	Judul Lagu	Grid View yang menampilkan informasi berupa judul lagu
gridview_songs_file	Grid View	Location File	Grid View yang menampilkan informasi berupa location file dari lagu
btn_update	Button	Update	Jika di klik, akan mengaktifkan Function Update
btn_delete	Button	Delete	Jika di klik, akan mengaktifkan Function Delete
btn_back	Button	Back	Jika di klik, akan mengaktifkan Function Back
btn_insertsong	Button	INSERT SONG	Jika di klik, akan membuka halaman Songs Update Page
btn_pages	Button	1 - dst (number)	Jika di klik, akan menuju halaman lain dari Songs Management Page

#### Page Songs Update Page

Id_Objek	JENIS	LABEL*	Keterangan**
txt_songs_name	Text Field editable	Title	Text Field yang digunakan untuk mengisi title dari sebuah lagu
drop_artist	Drop-down-list	Nama Musisi	Jika di klik, akan menampilkan daftar musisi dalam bentuk Drop-down-list
choose_songs_files	Button	Choose File	Jika di klik, akan membuka file explorer untuk input file lagu
choose_songs_images	Button	Choose File	Jika di klik, akan membuka file explorer untuk input cover album
btn_back	Button	Back	Jika di klik, akan mengaktifkan Function Back

Prodi S1 Informatika Tel-U	DPPL-001	Halaman 32 dari 53
Template dokumen ini dan informasi yang dimilikinya adalah milik Prodi S1 Informatika Tel-U dan bersifat rahasia. Dilarang me-reproduksi dokumen ini tanpa diketahui oleh Prodi S1 Informatika Tel-U.		

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>btn_submit</i>	Button	Submit	<i>Jika di klik, akan mensubmit perubahan yang ada</i>

### 3.1.8.3 Identifikasi Object Baru & Tipe Kelas #8 Kelola Lagu

*Identifikasi object yang terkait dengan use case tersebut.*

*Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.*

*Gunakan tabel di bawah:*

TABEL OBJECT PERANCANGAN

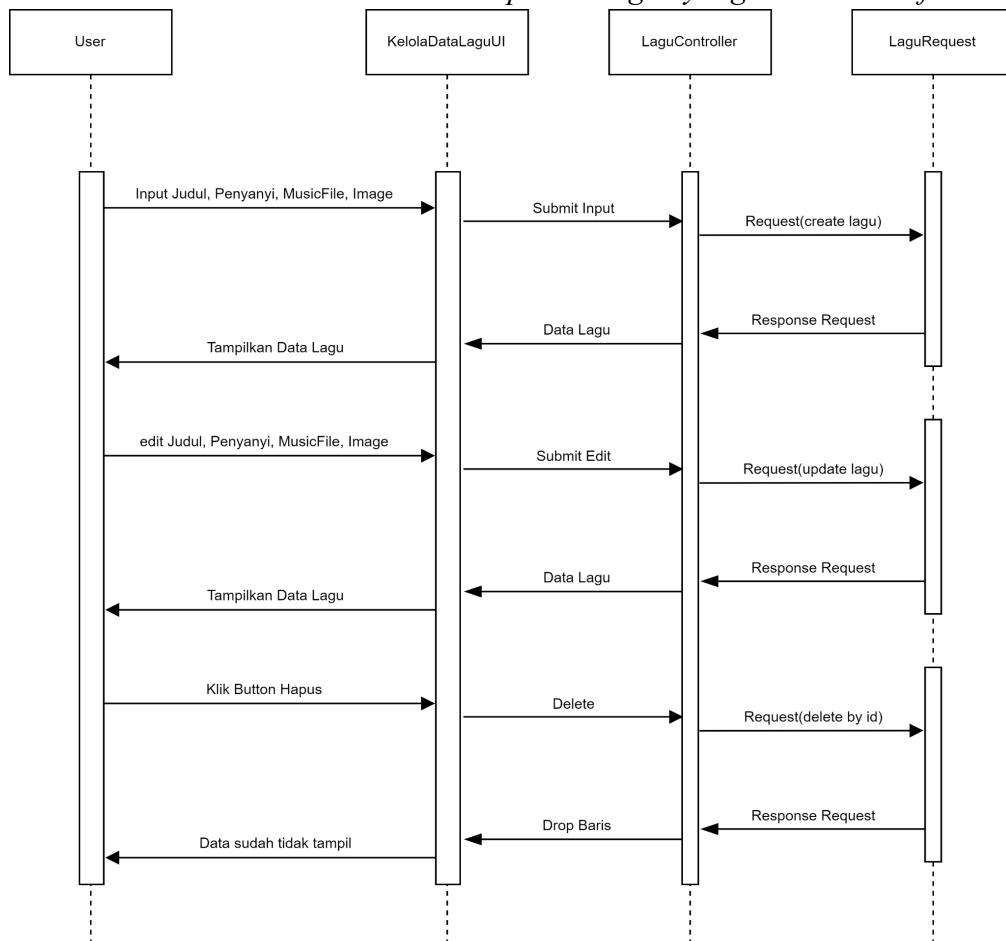
No	Nama Object Baru	Jenis / Tipe Kelas
1	User	Entity
2	Database	Entity
3	KelolaDataLaguUI	Boundary
4	LaguController	Controller
5	LaguRequest	Controller

\*Tipe kelas seperti Boundary(Interface), Entity(Database), Controller

### 3.1.8.4 Sequence Diagram #8 Kelola Lagu

*Buatlah diagram sequence untuk masing - masing use case sesuai skenario.*

*Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.*



### 3.1.9 Use Case #9 Kelola Musisi

#### 3.1.9.1 Use Case Scenario #9 Kelola Musisi

Skenario Use Case #9 :

- i. Pre-Condition : Admin belum masuk ke halaman *Artists Management Page*
- ii. Use Case Description
- iii. Primary Flow

Aktor	Sistem
Admin memilih menu kelola Musisi.	
	Sistem menampilkan halaman <i>Artists Management Page</i>
Admin melakukan tambah musisi	
	Sistem menampilkan halaman <i>Artists Update Page</i>
Admin mengisi form pengisian data musisi, lalu klik “SUBMIT”.	
	Sistem melakukan tambah data musisi dan menampilkan halaman <i>Artists Management Page</i> (Alternatif 2.1)
Admin melakukan hapus data musisi, lalu klik “DELETE”	
	Sistem melakukan delete data musisi
Admin melakukan edit data musisi, lalu klik “UPDATE”	
	Sistem menampilkan halaman <i>Artists Update Page</i>
Admin mengisi form pengisian data musisi, lalu klik “SUBMIT”.	
	Sistem melakukan edit musisi dan menampilkan halaman <i>Artists Management Page</i> (Alternatif 2.2)

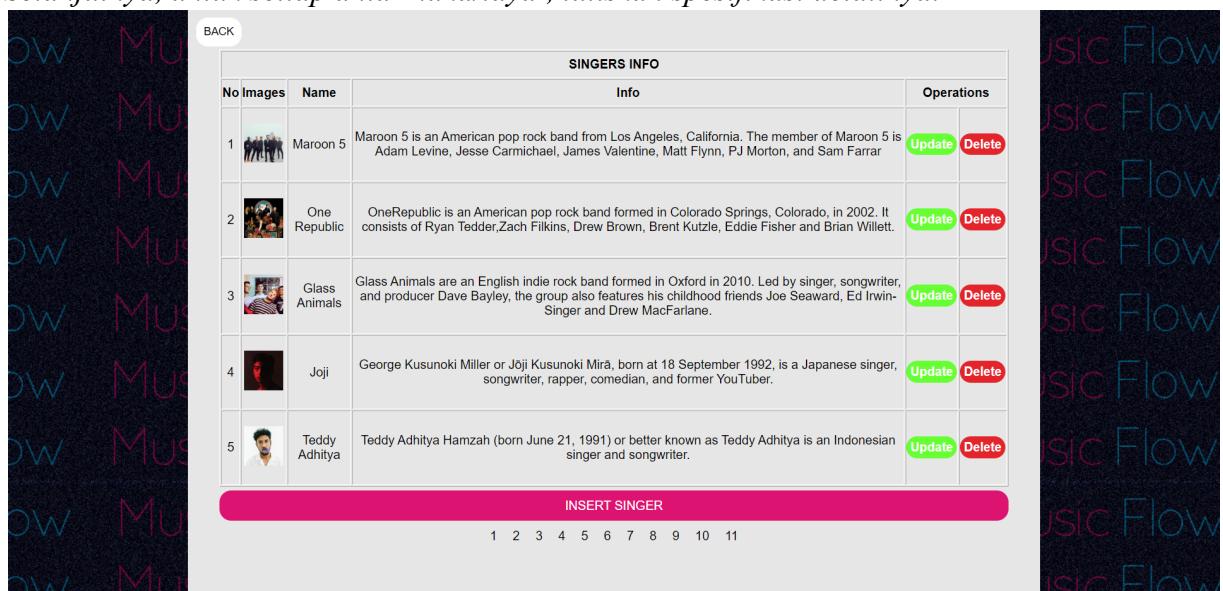
- iv. Alternative Flow

- 2.1 Admin tidak mengisi form tambah data musisi, lalu melakukan klik “SUBMIT”, maka Sistem akan memberikan alert “Singer's name can not be empty” “Info can not be empty” “Image field cannot be empty”
- 2.2 Admin mengedit nama musisi atau atribut yang lain dengan cara mengosongkan isi data nya, maka Sistem akan memberikan alert “Singer's name can not be empty” “Info can not be empty” “Image field cannot be empty”

v. Post-Condition : Admin masuk ke dalam halaman *Artists Update Page*

### 3.1.9.2 UI Design dan Deskripsi Objek UI #9 Kelola Penyanyi

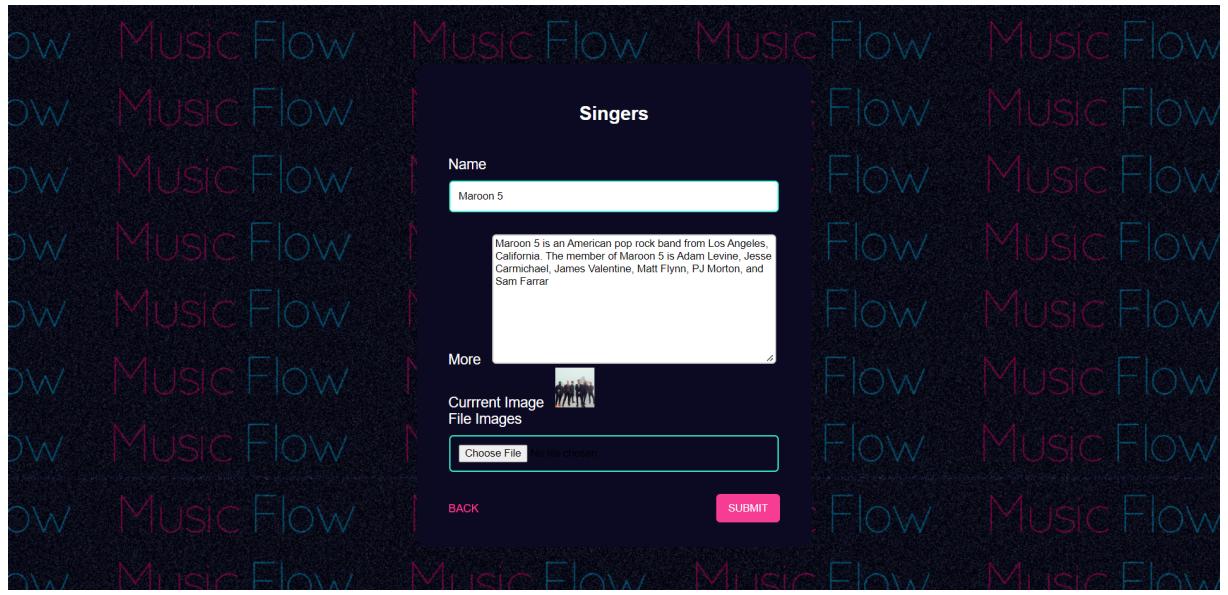
*Bagian ini diisi dengan versi awal prototipe antarmuka untuk per Use Case Selanjutnya, untuk setiap antarmuka/layar, tuliskan spesifikasi detailnya.*



No	Images	Name	Info	Operations	
				Update	Delete
1		Maroon 5	Maroon 5 is an American pop rock band from Los Angeles, California. The member of Maroon 5 is Adam Levine, Jesse Carmichael, James Valentine, Matt Flynn, PJ Morton, and Sam Farrar	<a href="#">Update</a>	<a href="#">Delete</a>
2		One Republic	OneRepublic is an American pop rock band formed in Colorado Springs, Colorado, in 2002. It consists of Ryan Tedder, Zach Filkins, Drew Brown, Brent Kutzle, Eddie Fisher and Brian Willett.	<a href="#">Update</a>	<a href="#">Delete</a>
3		Glass Animals	Glass Animals are an English indie rock band formed in Oxford in 2010. Led by singer, songwriter, and producer Dave Bayley, the group also features his childhood friends Joe Seaward, Ed Irwin-Singer and Drew MacFarlane.	<a href="#">Update</a>	<a href="#">Delete</a>
4		Joji	George Kusunoki Miller or Jōji Kusunoki Mirā, born at 18 September 1992, is a Japanese singer-songwriter, rapper, comedian, and former YouTuber.	<a href="#">Update</a>	<a href="#">Delete</a>
5		Teddy Adhitya	Teddy Adhitya Hamzah (born June 21, 1991) or better known as Teddy Adhitya is an Indonesian singer and songwriter.	<a href="#">Update</a>	<a href="#">Delete</a>

**INSERT SINGER**

1 2 3 4 5 6 7 8 9 10 11



Tabel Deskripsi Objek UI

ID. LAYAR	NAMA LAYAR	DESKRIPSI
010	<i>Artists Management Page</i>	<i>Halaman Artists Management Page pada aplikasi</i>
011	<i>Artists Update Page</i>	<i>Halaman Artists Update Page pada aplikasi</i>

UNTUK MASING – MASING ANTAR MUKA / PAGE dibuatkan spesifikasi detil  
*Antarmuka 010 dan 011: {diisi dengan no. layar atau no gambar rancangan antarmuka}*

Page *Artists Management Page*

Id_Objek	JENIS	LABEL*	Keterangan**
gridview_number	Grid View	1 - dst (number)	<i>Grid View yang menampilkan informasi urutan dengan angka</i>
gridview_artists_image	Grid View	-	<i>Grid View yang menampilkan informasi berupa gambar musisi</i>
gridview_artists_name	Grid View	Nama Penyanyi	<i>Grid View yang menampilkan informasi berupa nama musisi</i>
gridview_artists_info	Grid View	Informasi Musisi	<i>Grid View yang menampilkan informasi berupa deskripsi mengenai musisi</i>
btn_update	Button	Update	<i>Jika di klik, akan mengaktifkan Function Update</i>
btn_delete	Button	Delete	<i>Jika di klik, akan mengaktifkan Function Delete</i>
btn_back	Button	Back	<i>Jika di klik, akan mengaktifkan Function Back</i>
btn_insertartist	Button	INSERT SINGER	<i>Jika di klik, akan membuka halaman Artists Update Page</i>

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>btn_pages</i>	Button	1 - dst (number)	<i>Jika di klik, akan menuju halaman lain dari Artists Management Page</i>

Page Artists Update Page

<b>Id_Objek</b>	<b>JENIS</b>	<b>LABEL*</b>	<b>Keterangan**</b>
<i>txt_artists_name</i>	<i>Text Field editable</i>	<i>Singer Name</i>	<i>Text Field yang digunakan untuk mengisi nama musisi</i>
<i>txt_artists_info</i>	<i>Text Field editable</i>	<i>Singer Info</i>	<i>Text Field yang digunakan untuk mengisi informasi deskripsi dari musisi</i>
<i>choose_artists_images</i>	Button	<i>Choose File</i>	<i>Jika di klik, akan membuka file explorer untuk input gambar musisi</i>
<i>btn_back</i>	Button	<i>Back</i>	<i>Jika di klik, akan mengaktifkan Function Back</i>
<i>btn_submit</i>	Button	<i>Submit</i>	<i>Jika di klik, akan mensubmit perubahan yang ada</i>

### 3.1.9.3 Identifikasi Object Baru & Tipe Kelas #9 Kelola Penyanyi

*Identifikasi object yang terkait dengan use case tersebut.*

*Kelas pada tahap perancangan berbeda dengan kelas pada tahap analisis.*

*Gunakan tabel di bawah:*

TABEL OBJECT PERANCANGAN

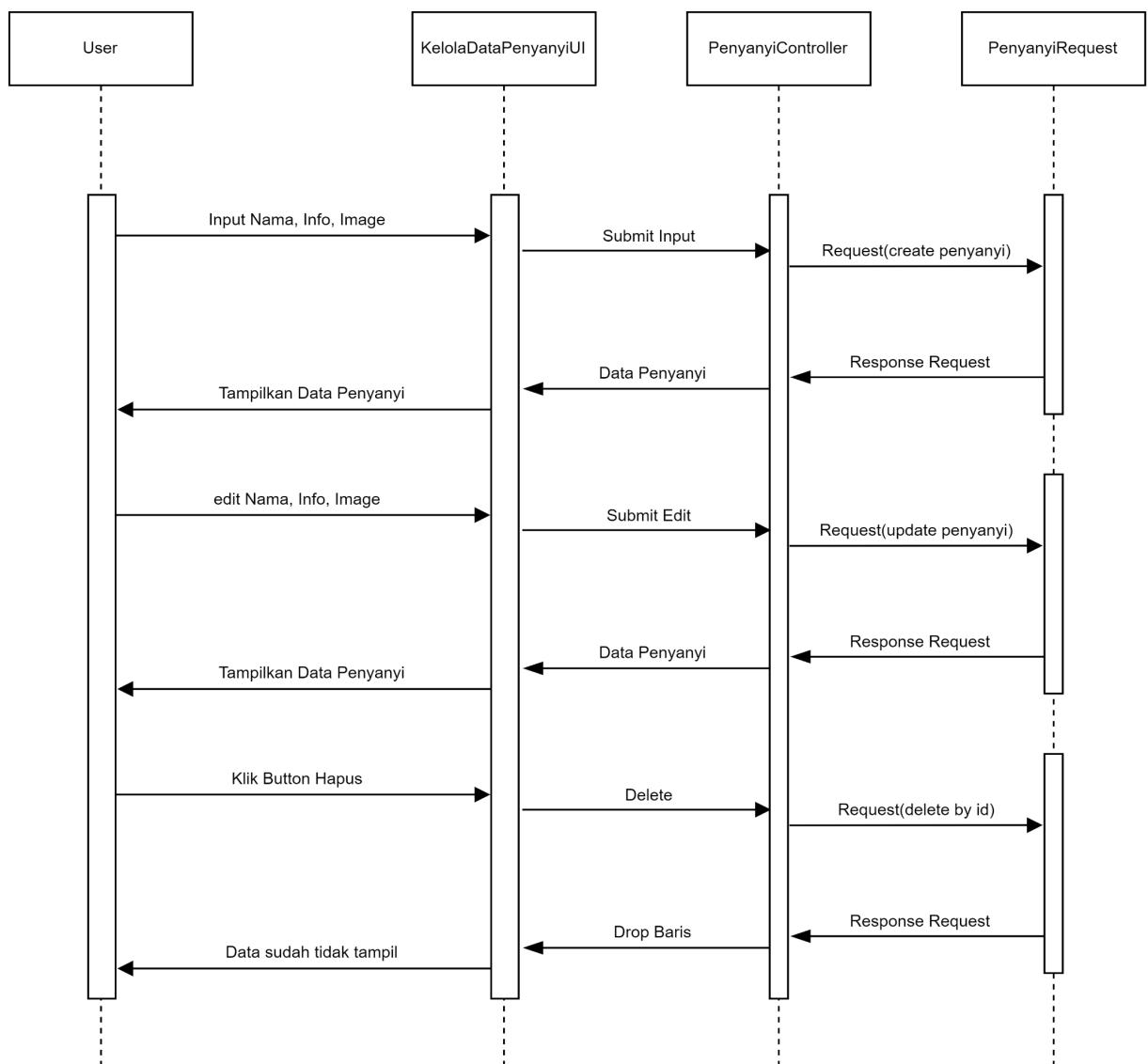
<b>No</b>	<b>Nama Object Baru</b>	<b>Jenis / Tipe Kelas</b>
1	<i>User</i>	<i>Entity</i>
2	<i>Database</i>	<i>Entity</i>
3	<i>KelolaDataPenyanyiUI</i>	<i>Boundary</i>
4	<i>PenyanyiController</i>	<i>Controller</i>
5	<i>PenyanyiRequest</i>	<i>Controller</i>

\*Tipe kelas seperti *Boundary(Interface)*, *Entity(Database)*, *Controller*

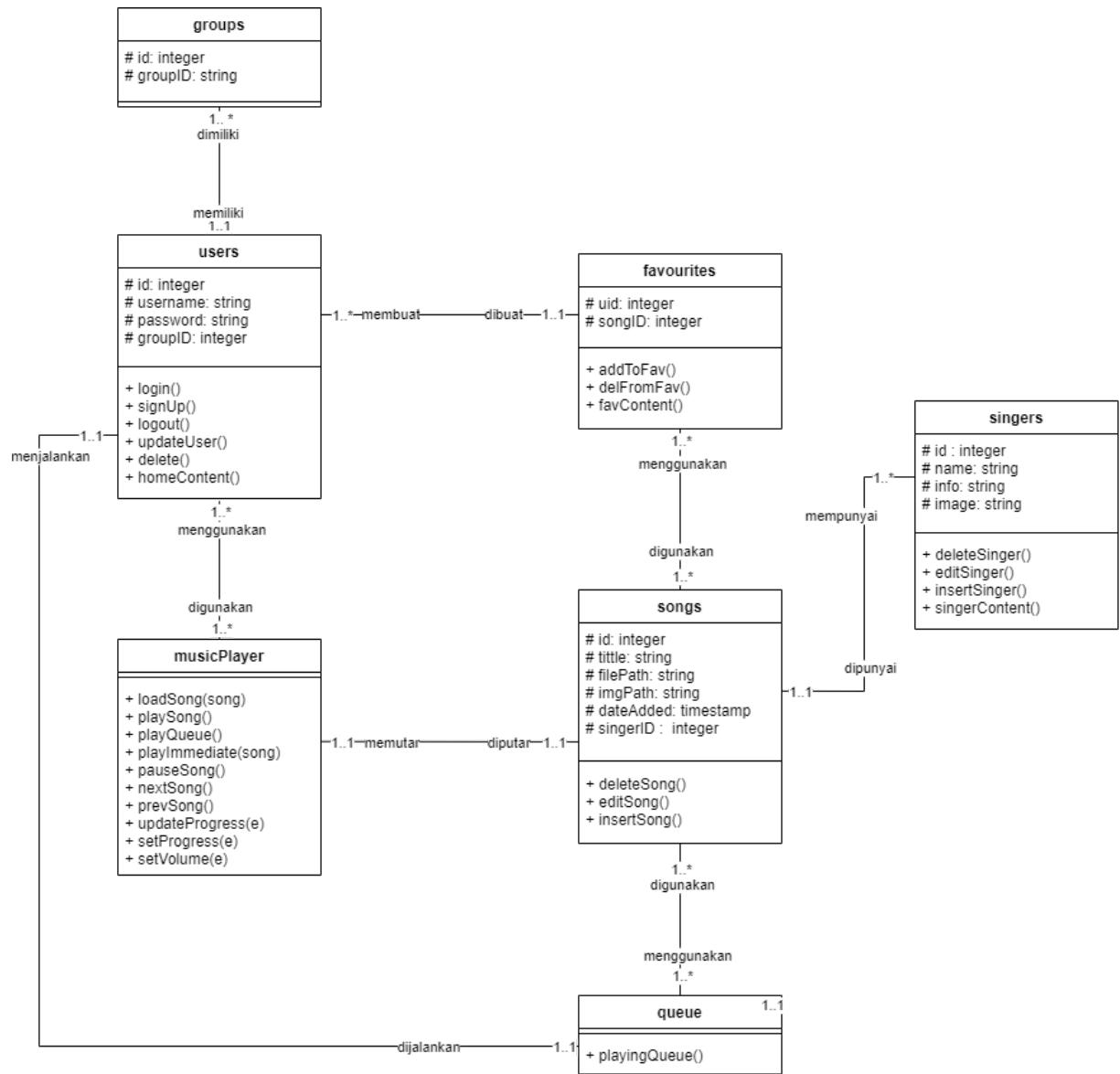
### 3.1.9.4 Sequence Diagram #9 Kelola Penyanyi

*Buatlah diagram sequence untuk masing - masing use case sesuai skenario.*

*Skenario harus melibatkan kelas-kelas perancangan yang baru diidentifikasi.*



## 3.2 Diagram Kelas Keseluruhan



Bagian ini diisi dengan diagram kelas keseluruhan yang akan digunakan dalam PL menggunakan model MVC

**Perancangan Detail Kelas:**

No	Nama Kelas Perancangan
1	users
2	groups
3	favourites

4	songs
5	singers
6	musicPlayer
7	queue

TABEL KELAS users :

Nama Atribut	Visibility	Tipe
id	protected	integer
username	protected	string
password	protected	string
groupID	protected	integer

Nama Operasi	Visibility
login()	public
signUp()	public
logout()	public
updateUser()	public
delete()	public
homeContent()	public

TABEL KELAS groups :

Nama Atribut	Visibility	Tipe
id	protected	integer
groupID	protected	string

TABEL KELAS favourites :

Nama Atribut	Visibility	Tipe
uid	protected	integer
songID	protected	integer

Nama Operasi	Visibility
addToFav()	public
delFromFav()	public
favContent()	public

TABEL KELAS songs :

Nama Atribut	Visibility	Tipe
id	protected	integer
title	protected	string

filePath	protected	string
imgPath	protected	string
dateAdded	protected	timestamp
singerID	protected	integer

Nama Operasi	Visibility
deleteSong()	public
editSong()	public
insertSong()	public

TABEL KELAS singers :

Nama Atribut	Visibility	Tipe
id	protected	integer
name	protected	string
info	protected	string
image	protected	string

Nama Operasi	Visibility
deleteSinger()	public
editSinger()	public
insertSinger()	public
singerContent()	public

TABEL KELAS mediaPlayer :

Nama Operasi	Visibility
loadSong(song)	public
playSong()	public
playQueue()	public
playImmediate(song)	public
pauseSong()	public
nextSong()	public
prevSong()	public
updateProgress(e)	public
setProgress(e)	public
setVolume(e)	public

TABEL KELAS queue :

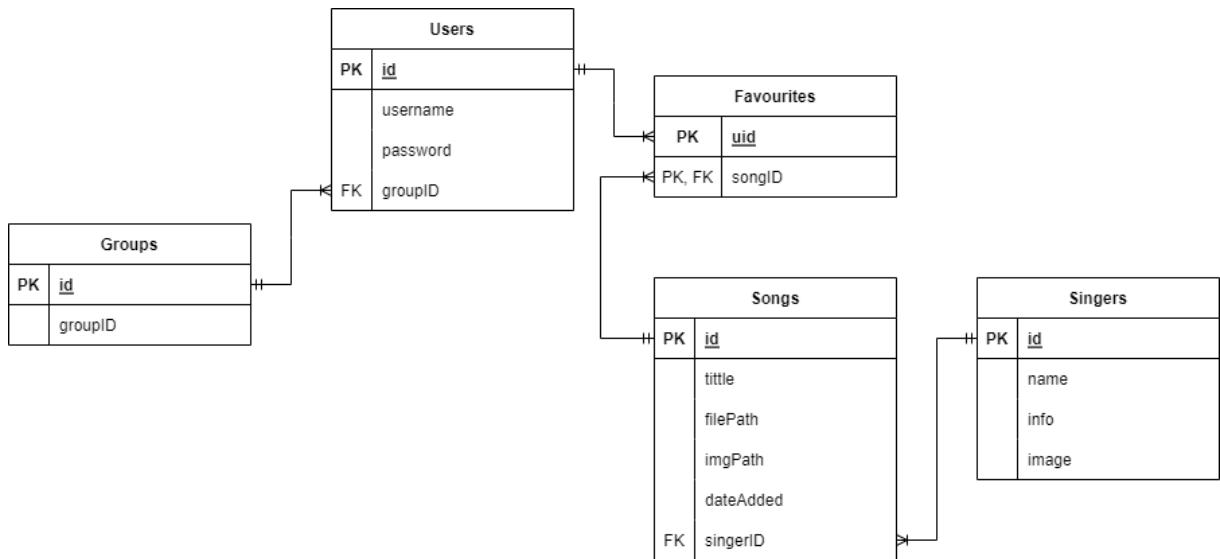
Nama Operasi	Visibility
playingQueue()	public

Untuk setiap kelas:

- identifikasi operasi (mengacu pada tanggung-jawab kelas), identifikasi atribut, termasuk visibility-nya

### 3.3 Perancangan Data / Basis Data

Bagian ini diisi dengan penjelasan rancangan dan rencana implementasi semua Basis Data yang muncul di Diagram Kelas



### 3.4 Perancangan Algoritma dan/atau Query

Bagian ini hanya diisi untuk kerangka algoritma untuk proses-proses yang dianggap cukup penting. Implementasi skeleton code juga sudah dapat dilakukan untuk kelas-kelas yang terdefinisi pada bahasa pemrograman tertentu. Boleh dibuat sub bab per kelas.

#### Algoritma #1

Contoh:

Nama Kelas : User

Nama Operasi : Login

Algoritma :

```
$username = $password = "";
$errors = array('username' => '', 'password' => '');

if (isset($_POST['submit'])) {
    function cleanData($data)
    {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    $username = cleanData($_POST['username']);
    $password = cleanData($_POST['password']);

    if (empty($username)) {
```

```

        $errors['username'] = "Username cant be empty!";
    }

    if (empty($password)) {
        $errors['password'] = "Password cant be empty!";
    } else {
        $password = md5($password);
        $sql = "SELECT * FROM users WHERE username = '$username' AND password = '$password'";
        $result = mysqli_query($conn, $sql);
        if (mysqli_num_rows($result) === 1) {
            $row = mysqli_fetch_assoc($result);
            if ($row['username'] === $username && $row['password'] === $password) {
                $_SESSION['username'] = $row['username'];
                $_SESSION['name'] = $row['name'];
                $_SESSION['id'] = $row['id'];
                $_SESSION['admin'] = ($row['groupID'] == 1) ? true : false;

                header("Location: ../index.php");
                exit();
            } else {
                $errors['password'] = "Incorrect username or password!";
            }
        } else {
            $errors['username'] = "Incorrect username or password!";
        }
    }
}

```

## Algoritma #2

Contoh:

*Nama Kelas : User*

*Nama Operasi : Register*

*Algoritma :*

```

$username = $password = $re_password = "";
$errors = array('username' => '', 'password' => '', 're_password' => '',
"matchPass" => "", "existUser" => "");
if (isset($_POST['submit'])) {
    function cleanData($data)

```

```

    {
        $data = trim($data);
        $data = stripslashes($data);
        $data = htmlspecialchars($data);
        return $data;
    }

    $username = cleanData($_POST['username']);
    $password = cleanData($_POST['password']);

    $re_password = cleanData($_POST['re_password']);

    if (empty($username)) {
        $errors['username'] = "Username can not be empty !";
    }
    if (empty($password)) {
        $errors['password'] = "Password can not be empty !";
    }
    if (empty($re_password)) {
        $errors['re_password'] = "Confirm password can not be empty !";
    }
    if ($password !== $re_password) {
        $errors['matchPass'] = "The confirmation password does not match !";
    }

    $sql = "SELECT * FROM users WHERE username = '$username'";
    $result = mysqli_query($conn, $sql);
    if (mysqli_num_rows($result) > 0) {
        $errors['existUser'] = "User is already exist !";
    }

    if (array_filter($errors)) {
        echo "Have errors";
    } else {
        $password = md5($password);
        $sql2 = "INSERT INTO users(username, password, groupID) VALUE ('$username', '$password', 2)";
        $result2 = mysqli_query($conn, $sql2);
        if ($result2) header("Location: login.php");
    }
}

```

### Algoritma #3

Contoh:

*Nama Kelas : Kelola Data Lagu*

*Nama Operasi : insertSong*

*Algoritma :*

```
$getSingers = "SELECT * from singers";
$result = mysqli_query($conn, $getSingers);
$singers = mysqli_fetch_all($result, MYSQLI_ASSOC);

// Update singer
$titleUpdate = $singerIDfff = "";

if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql2 = "SELECT * FROM songs WHERE id= '$id' ";
    $res2 = mysqli_query($conn, $sql2);
    $data = mysqli_fetch_array($res2);
    $titleUpdate = $data["title"];
    $singerIDfff = $data["singerID"];
}

$errors = array('title' => '', 'mp3' => '', 'img' => '');
$title = $mp3 = $img = $singerID = '';

// Save file to sv
function saveFile($fileInfo)
{
    $filename = $fileInfo['name'];
    $type = $fileInfo['type'];
    $folder = (strpos($type, "image") !== false) ? 'images/' :
'music/';

    $tmpPath = $fileInfo['tmp_name'];
    $destinationPath = $folder . $filename;

    if (move_uploaded_file($tmpPath, '../' . $destinationPath)) {
        echo "Successfully uploaded";
    } else {
        echo "Upload fail";
    }
}
```

```

        return $destinationPath;
    }

    // Handle submit
    if (isset($_POST['submit'])) {
        // Title
        if (empty($_POST['title'])) {
            $errors['title'] = "Title cannot be empty";
        } else {
            $title = $_POST['title'];
        }

        // Singer
        $singerID = $_POST['singer'];

        // File mp3
        if (empty($_FILES["mp3"]["name"])) {
            $errors['mp3'] = "Music File cannot be empty";
        } else {
            $mp3 = $_FILES['mp3'];
        }

        // File image
        if (empty($_FILES["image"]["name"])) {
            $errors['image'] = "Image file cannot be empty";
        } else {
            $img = $_FILES['image'];
        }

        // Checking for errors
        if (array_filter($errors)) {
            echo 'Form not valid';
        } else {
            // Insert or Update to db
            $mp3Path = saveFile($mp3);
            $imgPath = saveFile($img);

            if (isset($_GET['id'])) {
                $updateSong = "UPDATE songs SET title = '$title',
filePath = '$mp3Path', imgPath = '$imgPath', singerID =
'$singerID' WHERE id = $id";
            }
        }
    }
}

```

```

        $res3 = mysqli_query($conn, $updateSong);
        header("Location: editSong.php");
    } else {
        $insertSong = "INSERT INTO songs(title, filePath,
imgPath, singerID)
VALUES ('$title', '$mp3Path',
'$imgPath', $singerID)";
        if (!mysqli_query($conn, $insertSong)) {
            echo "Error: " . "<br>" . mysqli_error($conn);
        } else {
            header("Location: editSong.php");
        }
    }
}

```

#### Algoritma #4

Contoh:

*Nama Kelas : Kelola Penyanyi*

*Nama Operasi : insertSinger*

*Algoritma :*

```

$name = $infoSinger = $imgFile = "";

if (isset($_GET['id'])) {
    $id = $_GET['id'];
    $sql2 = "SELECT * FROM singers WHERE id= '$id' ";
    $res2 = mysqli_query($conn, $sql2);
    $data = mysqli_fetch_array($res2);
    $name = $data["name"];
    $imgFile = "../" . $data["image"];
    $infoSinger = $data["info"];
}

$errors = array('singername' => '', 'info' => '', 'img' => '');
$singername = $img = $info = '';

// INSERT SINGER INTO DATABASE
function saveFile($fileInfo)
{
    $filename = $fileInfo['name'];

```

```

$tmpPath = $fileInfo['tmp_name'];
$destinationPath = 'images/singers/' . $filename;

if (move_uploaded_file($tmpPath, '../' . $destinationPath)) {
    echo "Successfully uploaded";
} else {
    echo "Upload fail";
}

return $destinationPath;
}

if (isset($_POST['submit'])) {
    if (empty($_FILES["img"]["name"])) {
        if (!isset($_GET['id']))
            $errors['img'] = "Image field cannot be empty";
    } else {
        if (strpos($_FILES["img"]["type"], "image") !== false) {
            $img = $_FILES['img'];
        } else {
            $errors['img'] = "Wrong file format. Expect an image
file. Please check your file again.";
        }
    }
}

if (empty($_POST['singername'])) {
    $errors['singername'] = "Singer's name can not be empty";
} else {
    $singername = $_POST['singername'];
}

if (empty($_POST['info'])) {
    $errors['info'] = "Info can not be empty";
} else {
    $info = $_POST['info'];
}

if (array_filter($errors)) {
    echo 'Form not valid';
} else {

```

```

        if ($img != "") {
            $images = saveFile($img);
        } else {
            $images = $data["image"];
        }

        //IF GET ID -> UPDATE IT
        if (isset($_GET['id'])) {
            $updateSinger = "UPDATE singers SET name =
'$singername', info = '$info', image = '$images' WHERE id = $id";
            $res3 = mysqli_query($conn, $updateSinger);
            header("Location: editSinger.php");
        } else {
            $insertSinger = "INSERT INTO singers(name, info,
image)
VALUES ('$singername', '$info', '$images')";
            if (!mysqli_query($conn, $insertSinger)) {
                echo "Error: " . "<br>" . mysqli_error($conn);
            } else {
                header("Location: editSinger.php");
            }
        }
    }
}

```

### Algoritma #5

Contoh:

*Nama Kelas : Kelola User*

*Nama Operasi : delete*

*Algoritma :*

```

$id = $_GET['id'];

$sql = "DELETE FROM users WHERE id = '$id'";
$result = mysqli_query($conn, $sql);
if ($result) header("Location: updateUser.php");

```

### Algoritma #6

Contoh:

*Nama Kelas : Kelola Favourite*

*Nama Operasi : delFromFav*

*Algoritma :*

```

if (isset($_GET['songID'])) {

```

```

$songID = $_GET['songID'];

$addToFavQuery = "DELETE FROM favourites
                  WHERE favourites.uid=$uid and
songID=$songID;";

if (mysqli_query($conn, $addToFavQuery)) {
    echo json_encode("Delete " . $songID);
} else {
    echo json_encode("Error " . mysqli_error($conn));
}
}

```

### Algoritma #7

Contoh:

*Nama Kelas : Pencarian  
 Nama Operasi : searchContent  
 Algoritma :*

```

if (isset($_GET['search'])) {
    $filterTexts = $_GET['search'];
    // Get songs from database
    $songsFilterQuery = "SELECT songs.id, songs.title title,
singers.name singerName,
                           songs.filePath audio, songs.imgPath img,
singers.id singerID
                           FROM songs
                           LEFT JOIN singers on singers.id =
songs.singerID
                           WHERE title LIKE '%$filterTexts%' OR
singers.name LIKE '%$filterTexts%';

    $result = mysqli_query($conn, $songsFilterQuery);
    $songs = mysqli_fetch_all($result, MYSQLI_ASSOC);
}

```

### Algoritma #8

Contoh:

*Nama Kelas : Kelola Queue  
 Nama Operasi : insertToQueue  
 Algoritma :*

```

const insertToQueue = (song) => {
    if (!playingQueue.includes(song)) {

```

```

        playingQueue.push(song);
        alert(`Songs ${song["title"]} is added to queue!!`);
        resetPlayingQueue();
    } else {
        alert(`Songs ${song["title"]} is already in playing
queue!!`);
    }
};

```

### Algoritma #9

Contoh:

*Nama Kelas : Lagu*  
*Nama Operasi : nextSong*  
*Algoritma :*

```

function nextSong() {
    songIndex++;

    if (songIndex > playingQueue.length - 1) {
        songIndex = 0;
    }

    loadSong(playingQueue[songIndex]);
    playSong();
    resetPlayingQueue();
}

```

### Algoritma #10

Contoh:

*Nama Kelas : Lagu*  
*Nama Operasi : playSong*  
*Algoritma :*

```

function playSong() {
    if (audio.src.includes("#")) return;

    playBtn.querySelector("i.fas").classList.remove("fa-play");
    playBtn.querySelector("i.fas").classList.add("fa-pause");
    audio.play();
    isPlaying = true;
}

```

### Algoritma #11

Contoh:

*Nama Kelas : Lagu*

```

Nama Operasi : loadSong
Algoritma   :
function loadSong(song) {
    audio.src = song["audio"];
    coverImg.src = song["img"];
    title.innerText = song["title"];
    singerName.innerText = song["singerName"];
}

```

### Query #1

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Users

*SELECT \* FROM users*

Keterangan : Query untuk memanggil data Users

### Query #2

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Songs

*SELECT \* FROM songs*

Keterangan : Query untuk memanggil data Songs

### Query #3

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Singers

*SELECT \* FROM singers*

Keterangan : Query untuk memanggil data Singers

### Query #4

{Jika mengacu query tertentu, lengkapi query di bawah}

Query : Favourites

*SELECT \* FROM favourites*

Keterangan : Query untuk memanggil data Favourites

## 4. Matriks Kerunutan (Requirement Traceability Matrix)

### Mapping Requirement dengan Use Case yang direalisasikan

Kode FR	Nama Functional Requirement	Nama Use Case
FR-01	User dapat mencari lagu yang diinginkan melalui suatu kolom pencarian	Pencarian Lagu
FR-02	User dapat memutar lagu secara online	Memutar Lagu

FR-03	User dapat mengatur lagu, seperti pengaturan play, next, back, pause, dan mengatur volume lagu.	Mengatur Lagu
FR-04	User dapat mengelola favourites	Kelola Favourites
FR-05	User dapat mengelola antrian lagu	Kelola Queue
FR-06	Admin dapat mengelola data musisi	Kelola Musisi
FR-07	Admin dapat mengelola data lagu	Kelola Lagu
FR-08	Guest User dapat mencari lagu yang diinginkan melalui suatu kolom pencarian	Pencarian Lagu
FR-09	Guest User dapat memutar lagu secara online	Memutar Lagu
FR-10	Guest User dapat mengatur lagu seperti Play	Mengatur Lagu