

BAB III

ANALISIS DAN PERANCANGAN

III. Analisa Masalah

III.1. Analisis Sistem yang Berjalan

Analisa sistem yang berjalan bertujuan untuk mengidentifikasi serta melakukan evaluasi terhadap Rancang Bangun Aplikasi *Fetcher* Lirik Berbasis Android. Adapun masalah yang terdapat pada penelitian adalah belum berkembang suatu aplikasi *Fetcher* lirik berbasis android, banyaknya peminat musik yang terkadang tidak hafal lirik tersebut, lirik yang dilihat dari web ditampilkan secara keseluruhan sehingga kurang menarik jika dibaca saat musik dimainkan. Analisis dilakukan agar dapat menemukan masalah-masalah dalam melakukan pengembangan sistem yang diperlukan yang diberikan oleh pihak pengguna.

III.1.1.Strategi Pemecahan Masalah

Strategi dalam melakukan pemecahan masalah yang sedang dianalisa oleh penulis mengenai Rancang Bangun Aplikasi *Fetcher* Lirik Berbasis Android adalah sebagai berikut :

1. Merancang suatu aplikasi *Fetcher* lirik menggunakan android.
2. Merancang sebuah aplikasi *Fetcher* dengan tampilan yang sederhana dan menarik peminat musik.

III.2. Desain Sistem

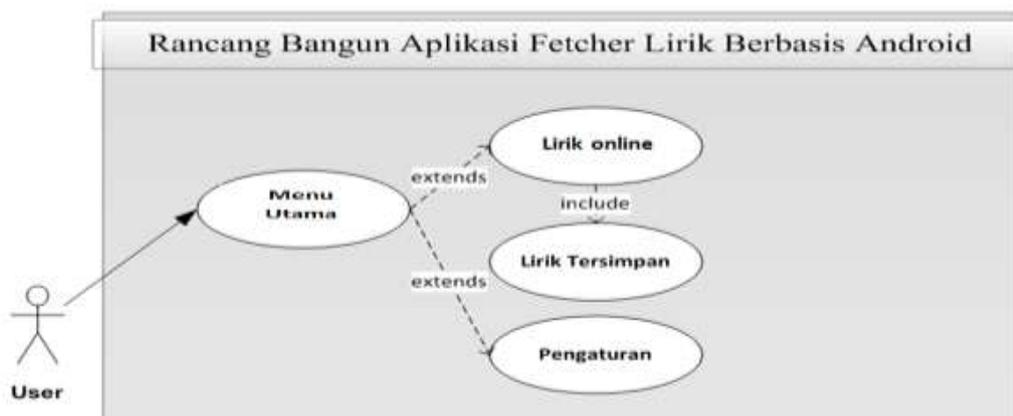
Desain sistem pada penelitian ini dibagi menjadi dua desain, yaitu desain sistem secara global untuk penggambaran model sistem secara garis besar dan desain sistem secara detail untuk membantu dalam pembuatan sistem.

III.2.1. Desain Sistem Secara Global

Desain sistem secara global menggunakan bahasa pemodelan UML yang terdiri dari *Usecase Diagram*, *Acitivity Diagram*, dan *Sequence Diagram*.

III.2.1.1. Usecase Diagram

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk diagram yang dapat menjelaskan suatu alur proses sistem yang akan di bangun. Dalam penulisan skripsi ini penulis menggunakan metode UML yang dalam metode itu penulis menerapkan diagram *Use Case*. Maka digambarlah suatu bentuk diagram *Use Case* yang dapat dilihat pada gambar dibawah ini



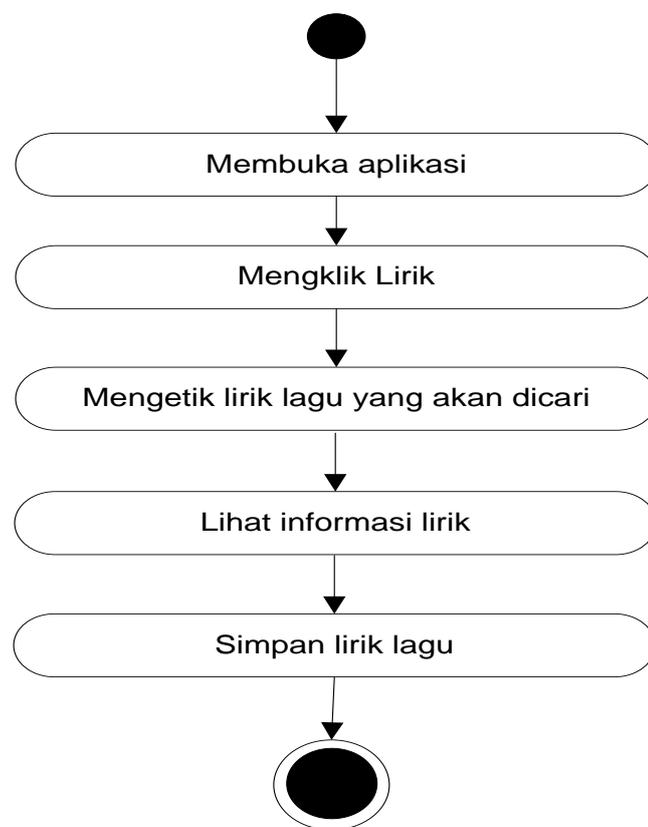
Gambar III.1. *Use Case Diagram* Rancang Bangun Aplikasi *Fetcher Lirik* Berbasis Android

III.2.1.2. Activity Diagram

Bisnis proses yang telah digambarkan pada *use case diagram* dijabarkan dengan *Activity diagram* :

1. Activity Diagram Lirik

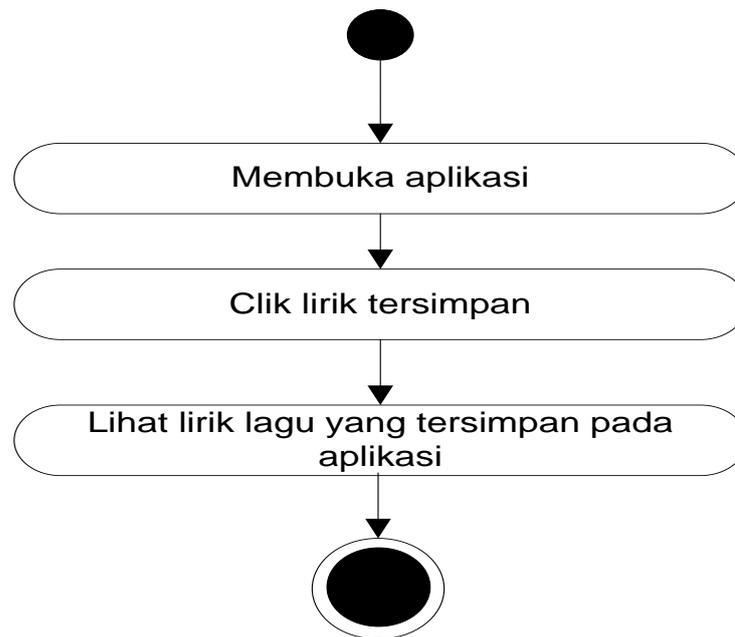
Aktifitas untuk melakukan aktifitas pada form lirik terlihat seperti pada gambar III.3 berikut :



Gambar.III.2. Activity Diagram Lirik

2. Activity Diagram Lirik Tersimpan

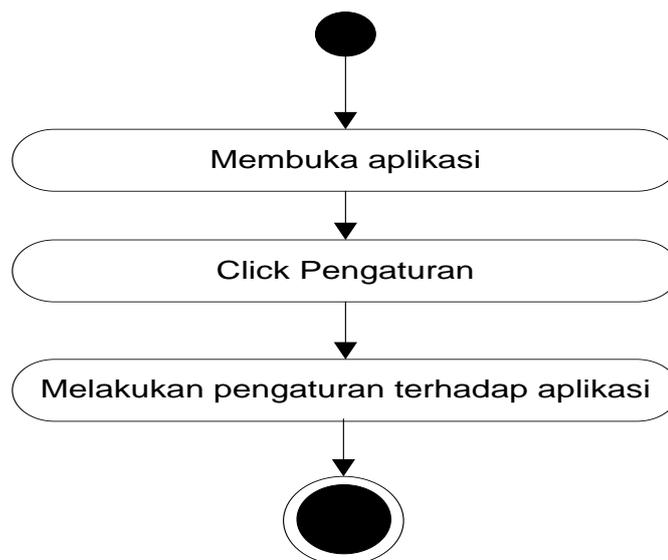
Aktifitas untuk melakukan aktifitas pada form lirik tersimpan terlihat seperti pada gambar III.4 berikut :



Gambar.III.3. Acitvity Diagram Lirik Tersimpan

3. Activity Diagram Pengaturan

Aktifitas untuk melakukan kegiatan pada form pengaturan terlihat seperti pada gambar III.5 berikut :



Gambar.III.4. Acitvity Diagram Pengaturan

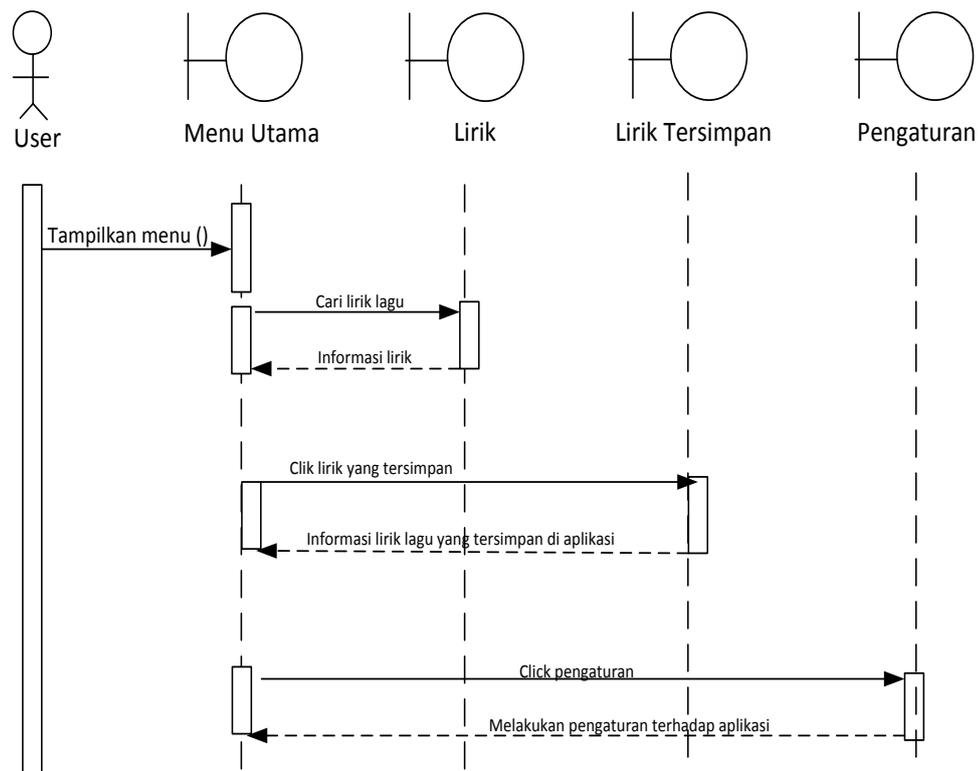
III.2.1.3. *Sequence Diagram*

Sequence Diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message*.

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah – langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu. *Sequence Diagram* diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan *output* apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan use case diagram. *Sequence diagram* juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam use case. *Sequence Diagram* juga dapat merubah atribut atau method pada class yang telah dibentuk oleh class diagram, bahkan menciptakan sebuah class baru. *Sequence Diagram* memodelkan aliran logika dalam sebuah sistem dalam cara yang visual.

1. *Sequence Diagram* Aplikasi *Fetcher* Aktifitas untuk melakukan pengolahan data aplikasi *fetcher* terlihat seperti pada gambar III.7 berikut :



Gambar.III.5. Sequence Diagram Aplikasi *Fetcher* Musik Pada Android

III.3. Desain *User Interface*

Tahap perancangan berikutnya yaitu desain sistem secara detail yang meliputi desain sistem.

1. Desain *Form* Menu Utama Aplikasi

Desain *Form* yang dirancang untuk tampilan menu utama dari aplikasi *fetcher* musik terlihat seperti pada gambar III.8 berikut :

...

Paindo Lirik

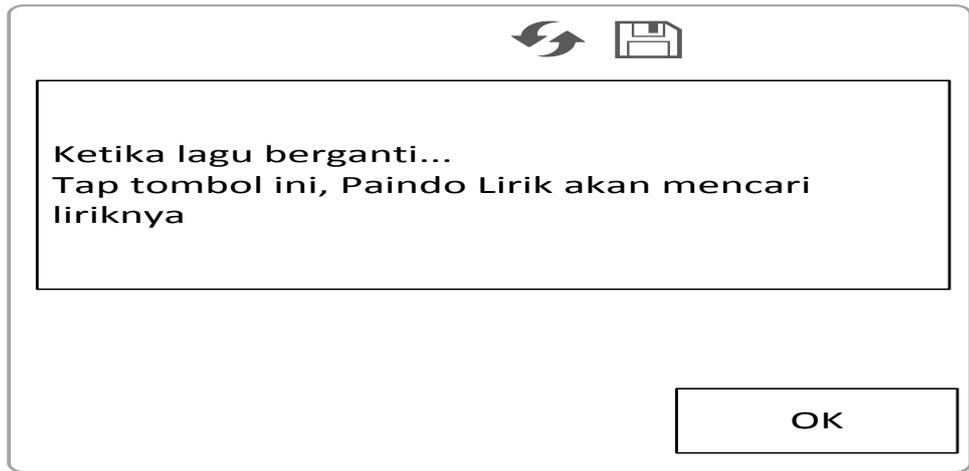
Selamat Datang !
Lirik yang anda cari ada disini ...

Ok

Gambar.III.6. Desain *Form* Menu Utama

2. Desain *Form* Menu Lirik

Desain *Form* menu lirik yang dirancang untuk melihat informasi lirik terlihat seperti pada gambar III.9 berikut :



Gambar.III.7. Desain *Form* Menu Lirik

3. Desain *Form* Lirik Tersimpan

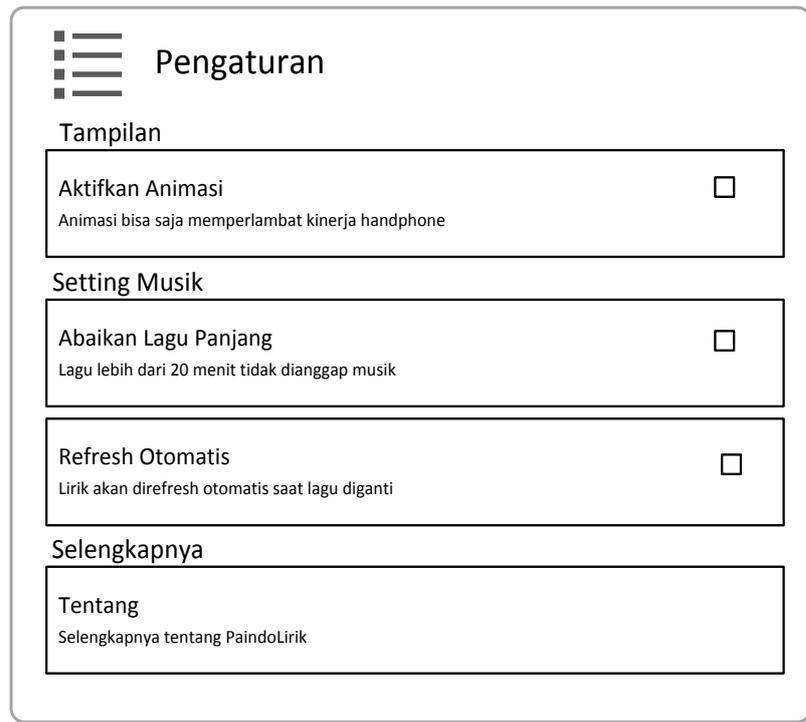
Desain *Form* lirik tersimpan yang dirancang untuk melihat informasi lirik yang telah tersimpan dan telah di pilih oleh *user* terlihat seperti pada gambar III.10 berikut :



Gambar.III.8. Desain *Form* Informasi Lirik Tersimpan

4. Desain *Form* Pengaturan

Desain *Form* pengaturan yang dirancang untuk melakukan pengaturan aplikasi yang telah di pilih oleh *user* terlihat seperti pada gambar III.11 berikut



The image shows a settings menu titled "Pengaturan" with a hamburger menu icon on the left. It is organized into three sections:

- Tampilan**: Contains the option "Aktifkan Animasi" with a checkbox and a subtext "Animasi bisa saja memperlambat kinerja handphone".
- Setting Musik**: Contains the option "Abaikan Lagu Panjang" with a checkbox and a subtext "Lagu lebih dari 20 menit tidak dianggap musik".
- Refresh Otomatis**: Contains the option "Refresh Otomatis" with a checkbox and a subtext "Lirik akan direfresh otomatis saat lagu diganti".

At the bottom, there is a section titled "Selengkapnya" with a subtext "Tentang Selengkapnya tentang PaindoLirik".

Gambar.III.9. Desain *Form* Pengaturan