

## BAB 4

### IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi yang dibangun. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan ke dalam bahasa pemrograman. Setelah implementasi maka dilakukan pengujian terhadap aplikasi. Aplikasi yang telah dibangun akan dilakukan pengujian untuk mengetahui apakah sistem tersebut dapat berjalan sesuai dengan tujuan atau tidak yang selanjutnya diadakan untuk pengembangan sistem.

#### 4.1. Implementasi Sistem

Implementasi sistem merupakan tahap untuk menerapkan perancangan yang telah dilakukan terhadap sistem sehingga siap untuk dioperasikan. Implementasi sistem meliputi implementasi perangkat keras, implementasi perangkat lunak, implementasi data, implementasi antar muka.

##### 4.1.1. Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk membangun sistem aplikasi ini adalah pada Tabel 4. 1 Tabel Implementasi Minimum Perangkat Keras :

**Tabel 4. 1 Tabel Implementasi Minimum Perangkat Keras**

Nama Perangkat Keras	Spesifikasi
RAM	1 GB
Memory Internal	4 GB
Versi GPS	29.19.15.220149
CPU	1.8 GHz

##### 4.1.2. Implementasi Perangkat Lunak

Perangkat lunak (*software*) untuk proses implementasi sistem meliputi sistem operasi dan perambah. Tabel 4. 2 Tabel Implementasi Kebutuhan Perangkat Lunak adalah untuk menjalankan aplikasi.

**Tabel 4. 2 Tabel Implementasi Kebutuhan Perangkat Lunak**

Nama Perangkat Lunak	Spesifikasi
Sistem Operasi	Android 6.0 Masmellow
Perambah	Google Chrome

**4.1.3. Implementasi Basis Data**

Implementasi basis data merupakan implementasi berdasarkan perancangan basis data yang dibuat sebelumnya. Secara fisik implementasi database ini menggunakan MySQL. Berikut adalah sintaks pembangun dari database yang digunakan :

## 1. Tabel pengguna

Implementasi pembuatan tabel pengguna dapat dilihat pada Tabel 4. 3 *Query SQL* Tabel pengguna di bawah ini :

**Tabel 4. 3 Query SQL Tabel pengguna**

Nama Tabel	Query SQL
pengguna	<pre>CREATE TABLE `pengguna` (   `email` varchar(50) NOT NULL,   `nama` varchar(50) NOT NULL,   `password` varchar(50) NOT NULL,   `telepon` varchar(20) NOT NULL,   PRIMARY KEY (`email`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1</pre>

2. Tabel *booking*

Implementasi pembuatan tabel booking dapat dilihat pada Tabel 4. 4 *Query SQL* Tabel *booking* di bawah ini :

**Tabel 4. 4 Query SQL Tabel *booking***

Nama Tabel	Query SQL
booking	<pre>CREATE TABLE `booking` (   `kode` varchar(20) NOT NULL,   `email_pemesan` varchar(50) NOT NULL,   `email_musisi` varchar(50) NOT NULL,   `tanggal` date NOT NULL,   `acara` varchar(20) NOT NULL,</pre>

Nama Tabel	<i>Query SQL</i>
	<pre> `lokasi` text NOT NULL, `status` varchar(20) NOT NULL DEFAULT 'Belum di verifikasi', PRIMARY KEY (`kode`), KEY `email_pemesan` (`email_pemesan`), KEY `email_musisi` (`email_musisi`), ) ENGINE=InnoDB DEFAULT CHARSET=latin1 </pre>

### 3. Tabel musisi

Implementasi pembuatan tabel musisi dapat dilihat pada Tabel 4. 5 *Query SQL* Tabel musisi di bawah ini.

**Tabel 4. 5 *Query SQL* Tabel musisi**

Nama Tabel	<i>Query SQL</i>
musisi	<pre> CREATE TABLE `musisi` ( `email` varchar(50) NOT NULL, `nama` varchar(50) NOT NULL, `password` varchar(50) NOT NULL, `foto` varchar(200) NOT NULL, `alamat` varchar(200) NOT NULL, `tarif` int(11) NOT NULL, `personel` int(11) NOT NULL, `deskripsi` text NOT NULL, `channel_id` varchar(100) NOT NULL, `id_genre` int(11) NOT NULL, `latitude` double NOT NULL, `longitude` double NOT NULL, PRIMARY KEY (`email`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 </pre>

### 4. Tabel detail\_musisi

Implementasi pembuatan tabel detail\_musisi dapat dilihat pada Tabel 4. 6 *Query SQL* Tabel detail\_musisi di bawah ini.

**Tabel 4. 6 *Query SQL* Tabel detail\_musisi**

Nama Tabel	<i>Query SQL</i>
detail_musisi	<pre> CREATE TABLE `detail_musisi` ( `id` int(11) NOT NULL AUTO_INCREMENT, </pre>

Nama Tabel	<i>Query SQL</i>
	<pre> `email_musisi` varchar(50) NOT NULL, `id_genre` int(11) NOT NULL, `id_kategori` int(11) NOT NULL, PRIMARY KEY (`id`), KEY `email_musisi` (`email_musisi`), KEY `id_genre` (`id_genre`), KEY `id_kategori` (`id_kategori`) ) ENGINE=InnoDB DEFAULT CHARSET=latin1 </pre>

#### 5. Tabel genre

Implementasi pembuatan tabel genre dapat dilihat pada Tabel 4. 7 *Query SQL* Tabel genre di bawah ini.

**Tabel 4. 7 *Query SQL* Tabel genre**

Nama Tabel	<i>Query SQL</i>
genre	<pre> CREATE TABLE `genre` ( `id` int(11) NOT NULL AUTO_INCREMENT, `nama` varchar(200) NOT NULL, `gambar` varchar(200) NOT NULL, PRIMARY KEY (`id`), KEY `nama` (`nama`), KEY `gambar` (`gambar`) )ENGINE=InnoDB AUTO_INCREMENT=5 DEFAULT CHARSET=latin1 </pre>

#### 6. Tabel kategori

Implementasi pembuatan tabel kategori dapat dilihat pada Tabel 4. 8 *Query SQL* Tabel kategori di bawah ini.

**Tabel 4. 8 *Query SQL* Tabel kategori**

Nama Tabel	<i>Query SQL</i>
kategori	<pre> CREATE TABLE `kategori` ( `id` int(11) NOT NULL AUTO_INCREMENT, `nama` varchar(50) NOT NULL, </pre>

<b>Nama Tabel</b>	<b><i>Query SQL</i></b>
	<pre>`gambar` varchar(200) NOT NULL,  PRIMARY KEY (`id`)  )ENGINE=InnoDB AUTO_INCREMENT=8 DEFAULT CHARSET=latin1</pre>

## 7. Constraint

Implementasi pembuatan relasi antar tabel dapat dilihat pada Tabel 4.9 *Query SQL Constraint* di bawah ini :

**Tabel 4.9 Query SQL Constraint**

```
TABLE `booking`
ADD CONSTRAINT `booking_ibfk_1` FOREIGN KEY (`email_pemesan`)
REFERENCES `pengguna` (`email`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `booking_ibfk_2` FOREIGN KEY (`email_musisi`)
REFERENCES `musisi` (`email`) ON DELETE CASCADE ON UPDATE CASCADE

ALTER TABLE `detail_musisi`
ADD CONSTRAINT `detail_musisi_ibfk_1` FOREIGN KEY (`email_musisi`)
REFERENCES `musisi` (`email`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `detail_musisi_ibfk_2` FOREIGN KEY (`id_genre`)
REFERENCES `genre` (`id`) ON DELETE CASCADE ON UPDATE CASCADE,
ADD CONSTRAINT `detail_musisi_ibfk_3` FOREIGN KEY (`id_kategori`)
REFERENCES `kategori` (`id`) ON DELETE CASCADE ON UPDATE CASCADE
```

### 4.1.4. Implementasi Class

Implementasi *class* berisi pemaparan setiap *class* dari fungsionalitas yang dibangun. Adapun implementasi *class* perangkat lunak yang dibangun terdiri dari nama *class* atau *file* yang mewakilinya dapat di lihat pada Tabel 4.10 *Tabel Implementasi Class*.

**Tabel 4. 10 Tabel Implementasi Class**

<b>Menu</b>	<b>Deskripsi</b>	<b>Nama File</b>
<i>Login</i>	Merupakan class untuk <i>login</i>	MasukActivity.java
Daftar Akun	Merupakan class untuk membuat akun baru	DaftarActivity.java
Lupa Password	Merupakan class untuk melakukan lupa <i>password</i>	LupaActivity.java
Mencari Musisi	Merupakan class untuk mencari musisi	MainActivity.java
Melakukan <i>Booking</i>	Merupakan class untuk melakukan <i>booking</i>	BookingActivity.java
Konfirmasi <i>Booking</i>	Merupakan class untuk melakukan konfirmasi <i>booking</i>	MusisiActivity.java

#### 4.1.5. Implementasi Antar Muka

Implementasi antarmuka berisi pemaparan setiap tampilan perangkat lunak yang dibangun. Adapun implementasi antarmuka perangkat lunak yang dibangun dapat dilihat pada Tabel 4. 11 Tabel Implementasi Antar Muka di bawah ini:

**Tabel 4. 11 Tabel Implementasi Antar Muka**

<b>Menu</b>	<b>Deskripsi</b>	<b>Nama File</b>
<i>Login</i>	Merupakan halaman untuk <i>login</i>	activity_login.xml
Daftar Akun	Merupakan halaman untuk membuat akun baru	activity_daftar.xml
Lupa <i>Password</i>	Merupakan halaman untuk melakukan lupa <i>password</i>	activity_lupa.xml
Mencari Musisi	Merupakan halaman untuk mencari musisi	activity_main.java

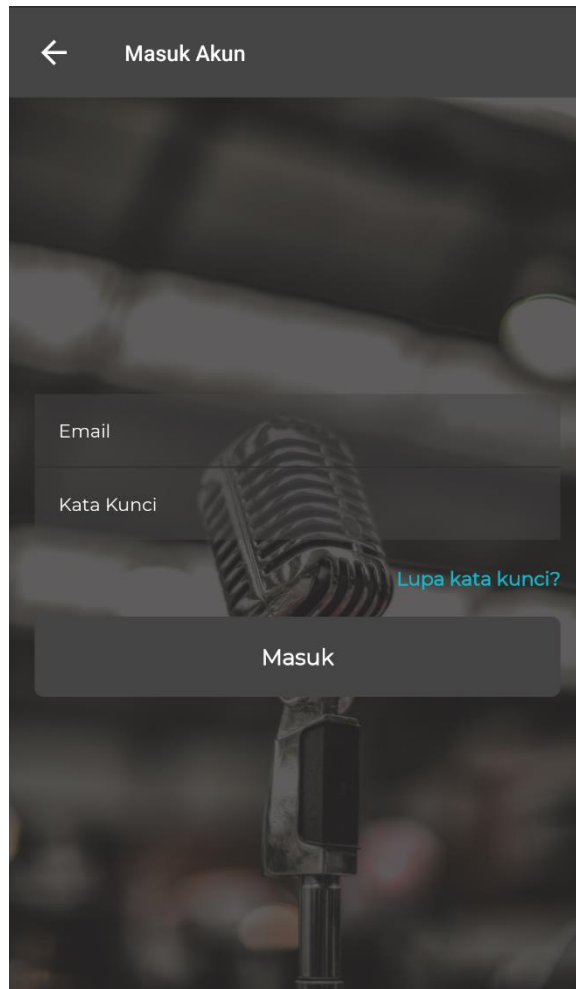
<b>Menu</b>	<b>Deskripsi</b>	<b>Nama File</b>
Melakukan <i>Booking</i>	Merupakan halaman untuk melakukan <i>booking</i>	activity_booking.java
Konfirmasi <i>Booking</i>	Merupakan halaman untuk melakukan konfirmasi <i>booking</i>	activity_musisi.java

Dari tabel implementasi di atas, adapun beberapa tampilan antarmuka dari aplikasi adalah sebagai berikut.



### 1. Implementasi Antarmuka *Login*

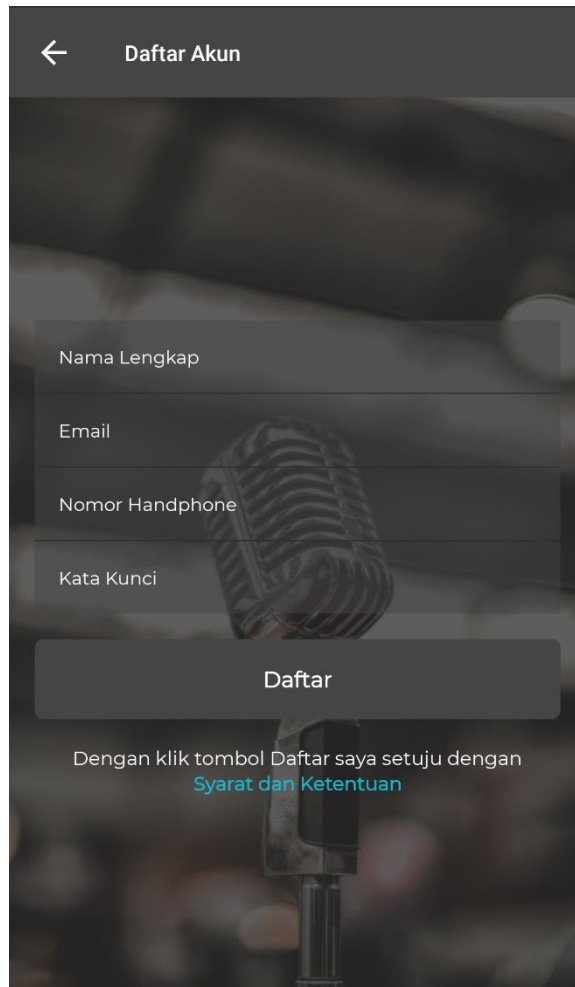
Implementasi antarmuka *login* merupakan antarmuka untuk *autentikasi user* masuk ke dalam system dapat dilihat pada Gambar 4. 1 Implementasi Antarmuka *Login* di bawah ini.



**Gambar 4. 1 Implementasi Antarmuka *Login***

## 2. Implementasi Antarmuka Pendaftaran Akun Pencari Musisi

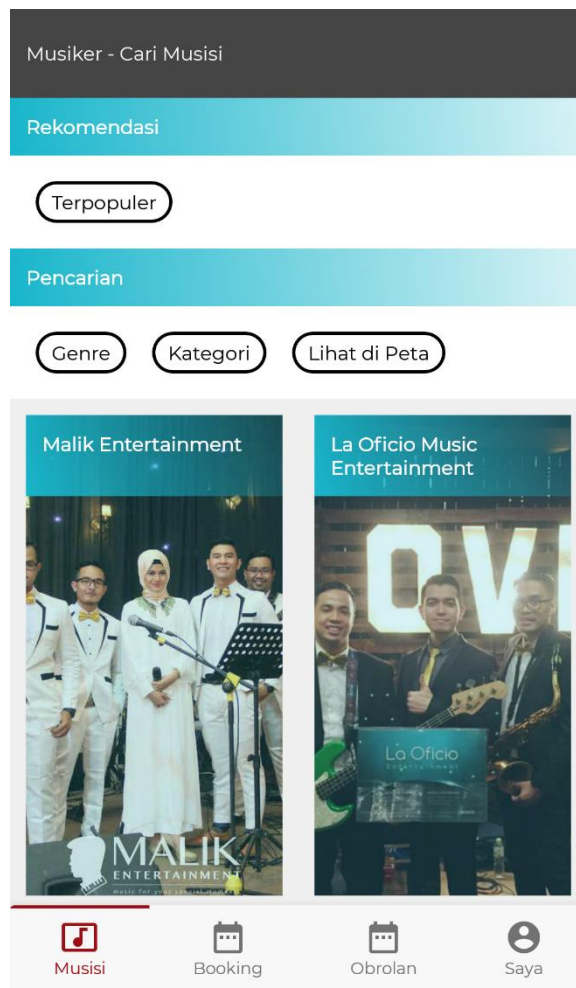
Implementasi antarmuka pendaftaran akun pencari musisi merupakan antarmuka untuk mendaftar agar dapat *login* ke sistem dapat dilihat pada Gambar 4. 2 Implementasi Antarmuka Pendaftaran di bawah ini.



**Gambar 4. 2 Implementasi Antarmuka Pendaftaran**

### 3. Implementasi Antarmuka Halaman Utama

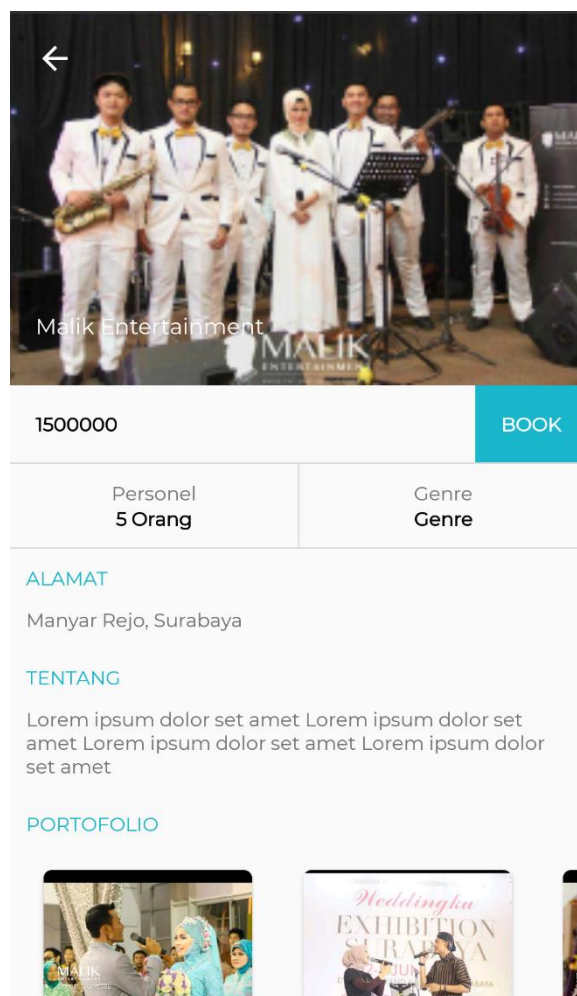
Implementasi antarmuka halaman utama aplikasi merupakan antarmuka untuk memulai melakukan pencarian musisi dapat dilihat pada Gambar 4. 3 Implementasi Antarmuka Halaman Utama di bawah ini .



**Gambar 4. 3 Implementasi Antarmuka Halaman Utama**

#### 4. Implementasi Antarmuka Halaman Portofolio

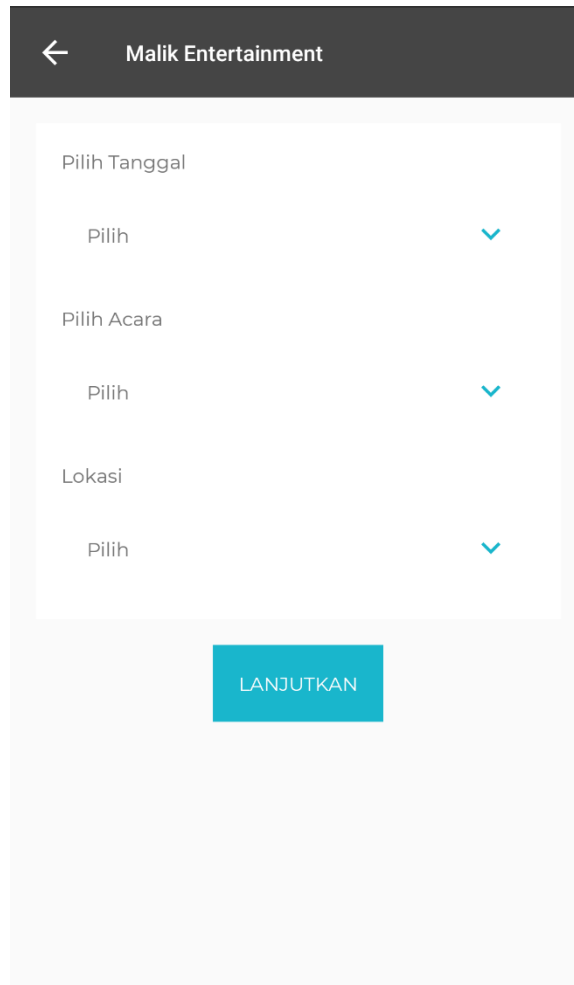
Implementasi antarmuka halaman portofolio merupakan antarmuka untuk melihat portofolio dari musisi di halaman ini pencari musisi dapat melihat detail dari musisi dapat dilihat pada Gambar 4. 4 Implementasi Antarmuka Halaman Portofolio Musisi di bawah ini.



**Gambar 4. 4 Implementasi Antarmuka Halaman Portofolio Musisi**

## 5. Implementasi Antarmuka Melakukan *Booking* Musisi

Implementasi antarmuka *booking* musisi merupakan antarmuka untuk melakukan *booking* dengan mengisi data – data booking dapat dilihat pada Gambar 4. 5 Implementasi Antarmuka Melakukan *Booking* Musisi di bawah ini.

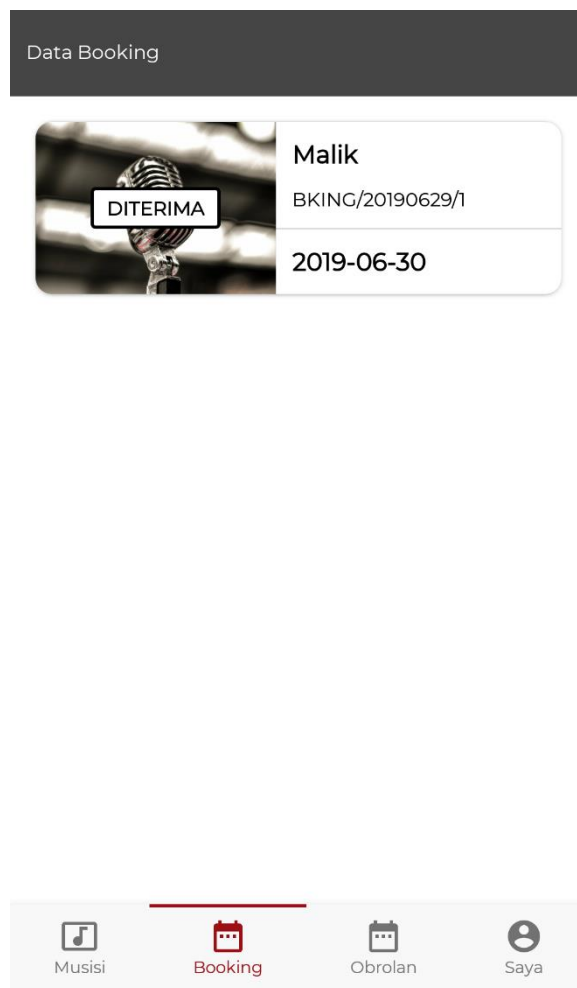


The image shows a mobile application interface for booking a musician. At the top, there is a dark header with a back arrow and the text "Malik Entertainment". Below the header, there is a white card containing three dropdown menus. The first dropdown is labeled "Pilih Tanggal" and has a "Pilih" label with a downward arrow. The second dropdown is labeled "Pilih Acara" and has a "Pilih" label with a downward arrow. The third dropdown is labeled "Lokasi" and has a "Pilih" label with a downward arrow. Below the card, there is a teal button with the text "LANJUTKAN".

**Gambar 4. 5 Implementasi Antarmuka Melakukan *Booking* Musisi**

## 6. Implementasi Antarmuka Halaman Data *Booking*

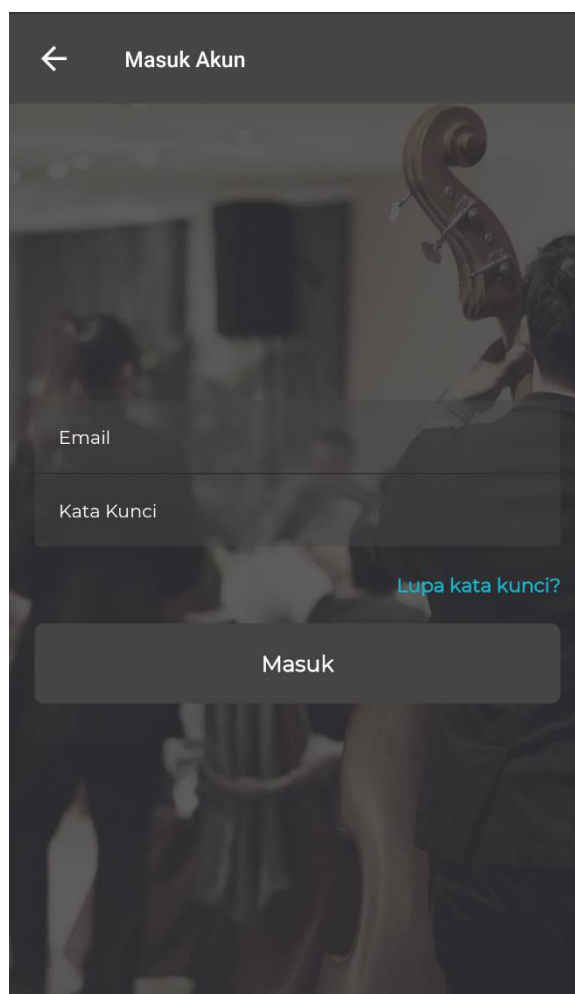
Implementasi antarmuka halaman data *booking* merupakan antarmuka untuk melihat data *booking* dapat dilihat pada Gambar 4. 6 Implementasi Antarmuka Halaman Data *Booking* di bawah ini.



**Gambar 4. 6 Implementasi Antarmuka Halaman Data *Booking***

## 7. Implementasi Antarmuka *Login* Musisi

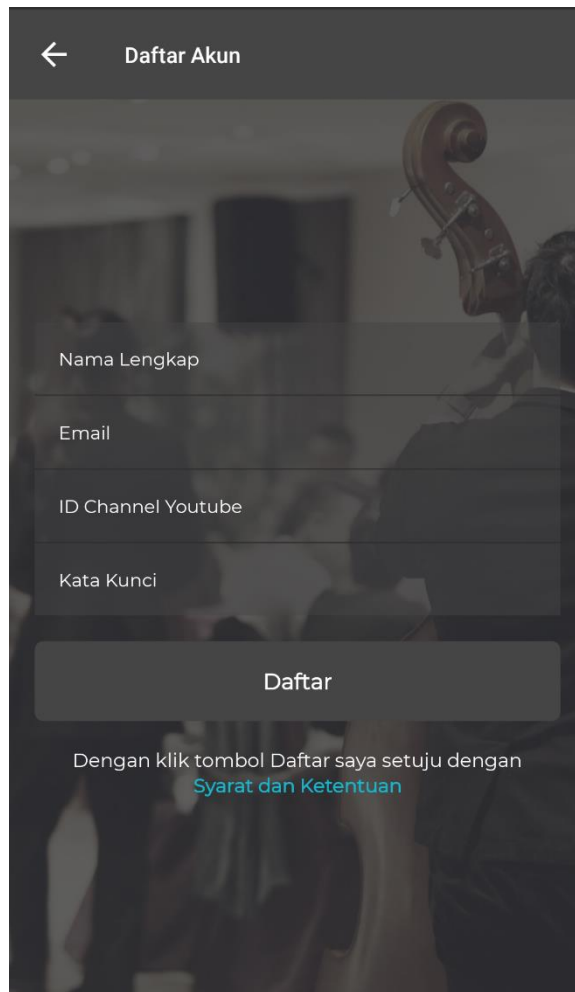
Implementasi antarmuka *login* musisi merupakan antarmuka untuk *autentikasi* musisi masuk ke dalam system dapat dilihat pada Gambar 4. 7 Implementasi Antarmuka *Login* Musisi di bawah ini.



**Gambar 4. 7 Implementasi Antarmuka *Login* Musisi**

## 8. Implementasi Antarmuka Pendaftaran Musisi

Implementasi antarmuka pendaftaran musisi merupakan antarmuka untuk mendaftar agar dapat *login* ke sistem dapat dilihat pada Gambar 4. 8 Implementasi Antarmuka Pendaftaran Musisi di bawah ini.



The image shows a mobile application interface for musician registration. At the top, there is a back arrow and the title "Daftar Akun". Below the title, there is a background image of a person playing a double bass. The form consists of several input fields: "Nama Lengkap", "Email", "ID Channel Youtube", and "Kata Kunci". Below these fields is a "Daftar" button. At the bottom, there is a text prompt: "Dengan klik tombol Daftar saya setuju dengan [Syarat dan Ketentuan](#)".

**Gambar 4. 8 Implementasi Antarmuka Pendaftaran Musisi**



## 9. Implementasi Antarmuka Halaman Data *Booking*

Implementasi antarmuka halaman data *booking* merupakan antarmuka untuk melihat data *booking* yang dan halaman untuk musisi mengkonfirmasi *booking* dari pemesan dapat di lihat pada Gambar 4. 9 Implementasi Antarmuka Data *Booking* di bawah ini.



**Gambar 4. 9 Implementasi Antarmuka Data *Booking***

## 10. Implementasi Antarmuka Halaman Portofolio

Implementasi antarmuka halaman portofolio merupakan antarmuka untuk melihat portofolio musisi dapat di lihat pada Gambar 4.10 Implementasi Antarmuka Portofolio di bawah ini.



**Gambar 4. 10 Implementasi Antarmuka Portofolio**

## 4.2. Pengujian Sistem

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan atau kekurangan pada sistem informasi yang diuji. Pengujian sistem dimaksud untuk mengetahui kinerja sistem informasi yang telah dibuat sesuai dengan tujuan perancangan sistem informasi.

Pengujian dilakukan dengan teknik pengujian *black box* yang lebih berfokus pada menemukan kesalahan program secara fungsional. Sedangkan pada *User Acceptance Test* dilakukan secara langsung kepada responden dengan cara melakukan wawancara dan observasi pada pengguna aplikasi yang dibangun.

### 4.2.1. Pengujian Blackbox

Pengujian *black box* menggunakan metode *Equivalence Partitioning* yaitu teknik pengujian yang difokuskan pada pembagian *domain input* dari program ke dalam kelas-kelas data sehingga *test case* dapat diperoleh, pengujian juga dilakukan secara *Requirement Testing* yaitu menguji apakah fungsi sudah berjalan sesuai kebutuhan.

#### 4.2.1.1. Skenario Pengujian Black Box

Skenario pengujian akan dilakukan pada perangkat lunak ini dapat dilihat pada Tabel 4. 12 Tabel Skenario Pengujian Black Box.

**Tabel 4. 12 Tabel Skenario Pengujian Black Box**

Kelas Uji	Poin Pengujian	Jenis Pengujian
Login	Input data login	Black Box
	Validasi data login	Black Box
Daftar Akun	Input data pendaftaran	Black Box
	Validasi data pendaftaran	Black Box
	Menyimpan data pendaftaran ke database	Black Box
Mencari Musisi	Input data pencarian	Black Box
	Validasi data pencarian	Black Box

Kelas Uji	Poin Pengujian	Jenis Pengujian
	Menampilkan data	<i>Black Box</i>
<i>Booking</i>	<i>Input data booking</i>	<i>Black Box</i>
	Validasi data <i>booking</i>	<i>Black Box</i>
	Menyimpan data <i>booking</i> ke database	<i>Black Box</i>
Konfirmasi <i>Booking</i>	<i>Input data konfirmasi</i>	<i>Black Box</i>
	Validasi data konfirmasi	<i>Black Box</i>
	Menyimpan data konfirmasi ke database	<i>Black Box</i>

#### 4.2.1.2. Hasil Pengujian

Hasil pengujian menampilkan hasil dari pengujian yang dilakukan sesuai dengan rencana dan skenario pengujian. Hasil pengujian ini terbagi menjadi dua yaitu hasil pengujian fungsionalitas dan hasil pengujian beta.

##### 1. Hasil Pengujian Fungsionalitas

1) Hasil Pengujian *Login* dapat dilihat pada Tabel 4. 13 Hasil Pengujian

*Login* di bawah ini.

**Tabel 4. 13 Hasil Pengujian *Login***

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Email: lionelrefli75@gmail .com  <i>Password: testes</i>	Sistem Menampilkan menu utama	Tampil Menu Utama	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Email:{kosong}  Password:{kosong}	Sistem menampilkan pesan “Email atau <i>password</i> tidak boleh kosong”.	Tampil pesan “Email atau <i>password</i> tidak boleh kosong”	[ <input checked="" type="checkbox"/> ] Diterima  [ <input type="checkbox"/> ] Ditolak

- 2) Hasil Pengujian Daftar Akun dapat di lihat pada Tabel 4. 14 Hasil Pengujian Daftar Akun di bawah ini.

**Tabel 4. 14 Hasil Pengujian Daftar Akun**

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama Pengguna:  lionel refl  Email: lionelrefli75@gmail.com  Password: testes	Sistem menampilkan pesan “Data akun berhasil tersimpan”	Tampil pesan “Data akun berhasil tersimpan”	[ <input checked="" type="checkbox"/> ] Diterima  [ <input type="checkbox"/> ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Hasil yang diharapkan	Pengamatan	Kesimpulan
Nama Pengguna: {kosong}  Email: {kosong}  Password: {kosong}	Sistem menampilkan pesan “Lengkapi data akun”	Tampil pesan “Lengkapi data akun”	[ <input checked="" type="checkbox"/> ] Diterima  [ <input type="checkbox"/> ] Ditolak

- 3) Hasil Pengujian Pencarian Musisi dapat di lihat pada Tabel 4. 15 Hasil Pengujian Pencarian Musisi di bawah ini.

**Tabel 4. 15 Hasil Pengujian Pencarian Musisi**

<b>Kasus dan Hasil Uji (Data Benar)</b>			
<b>Data Masukan</b>	<b>Hasil yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Data Pencarian: {terisi}	Sistem menampilkan data musisi	Tampil data musisi	[ <input checked="" type="checkbox"/> ] Diterima  [ <input type="checkbox"/> ] Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
<b>Data Masukan</b>	<b>Hasil yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Data Pencarian: {kosong}	Sistem menampilkan pesan “Data tidak ditemukan”	Tampil pesan “Data tidak ditemukan”	[ <input checked="" type="checkbox"/> ] Diterima  [ <input type="checkbox"/> ] Ditolak

- 4) Hasil Pengujian Melakukan *Booking* dapat di lihat pada Tabel 4. 16 Hasil Pengujian Melakukan Booking di bawah ini.

**Tabel 4. 16 Hasil Pengujian Melakukan *Booking***

<b>Kasus dan Hasil Uji (Data Benar)</b>			
<b>Data Masukan</b>	<b>Hasil yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Data <i>Booking</i> : {terisi}	Sistem menampilkan pesan “Proses <i>booking</i> berhasil”	Tampil pesan “Proses <i>booking</i> berhasil”	[ <input checked="" type="checkbox"/> ] Diterima  [ <input type="checkbox"/> ] Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
<b>Data Masukan</b>	<b>Hasil yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>

Data <i>Booking</i> : {kosong}	Sistem menampilkan pesan “Lengkapi data”	Tampil pesan “Lengkapi data”	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak
-----------------------------------	--	------------------------------	--

- 5) Hasil Pengujian Melakukan Konfirmasi *Booking* dapat di lihat pada Tabel 4. 17 Hasil Pengujian Konfirmasi *Booking* di bawah ini.

**Tabel 4. 17 Hasil Pengujian Konfirmasi *Booking***

<b>Kasus dan Hasil Uji (Data Benar)</b>			
<b>Data Masukan</b>	<b>Hasil yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Data <i>Booking</i> : {terisi}	Sistem menampilkan pesan “Data <i>booking</i> berhasil dikonfirmasi”	Tampil pesan “Data <i>booking</i> berhasil dikonfirmasi”	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
<b>Data Masukan</b>	<b>Hasil yang diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Data <i>Booking</i> : {kosong}	Sistem menampilkan pesan “Gagal konfirmasi <i>booking</i> ”	Tampil pesan “Gagal konfirmasi <i>booking</i> ”	[ <input checked="" type="checkbox"/> ] Diterima [ <input type="checkbox"/> ] Ditolak

#### **4.2.2. User Acceptance Test**

*User Acceptance Test* merupakan pengujian yang dilakukan secara objektif di mana pengujian dilakukan secara langsung oleh pengguna, dengan cara melakukan wawancara mengenai tanggapan pengguna terhadap sistem yang dibangun.

##### **4.2.2.1. Skenario Pengujian**

Pengujian *beta* ini dilakukan dengan cara melakukan pengujian langsung di tempat penelitian dengan menggunakan teknik *User Acceptance Test* (LAMPIRAN B) yaitu teknik pengujian penerimaan aplikasi oleh pengguna dengan cara melakukan *test* terhadap sejumlah fungsi sistem

##### **4.2.2.2. Data Kuisiner Pengujian**

Kuisiner yang berjudul aplikasi pencarian dan pemesanan musisi untuk *event* berdasarkan portofolio memanfaatkan API Youtube dan sensor GPS pada

*smartphone* android (Musiker) ini diajukan kepada responden mempunyai 5 pertanyaan dengan menggunakan skala 1 sampai 5. Berikut ini merupakan tabel kriteria interpretasi skor berdasarkan interval yang dapat dilihat pada Tabel 4. 18 Tabel Skala Jawaban Kuesioner.

**Tabel 4. 18 Tabel Skala Jawaban Kuesioner Kuesioner**

Jawaban	Skor
SS = Setuju	5
S= Tidak Setuju	4
RR= Ragu-ragu	3
TS=Tidak Setuju	2
STS=Sangat Tidak Setuju	1

- Untuk menghitung jumlah maksimum skor kriterium jumlah pertanyaan = 5, dan jumlah responden = 30 yaitu dengan rumus: Kriterium = Jumlah Soal X Jumlah Responden maka nilai dari kuisisioner yang dibuat adalah  $5 \times 30 = 150$
- Sedangkan untuk mengetahui jumlah jawaban yang diajukan kepada 30 orang dari responden dalam bentuk persentase, digunakan rumus sebagai berikut:

$$Y = P/Q * 100\%$$

Keterangan :

Y = Nilai persentase yang dicari

P = Jumlah frekuensi dikalikan nilai yang ditetapkan jawaban

Q = Nilai tertinggi dilakukan dengan jumlah sampel

Berikut ini adalah pertanyaan kuisisioner yang diajukan kepada responden yang nantinya akan menggunakan aplikasi pencarian dan pemesanan musisi untuk *event* berdasarkan portofolio memanfaatkan API Youtube dan sensor GPS pada *smartphone* android (Musiker) dan bisa dilihat pada Tabel 4. 19 Tabel Pertanyaan :



**Tabel 4. 19 Tabel Pertanyaan**

No	Pertanyaan
1	Apakah aplikasi ini memudahkan anda untuk menemukan musisi ?
2	Apakah aplikasi ini memudahkan anda dalam memesan musisi yang sesuai dengan yang Anda inginkan?
3	Apakah aplikasi ini mudah di operasikan?
4	Apakah tampilan aplikasi ini nyaman untuk dilihat?
5	Apakah fitur <i>chatting</i> di aplikasi ini memudahkan anda untuk berkomunikasi dengan musisi?

#### 4.2.2.3.Perhitungan Hasil Kuisisioner

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya. Kuisisioner ini telah diajukan kepada 0 orang calon pengguna pembangunan pencarian dan pemesanan musisi untuk *event* berdasarkan portofolio memanfaatkan API Youtube dan sensor GPS pada *smartphone* android (Musiker). Maka hasil perhitungan persentase masing-masing jawaban dapat dilihat sebagai berikut:

##### 1. Apakah aplikasi ini memudahkan anda untuk menemukan musisi?

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Presentasi	Keputusan
Sangat Setuju	5	10	50	$(110/(30*5))*100=73.3$ %	Setuju
Setuju	4	20	80		
Ragu-Ragu	3	0	0		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	110		

Berdasarkan perhitungan di atas, total skor yang didapat adalah 110 skor dengan skala kategori jawaban di antaranya setuju dan sangat setuju. Sedangkan

hasil dari nilai presentasi responden 73.3% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini sudah dapat memenuhi tujuan untuk kemudahan para pengguna untuk menemukan musisi.

2. Apakah aplikasi ini memudahkan anda dalam memesan musisi yang sesuai dengan kriteria?

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Presentasi	Keputusan
Sangat Setuju	5	14	70	$(132/(30*5))*100$ =88%	Setuju
Setuju	4	14	56		
Ragu-Ragu	3	2	6		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	132		

Berdasarkan perhitungan di atas, total skor yang didapat adalah 132 skor dengan skala kategori jawaban di antaranya ragu-ragu, setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden 88% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini sudah dapat memenuhi tujuan untuk kemudahan para pengguna untuk melakukan pemesanan musisi yang sesuai dengan kriteria yang diinginkan.

3. Apakah aplikasi ini mudah di operasikan?

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Presentasi	Keputusan
Sangat Setuju	5	10	50	$(127/(30*5))*100$ =84.7%	Setuju
Setuju	4	17	68		
Ragu-Ragu	3	3	9		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	127		

Berdasarkan perhitungan di atas, total skor yang didapat adalah 127 skor dengan skala kategori jawaban di antaranya ragu-ragu, setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden 84.7% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat dengan mudah dioperasikan.

4. Apakah tampilan aplikasi ini nyaman untuk dilihat?

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Presentasi	Keputusan
Sangat Setuju	5	10	50	$(110/(30*5))*100$ =73.3%	Setuju
Setuju	4	20	80		
Ragu-Ragu	3	0	0		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	110		

Berdasarkan perhitungan di atas, total skor yang didapat adalah 110 skor dengan skala kategori jawaban di antaranya setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden 73.3% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini nyaman untuk dilihat.

5. Apakah fitur *chatting* di aplikasi ini memudahkan anda untuk berkomunikasi dengan musisi?

Kategori Jawaban	Skor	Frekuensi Jawaban	Total Skor	Nilai Presentasi	Keputusan
Sangat Setuju	5	3	15	$(113/(30*5))*100$ =75.3%	Tidak Setuju
Setuju	4	17	68		
Ragu-Ragu	3	10	30		
Tidak Setuju	2	0	0		
Sangat Tidak Setuju	1	0	0		
Jumlah		30	113		

Berdasarkan perhitungan di atas, total skor yang didapat adalah 113 skor dengan skala kategori jawaban di antaranya setuju dan sangat setuju. Sedangkan hasil dari nilai presentasi responden 75.3% dari nilai yang diharapkan sebesar 100%. Maka dapat disimpulkan bahwa aplikasi ini dapat melakukan *chatting* pada aplikasi.

#### **4.2.2.4. Kesimpulan**

Berdasarkan hasil persentasi perhitungan untuk pengujian beta kepada masyarakat sebagai pengguna pencarian dan pemesanan musisi untuk *event* berdasarkan portofolio memanfaatkan *API Youtube* dan sensor *GPS* pada *smartphone android* (Musiker), maka dengan ini disimpulkan bahwa pembangunan pencarian dan pemesanan musisi untuk *event* berdasarkan portofolio memanfaatkan *API Youtube* dan sensor *GPS* pada *smartphone android* sudah sesuai dengan tujuan yang diharapkan.