

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini dijelaskan tentang implementasi sistem sesuai dengan perancangan yang dilakukan pada bab sebelumnya dan pengujian kesesuaian sistem yang dibangun dengan alur proses *booking* dan pembayaran non tunai studio yang diusulkan pada Studio Musik 18 Padang. Tahap implementasi dan tahap pengujian sistem dilakukan pada aplikasi *web* dan *mobile*.

5.1 Implementasi Sistem

Aplikasi sistem informasi pengelolaan *booking* dan pembayaran non tunai menggunakan teknologi QR Code berbasis *Web* dan *Mobile* ini difungsikan sebagai sistem yang mengelola dan menyajikan data dan informasi yang berkaitan dengan pembayaran dan data *member*. Pada sistem ini terdapat dua aktor yang terlibat dan dua belas fungsional yang dapat dijalankan. Fungsional tersebut mewakili setiap tugas dari *user* dan diharapkan dapat memberikan efisiensi dan efektifitas terhadap semua proses pada system.

Implementasi aplikasi *web* dibangun menggunakan bahasa pemrograman PHP (*Pearl Hypertext Preprocessor*) dan *framework* Laravel. *Framework* Laravel yang digunakan adalah Laravel versi 5.7.28 dan *web server apache* XAMPP versi 3.2.3. *Web server* berkomunikasi dengan perangkat lunak yang disebut dengan *middleware* dan *middleware* yang berhubungan dengan *database* (Habibie, Purnama, & Triyono, 2014). *Database* yang digunakan sebagai tempat penyimpanan data adalah *database* MySQL. Implementasi aplikasi *mobile* dibangun menggunakan Android Studio IDE versi 3.4. Android Studio dipilih karena memiliki keunggulan dibandingkan aplikasi pemrograman *mobile* lain, diantaranya karena pembangunan antar muka yang lebih mudah dan efisien, memiliki banyak *library* dan dapat mendukung semua perintah pengkodean *mobile* serta dilengkapi dengan emulator yang mendukung semua *devices*.

Implementasi sistem ini dilakukan dengan menggunakan perangkat keras komputer dengan spesifikasi sebagai berikut:

1. Komputer dengan *processor* Intel core i3-4010U CPU 1.70 GHz.
2. Memori Komputer (RAM) 4 GB.
3. *Harddisk* Komputer dengan kapasitas 465 GB.

Untuk implementasi aplikasi *mobile* digunakan perangkat *smartphone* dengan spesifikasi sebagai berikut:

1. *Processor* Octa-core Max 1.5 GHz.
2. Memori (RAM) 3 GB.
3. *Harddisk internal smartphone* 32 GB.

Spesifikasi perangkat lunak yang digunakan dalam implementasi sistem ini adalah sebagai berikut:

1. Sistem Operasi windows 10 pro
2. Sistem Operasi Android Lollipop 5.0.
3. *Web browser* yang digunakan adalah google chrome versi 73.

Berikut ini spesifikasi *system requirement* untuk implementasi sistem:

1. Sistem operasi Windows 7, 8, 8.1, 10.
2. *Web server* dengan modul PHP 7.3.2 atau setelahnya
3. *Web browser* yang mendukung HTML 5
4. *Android Operating System* versi 5.0 (Lollipop) atau setelahnya.

5.1.1 Pengkodean Program

Pada bagian pengkodean program ini dijelaskan kode program yang dibuat berdasarkan arsitektur aplikasi yang digunakan. Bagian *view*, *model*, dan *controller* dijelaskan secara terpisah berdasarkan fungsi dan kegunaan masing-masing. Berikut dijelaskan fungsi dan kegunaan masing-masing bagian pada aplikasi. Untuk penjelasan kode program lebih lanjut dapat dilihat pada lampiran G.

5.1.1.1 Kode Program *Routing* Aplikasi

Routing merupakan bagian yang mengatur arah perpindahan permintaan (*request*) dari *user*. *Route* dapat meng-*handle* semua perintah yang dideklarasikan dan mengirimkannya kepada *controller* serta *method* yang sesuai. Kode program *routing* aplikasi dapat dilihat pada Gambar 5.1.

```

Auth::routes();

Route::middleware('auth')->group(function (
){
Route::get('/', 'HomeController@index')->name('home');
Route::resource('users', 'UserController');
Route::resource('members', 'MemberController');
Route::resource('ruangan', 'RuanganController');
Route::resource('alat', 'AlatController');
Route::get('alat/created/{id}', 'AlatController@create')>name('alat.crea
te
d');
Route::get('histoty/topup/{id}', 'HistoryTopUpController@create')>name('
hi
storyTopup.create');
Route::post('histoty/store', 'HistoryTopUpController@store')>name('histo
ry
Topup.store');

Route::resource('transaksi', 'TransaksiController');
Route::get('detailTransaksi/create/{id}', 'DetailTransaksiController@cre
at
e')->name('detailTransaksi.creates');
Route::resource('detailTransaksi', 'DetailTransaksiController');
Route::get('getJadwal/{tanggal}/{jam}', 'TransaksiController@getJadwal')
-
>name('getJadwal');
Route::post('transaksi/tunai', 'TransaksiController@tunai')>name('transa
ks
i.tunai');
Route::get('formQr/{id}', 'TransaksiController@formQr')->name('formQr');
Route::post('bayarQr', 'TransaksiController@bayarQr')->name('bayarQr');
Route::get('print/{id}', 'TransaksiController@printNota')>name('transaks
i.
print');
Route::get('laporan', 'LaporanController@index')->name('laporan.index');
Route::post('laporan/filter', 'LaporanController@filter')>name('laporan.
fi
lter');
});

Route::prefix('mobile')->group(function () {
Route::get('login/{email}/{password}', 'MobileController@login');
Route::get('register/{name}/{email}/{password}', 'MobileController@regis
te
r');
Route::get('profil/{id}', 'MobileController@profil');
Route::get('profiledit/{id}/{namex}/{phonex}/{addressx}', 'MobileControl
le
r@profiledit');
Route::get('cekjadwalVIP/{date}/{time}', 'MobileController@cekjadwalVIP'
);
Route::get('cekjadwalRegular/{date}/{time}', 'MobileController@cekjadwal
Re
gular');
Route::get('getjadwal/{date}/{room}', 'MobileController@getjadwal');
Route::get('booking/{member_id}/{room}/{date}/{time}/{dp}', 'MobileContr
ol
ler@booking');
Route::get('history/{id}', 'MobileController@history');
Route::get('history_detail/{id}', 'MobileController@history_detail');
});

```

Gambar 5. 1 Kode Program Routing Aplikasi

5.1.1.2 Kode Program *Controller Top Up Saldo*

Controller merupakan bagian yang mengatur seluruh alur dan *method* pada aplikasi sistem informasi dengan menggunakan arsitektur MVC. Salah satu *controller* yang dibangun adalah *SaldoController*. Kode program *SaldoController* dapat dilihat pada Gambar 5.2.

```
<?php
namespace App\Http\Controllers;

use App\Models\HistoryTopUp;
use Carbon\Carbon;
use Illuminate\Http\Request;

class HistoryTopUpController extends Controller
{
    public function create($id){
        return view('history_top_ups.create',compact('id'));
    }

    public function store(Request $request){
        $input = $request->all();
        $last = HistoryTopUp::where('member_id',$request->
            >member_id)->orderBy('id')->first();
        $input['tanggal'] = Carbon::now();
        if (isset($last)){
            $sekarang = $last->saldo_sekarang;
        }else{
            $sekarang = 0;
        }

        if ($input['tipe'] == 1){
            $sekarang += $input['nominal'];
        }else{
            $sekarang = $sekarang - $input['nominal'];
        }
        $input['saldo_sekarang'] = $sekarang;
        HistoryTopUp::create($input);
        return redirect(route('members.show',['id' => $request->
            >member_id]));
    }
}
```

Gambar 5. 2 Kode Program *HistoryTopUpController*

5.1.1.3 Kode Program *Model Member*

Model merupakan bagian yang berfungsi memanggil data dari *database* serta mengirimkan data ke *controller* dengan menggunakan sintax SQL (*Structured Query Language*). Salah satu *model* yang dibangun adalah *model member* untuk memanggil entitas data member pada *database*. Kode program *model* pendaftaran dapat dilihat pada Gambar 5.3.

```
<?php
```

```

namespace App\Models;

use Eloquent as Model;

class Member extends Model
{
    public $table = 'member';

    const CREATED_AT = 'created_at';
    const UPDATED_AT = 'updated_at';

    public $fillable = [
        'user_id',
        'status',
        'alamat',
        'nama',
        'no_hp',
        'reg_date',
        'expired',
    ];

    protected $casts = [
        'id' => 'integer',
        'user_id' => 'integer',
        'status' => 'integer',
        'alamat' => 'string',
        'no_hp' => 'string',
        'nama' => 'string',
        'reg_date' => 'date',
        'expired' => 'date',
    ];

    public static $rules = [];
    public function user(){
        return $this->belongsTo(\App\User::class);
    }
    public function historyTopups(){
        return $this->hasMany(\App\Models\HistoryTopUp::class);
    }
    public function transaksis(){
        return $this->hasMany(\App\Models\Transaksi::class);
    }
    public function scopeGetLastSaldo($query, $id){
        return HistoryTopUp::where('member_id', $id)->orderBy(
            'id', 'DESC')->first();
    }
}

```

Gambar 5. 3 Kode Program Model Member

5.1.1.4 Kode Program View Transaksi

View merupakan antar muka aplikasi yang berinteraksi langsung dengan *user*. Data dari *database* dan perintah dari *controller* kemudian ditampilkan pada *view*. Salah satu *view* yang dibangun adalah *view* transaksi yang berfungsi untuk menampilkan halaman “transaksi”. Kode program *view* daftar dapat dilihat pada Gambar 5.4.

```

@extends('layouts.app')

@section('content')
  <div class="row">
    <div class="col-sm-12">
      <div class="card">
        <div class="card-header">
          <h3 class="pull-left">Transaksi</h3>
          <h3 class="pull-right">
            <a class="btn btn-sm btn-primary pull-
right" href="{!! route('transaksi.create') !!}">Add New</a>
          </h3>
        </div>
        <div class="card-body">
          <div class="table-stats order-table ov-h">
            @include('transaksi.table')
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
@endsection

```

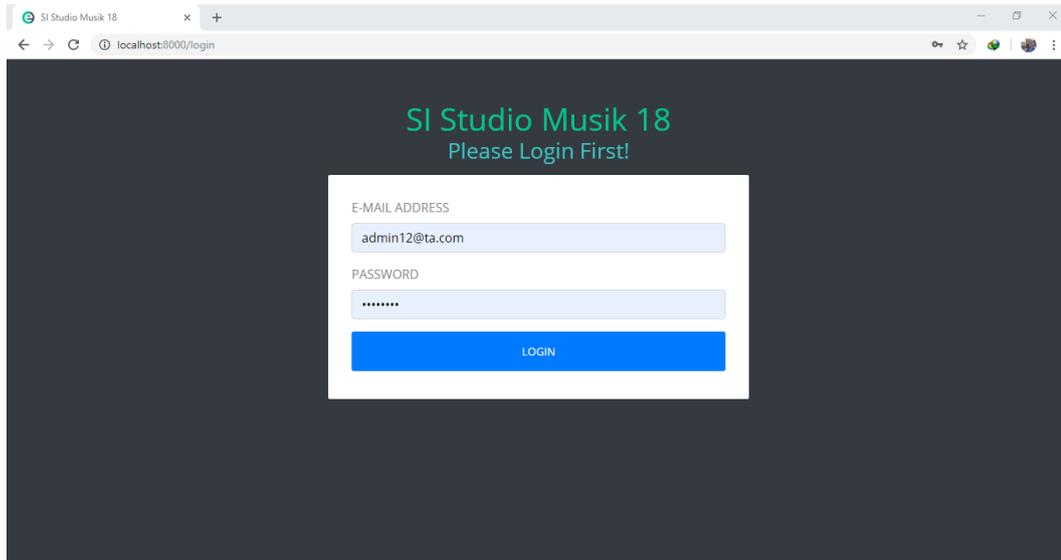
Gambar 5. 4 Kode Program View Transaksi

5.1.2 Implementasi Antar Muka Aplikasi Web

Aplikasi *web* yang dibangun dapat diakses oleh *user* menggunakan *web browser*. Berikut dijelaskan tampilan aplikasi *web* pembangunan aplikasi *e-Booking* dan pembayaran non tunai menggunakan teknologi QR Code untuk pengelolaan sewa studio band pada studio musik 18 sesuai dengan implementasi sistem. Untuk penjelasan antar muka lain dapat dilihat pada lampiran H.

5.1.2.1 Halaman Login

Halaman *login* merupakan halaman *website* yang pertama kali ditampilkan ketika *user* membuka aplikasi *web*. Hal ini dimaksudkan agar aplikasi ini hanya dapat digunakan oleh *user* yang terlibat dalam aplikasi *e-Booking* dan memiliki hak akses. Aplikasi ini memiliki satu level hak akses *user*, yaitu admin. Berikut tampilan halaman *login* dapat dilihat pada Gambar 5.5.

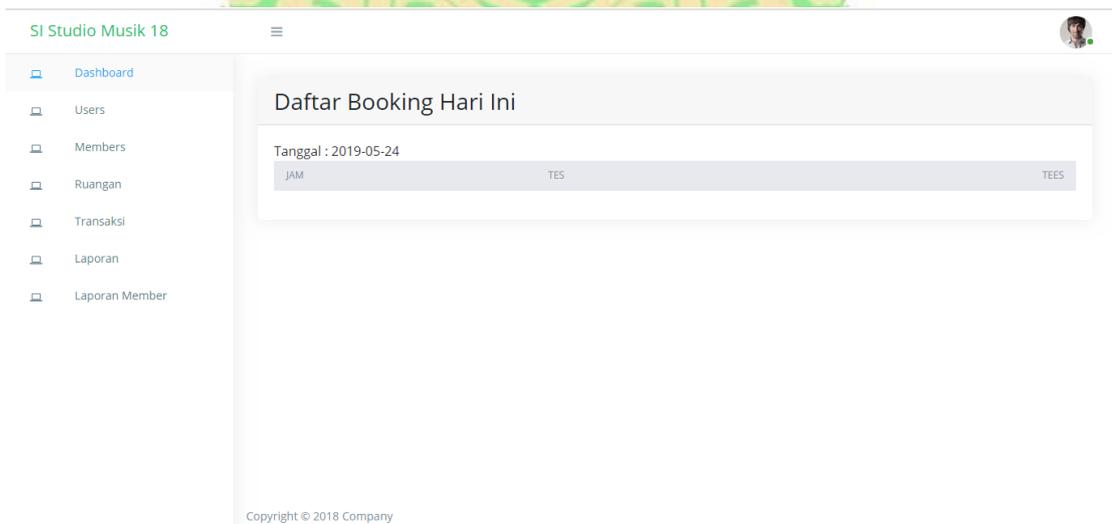


Gambar 5. 5 Tampilan Halaman *Login*

Berdasarkan Gambar 5.5 *user* diharuskan untuk *login* agar dapat masuk kedalam aplikasi dengan men-*input*-kan *email* dan *password* yang telah ditentukan pada *database* lalu menekan tombol “LOGIN”. Jika *user* berhasil *login* maka sistem menampilkan halaman utama aplikasi.

5.1.2.2 Halaman Admin

Halaman admin diberikan kepada seorang *user* dengan level hak akses sebagai admin. Admin memiliki wewenang dalam mengelola data *user*, data *member*, data ruangan, data transaksi dan data laporan. Tampilan halaman admin dapat dilihat pada Gambar 5.6.



Gambar 5. 6 Tampilan Halaman Admin

Berdasarkan Gambar 5.6 dapat dijelaskan bahwa halaman admin memiliki enam menu utama dan satu menu *logout* yang diuraikan sebagai berikut:

1. Menu *Dashboard*

Menu *dashboard* merupakan halaman utama yang menampilkan informasi data booking oleh member maupun non member pada hari ini.

2. Menu *User*

Menu *user* berfungsi sebagai menu untuk mengelola data akun yang ada pada sistem. *User* dapat menambah data, merubah data dan menghapus data terpilih.

3. Menu *Member*

Menu *member* berfungsi sebagai menu untuk mengelola akun *member*. *User* dapat menambah data, merubah data dan menghapus data terpilih.

4. Menu *Ruangan*

Menu *ruangan* berfungsi sebagai menu untuk mengelola ruangan studio beserta harga sewa. *User* dapat menambah, merubah dan menghapus data.

5. Menu *Transaksi*

Menu *transaksi* menampilkan halaman transaksi member maupun non member dan admin dapat melakukan pengelolaan pembayaran transaksi dengan menggunakan QR-Code maupun tunai

6. Menu *Laporan*

Menu *laporan* berfungsi untuk menampilkan semua transaksi yang sudah berstatus selesai dan masa aktif member dalam kurun waktu tertentu.

7. Menu *Logout*

Menu *logout* berfungsi untuk menutup halaman utama dan menampilkan halaman *login*.

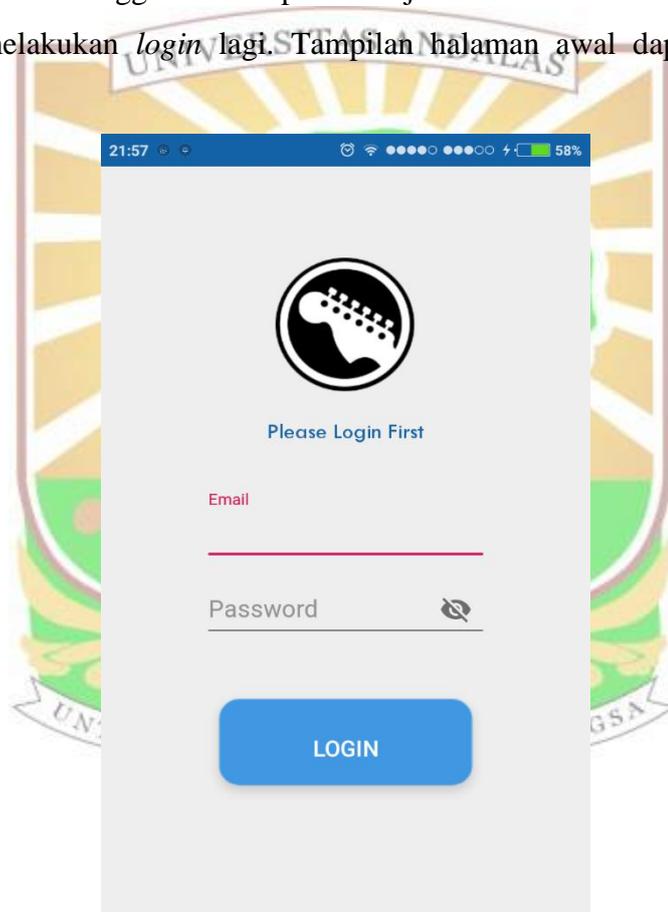
5.1.3 Implementasi Antar Muka Aplikasi *Mobile*

Aplikasi *mobile* merupakan salah satu hasil implementasi sistem informasi pembangunan aplikasi *e-Booking* dan pembayaran non tunai menggunakan teknologi QR Code untuk pengelolaan sewa studio band pada studio musik 18. Aplikasi *mobile* yang dibangun dapat diakses oleh *user* menggunakan *smartphone* atau *gadget*. Aplikasi *mobile* dapat digunakan oleh *user* yang memiliki level hak

akses *member*. Pada aplikasi *mobile* ini empat fungsional, yaitu booking studio, melihat ruangan, transaksi menggunakan *QR Code* dan edit profile.

5.1.3.1 Halaman *Login*

Halaman awal yang ditampilkan saat mengakses aplikasi *mobile* ini adalah halaman *login user*. Pada halaman ini *user* harus meng-*input*-kan *email* dan *password* untuk masuk kedalam aplikasi. Halaman ini muncul saat pertama kali aplikasi dijalankan, dimana setelah *login* maka aplikasi akan menyimpan data dari *member* tersebut sehingga ketika aplikasi dijalankan untuk selanjutnya *member* tidak perlu melakukan *login* lagi. Tampilan halaman awal dapat dilihat pada Gambar 5.7.

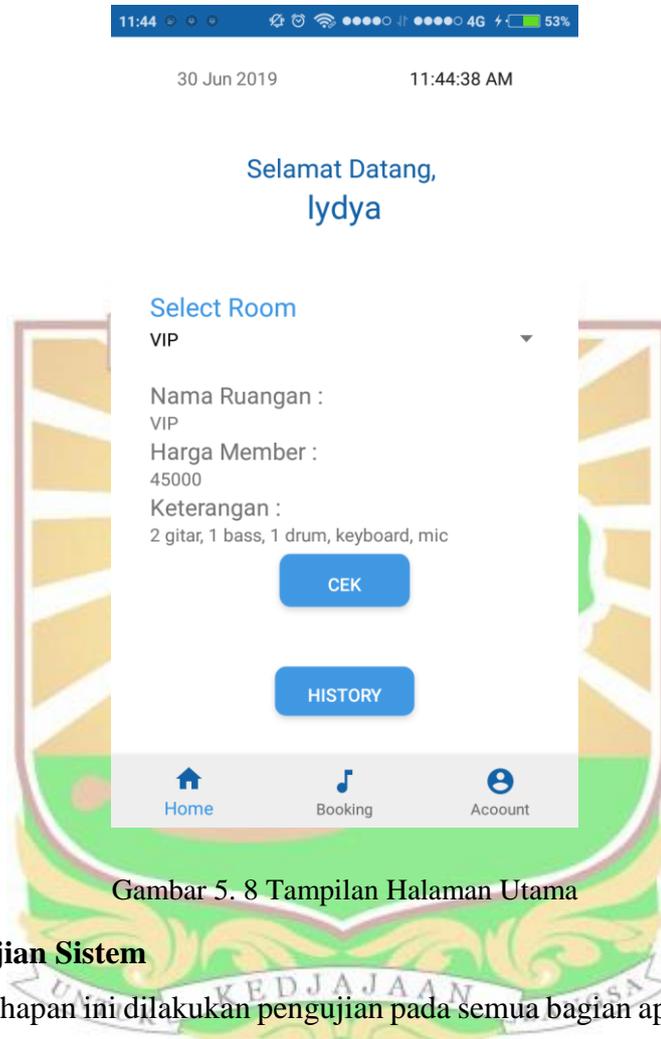


Gambar 5. 7 Tampilan Halaman Login

5.1.3.2 Halaman Utama

Setelah *member* melakukan *login*, maka muncul halaman utama. Pada halaman ini *member* dapat mencheck status ruangan studio dan melihat *history*

booking. Member juga dapat melihat data diri dan sisa saldo pada menu akun. Tampilan halaman utama dapat dilihat pada Gambar 5.8.



Gambar 5. 8 Tampilan Halaman Utama

5.2 Pengujian Sistem

Pada tahapan ini dilakukan pengujian pada semua bagian aplikasi yang telah dibangun. Proses ini dilakukan agar aplikasi yang dibangun berjalan sesuai dengan perancangan sistem. Pengujian ini dilakukan dengan menggunakan metode *black box testing*, yaitu metode pengujian yang berfokus pada pemeriksaan ketersediaan fungsional yang sudah dirancang pada aplikasi. Dengan menggunakan metode ini diketahui apakah fungsional aplikasi sudah berjalan sebagaimana mestinya dan sesuai dengan harapan atau tidak. Kelebihan metode ini dibandingkan metode *white box testing* adalah pada metode *black box testing* tidak diperlukan pengetahuan khusus tentang kode program aplikasi, sedangkan pada metode *white box testing*

penguji diharuskan untuk memeriksa komponen internal kode program aplikasi. Pengujian ini dilakukan oleh penulis dan pemilik Studio Musik 18.

5.2.1 Fokus Pengujian

Fokus pengujian aplikasi ini menggunakan data uji berdasarkan data yang telah didapatkan dari aplikasi *web* dan *mobile*. Pada pengujian ini terdapat dua belas item uji yang terdiri dari tujuh item uji aplikasi *web* dan lima item uji aplikasi *mobile*. Fokus Pengujian dapat dilihat pada Tabel 5.1 berikut ini.

Tabel 5. 1 Fokus Pengujian

No	Item Uji	Jenis Aplikasi	Detail Pengujian
1.	Mengelola data seluruh <i>member</i>	<i>Web</i>	Lihat, edit, hapus
2.	Mengelola pendaftaran <i>member</i>	<i>Web</i>	Lihat, tambah
3.	Mengelola tambah saldo	<i>Web</i>	Lihat, tambah
4.	Mengelola informasi ruangan studio	<i>Web</i>	Lihat, tambah, edit, hapus
5.	Mengelola transaksi pembayaran sewa studio dengan QR-Code	<i>Web</i>	Lihat, tambah, edit, hapus
6.	Melakukan cetak struk bukti pembayaran sewa studio	<i>Web</i>	Cetak
7.	Mengelola laporan	<i>Web</i>	Lihat, filter
8.	Mengelola Profil	<i>Mobile</i>	Lihat, <i>edit</i>
9.	Melihat status ruangan	<i>Mobile</i>	Lihat
10.	Melakukan <i>booking</i> studio band	<i>Mobile</i>	Tambah

11.	Melakukan transaksi pembayaran dengan QR-Code	<i>Mobile</i>	Lihat
-----	---	---------------	-------

Tabel 5.1. Fokus Pengujian (lanjutan)

12.	Melihat riwayat transaksi	<i>Mobile</i>	Lihat
-----	---------------------------	---------------	-------

5.2.2 Kasus Hasil Pengujian

Bagian ini membahas tentang kasus-kasus serta hasil dari pengujian sistem. Pengujian yang dilakukan berpedoman pada fokus pengujian yang telah ditentukan sebelumnya. Pengujian ini dilakukan berdasarkan fungsional dengan memperhatikan masukan ke sistem dan keluaran yang dihasilkan sistem. Pada bagian ini dijelaskan tiga kasus pengujian, yaitu pengujian tambah pendaftaran *member*, tambah saldo, dan *booking* studio musik.

5.2.2.1 Pengujian Tambah Pendaftaran *Member* (Web)

Pada pengujian ini dilakukan *input* data pendaftaran *member* pada aplikasi *web*. Hasil pengujian tambah data pendaftaran *member* dapat dilihat pada Tabel 5.2 dan Tabel 5.3.

Tabel 5. 2 Pengujian Tambah Pendaftaran Member (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Seluruh data yang dibutuhkan dalam halaman tambah <i>member</i>
Yang diharapkan	Sistem menampilkan halaman member dan list member yang telah mendaftar
Pengamatan	Menginput data pada halaman tambah pendaftaran <i>member</i> serta menekan tombol “Simpan”
Hasil	Sesuai

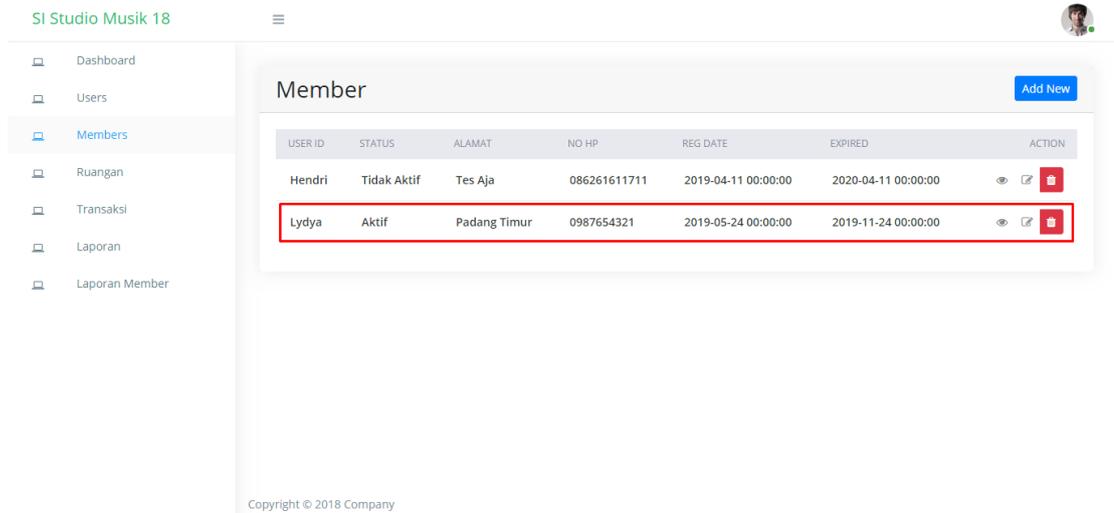
Tabel 5. 3 Pengujian Tambah Pendaftaran Member (Alternatif)

Kasus dan Hasil Uji (Alternatif)	
Data masukan	<i>User</i> menekan tombol “Simpan” sebelum mengisi <i>form</i>
Yang diharapkan	Sistem menampilkan notifikasi kesalahan
Pengamatan	<i>User</i> tidak bisa menambah data pendaftaran <i>member</i>
Hasil	Sesuai

A. Pengujian Tambah Pendaftaran *Member* (Benar)

Pengujian aplikasi dilakukan dengan memilih dan menginput data pendaftaran *member* pada halaman tambah *member* dan menekan tombol “Simpan”. Jika berhasil maka sistem menampilkan halaman *member* kembali dan menampilkan list *member* yang telah mendaftar. Hasil pengujian tambah pendaftaran *member* (benar) dapat dilihat pada Gambar 5.9 dan gambar 5.10.

Gambar 5. 9 Pengujian Masukan Tambah Pendaftaran Member (Benar)

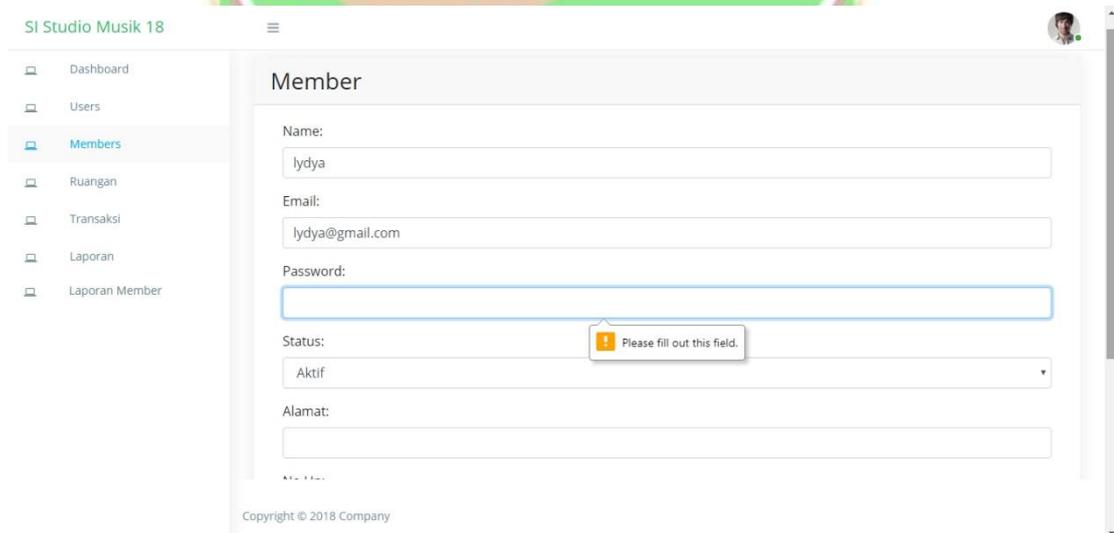


Gambar 5. 10 Pengujian Keluaran Tambah Pendaftaran Member (Benar)

Berdasarkan gambar 5.9 dan Gambar 5.10 dapat dilihat bahwa pengujian tambah pendaftaran *member* berhasil dilakukan.

B. Pengujian Tambah Pendaftaran *Member* (Alternatif)

Pengujian aplikasi dilakukan dengan menekan tombol “Simpan” sebelum mengisi *form*, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian tambah pendaftaran *member* (alternatif) dapat dilihat pada Gambar 5.11.



Gambar 5. 11 Pengujian Tambah Pendaftaran Member (Alternatif)

Berdasarkan Gambar 5.11 dapat dilihat bahwa sistem meminta *user* mengisi form yang telah disediakan sebelum menekan tombol “Simpan”.

5.2.2.2 Pengujian Tambah Isi Saldo (*Web*)

Pada pengujian ini dilakukan *input* data isi saldo pada aplikasi *web*. Hasil pengujian tambah isi saldo dapat dilihat pada Tabel 5.4 dan Tabel 5.5.

Tabel 5. 4 Pengujian Tambah Isi Saldo (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Seluruh data yang dibutuhkan dalam halaman tambah saldo
Yang diharapkan	Sistem menampilkan halaman lihat member kembali dan jumlah saldo yang bertambah.
Pengamatan	Dapat menginput data pada halaman tambah isi saldo serta menekan tombol “Simpan”
Hasil	Sesuai

Tabel 5. 5 Pengujian Tambah Isi Saldo (Alternatif)

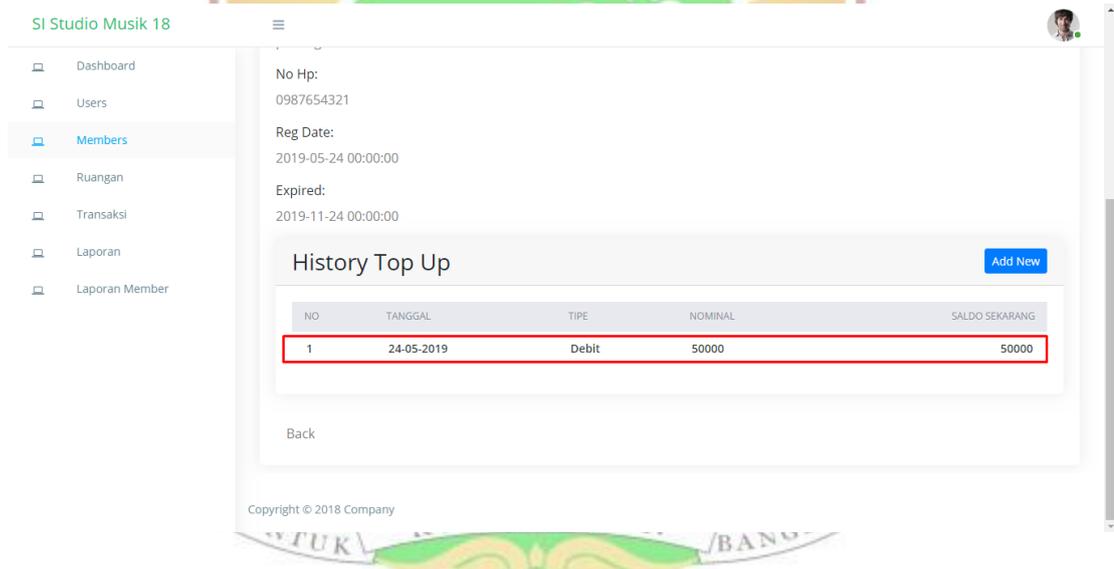
Kasus dan Hasil Uji (Alternatif)	
Data masukan	<i>User</i> menekan tombol “Simpan” sebelum mengisi <i>form</i>
Yang diharapkan	Sistem menampilkan notifikasi kesalahan
Pengamatan	<i>User</i> tidak bisa menambah saldo
Hasil	Sesuai

A. Pengujian Tambah Isi Saldo (Benar)

Pengujian aplikasi dilakukan dengan memilih dan menginput data isi saldo pada halaman tambah saldo dan menekan tombol “Simpan”. Jika berhasil maka sistem menampilkan halaman lihat member dan menampilkan jumlah saldo yang bertambah. Hasil pengujian tambah isi saldo (benar) dapat dilihat pada Gambar 5.12 dan Gambar 5.13.



Gambar 5. 12 Pengujian Masukan Tambah Isi Saldo (Benar)

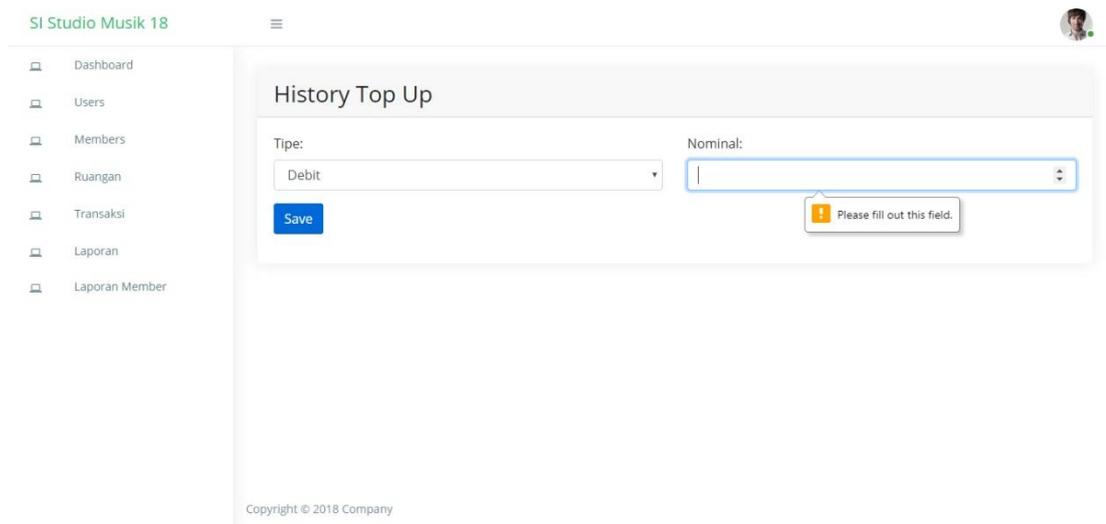


Gambar 5. 13 Pengujian Keluaran Tambah Isi Saldo (Benar)

Berdasarkan Gambar 5.12 dan Gambar 5.13 terlihat bahwa pengujian tambah data permintaan berhasil dilakukan dan sistem menampilkan keluaran yang benar.

B. Pengujian Tambah Isi Saldo (Alternatif)

Pengujian aplikasi dilakukan dengan menekan tombol “Simpan” sebelum mengisi *form*, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian tambah isi saldo (alternatif) dapat dilihat pada Gambar 5.14.



Gambar 5. 14 Pengujian Tambah Isi Saldo(Alternatif)

Berdasarkan Gambar 5.14 dapat dilihat bahwa sistem menampilkan pesan kesalahan pada *form* tambah isi saldo sebelum mengisi data ketika menekan tombol “Simpan”.

5.2.2.3 Pengujian *Booking Studio Musik (Mobile)*

Pada pengujian ini dilakukan *booking* studio musik. Hasil pengujian dapat dilihat pada Tabel 5.6 dan Tabel 5.7.

Tabel 5. 6 Pengujian Booking Studio Musik (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Member memilih tanggal dan jam <i>booking</i> dan mengisi <i>down payment</i>
Yang diharapkan	Sistem menampilkan halaman <i>booking</i>
Pengamatan	Dapat memilih tanggal dan jam yang diinginkan
Hasil	Sesuai

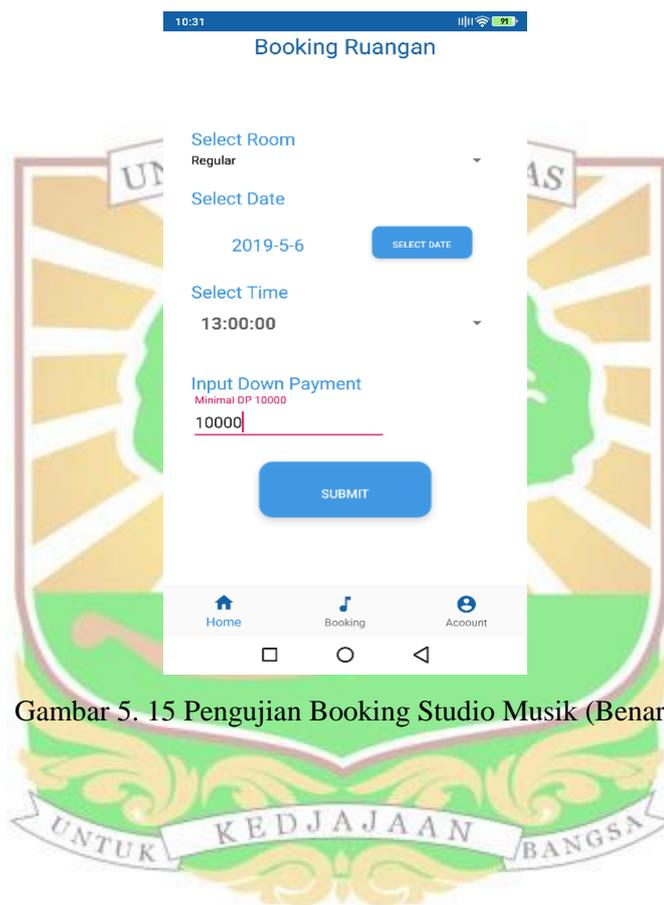
Tabel 5. 7 Pengujian Booking Studio Musik (Alternatif)

Kasus dan Hasil Uji (Alternatif)	
Data masukan	Member membuka menu <i>booking</i> dan data tidak ada
Yang diharapkan	Sistem tidak bisa melakukan proses pembayaran
Pengamatan	<i>User</i> tidak bisa melakukan proses <i>booking</i>

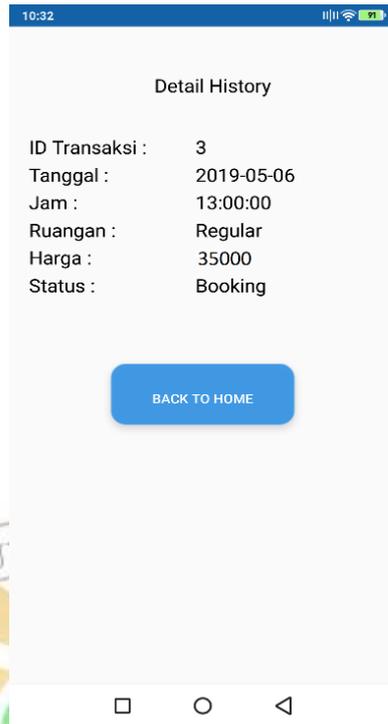
Hasil	Sesuai
-------	--------

A. Pengujian *Booking Studio Musik* (Benar)

Pengujian aplikasi dilakukan dengan memilih jam dan tanggal yang diinginkan dan mengisi *down payment*. Jika berhasil maka sistem menampilkan halaman *detail booking*. Hasil pengujian pembayaran pencucian mobil (benar) dapat dilihat pada Gambar 5.15 dan Gambar 5.16.



Gambar 5. 15 Pengujian Booking Studio Musik (Benar)

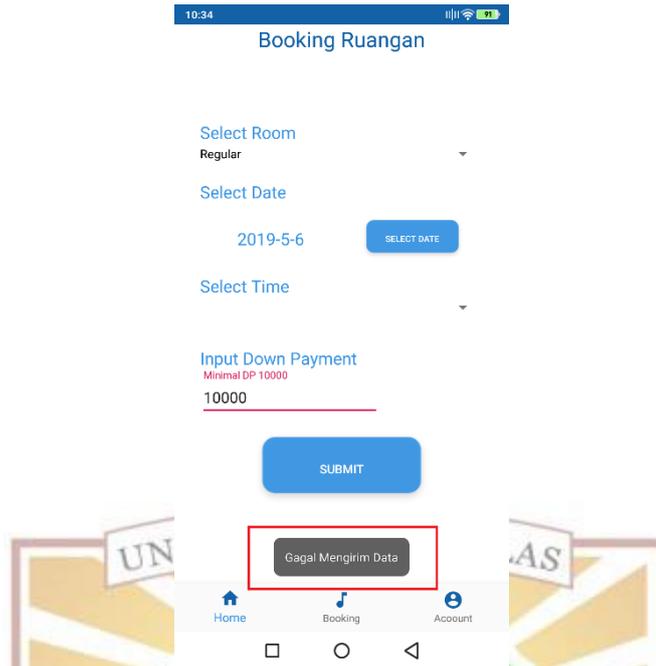


Gambar 5. 16 Pengujian Booking Studio Musik (Benar)

Berdasarkan Gambar 5.15 dan Gambar 5.16 terlihat bahwa pengujian berhasil dilakukan dan sistem menampilkan keluaran yang benar.

B. Pengujian *Booking* Studio Musik (Alternatif)

Pengujian aplikasi dilakukan dengan membuka menu *booking* dan data untuk melakukan proses pembayaran tidak ada. Hasil pengujian *booking* studio musik (alternatif) dapat dilihat pada Gambar 5.17.



Gambar 5. 17 Pengujian Booking Studio Musik (Alternatif)

Berdasarkan Gambar 5.17 terlihat bahwa sistem tidak bisa melakukan proses pembayaran karena tidak ada data yang akan diproses.

5.3 Kesimpulan Hasil Pengujian

Pengujian yang dilakukan terhadap aplikasi yang dibangun berfokus pada ketersediaan dan kesesuaian fungsional sistem yang diuji secara manual. Berdasarkan pengujian yang telah dilakukan diperoleh kesesuaian hasil antara perancangan dengan keluaran sistem, selain itu juga tidak ditemukan kegagalan dalam setiap proses pada masing-masing fungsional. Dapat disimpulkan bahwa pemanfaatan teknologi *QR Code* dalam pembangunan aplikasi *e-booking* dan pembayaran non tunai berbasis *web* dan *mobile* pada Studio Musik 18 telah berjalan sesuai dengan fungsional yang dirancang baik untuk aplikasi *web* maupun *mobile*. Hasil pengujian secara lengkap dijelaskan pada lampiran I. Kesimpulan dari hasil pengujian sistem dapat dilihat pada Tabel 5.8.

Tabel 5. 8 Hasil Pengujian

No	Fungsional	Skenario Pengujian		Hasil yang Diharapkan		Hasil
		Benar	Alternatif	Benar	Alternatif	
1	Mengelola informasi studio	Admin membuka menu ruangan, admin menjalankan fungsi tambah dan hapus.	Admin membuka menu ruangan (tidak ada data di <i>database</i>)	Sistem menampilkan seluruh data dan informasi tentang ruangan	Sistem menampilkan halaman ruangan dan data tidak ada	Sesuai
2	Mengelola registrasi <i>member</i>	Admin membuka menu <i>member</i> , admin memilih tombol tambah, mengisi form yang ada kemudian menekan tombol simpan	Admin membuka menu <i>member</i> , memilih tombol tambah, tidak mengisi form yang ada kemudian menekan tombol simpan	Sistem menampilkan kembali halaman <i>member</i> dan list <i>member</i> yang telah mendaftar	Sistem menampilkan notifikasi kesalahan input data	Sesuai

Tabel 5.8. Hasil Pengujian (lanjutan)

3.	Mengelola data <i>user</i>	Admin membuka menu user, admin bisa menjalankan fungsi tambah, edit, <i>detail</i> dan hapus	Admin membuka menu user, admin tidak bisa menambahkan data jika formnya belum diisi	Sistem menampilkan kembali halaman user dan menampilkan data user yang telah ditambahkan	Sistem menampilkan notifikasi kesalahan input data	Sesuai
4.	Melakukan pengisian saldo	Admin membuka menu member, memilih lihat pada member yang diinginkan, menekan tombol tambah <i>top up</i> dan menekan tombol save	Admin mengisi saldo dan tidak mengisi form yang ada kemudian menekan tombol simpan	Sistem menampilkan kembali halaman lihat member dan menampilkan history <i>top up</i> yang telah di tambahkan	Sistem menampilkan notifikasi kesalahan input data	Sesuai

Tabel 5.8. Hasil Pengujian (lanjutan)

5.	Mengelola transaksi pembayaran menggunakan <i>QR-Code</i>	Admin membuka menu transaksi, admin dapat menjalankan fungsi tambah, <i>edit</i> , <i>detail</i> dan hapus.	Admin membuka menu transaksi, memilih tombol tambah, tidak mengisi form yang ada kemudian menekan tombol simpan	Sistem menampilkan kembali halaman transaksi dan list transaksi yang telah diinputkan	Sistem menampilkan notifikasi kesalahan input data	Sesuai
6.	Mencetak bukti transaksi	Admin membuka menu transaksi tombol lihat, memilih jenis pembayaran, melakukan pembayaran dan cetak bukti transaksi	Admin memilih jenis pembayaran QR-Code dan saldo member tidak cukup	Sistem menampilkan kembali halaman cetak bukti transaksi	Sistem menampilkan notifikasi kesalahan	Sesuai

Tabel 5.8. Hasil Pengujian (lanjutan)

7	Melihat laporan transaksi	Admin membuka menu laporan, memilih rentang waktu yang diinginkan dan menekan tombol filter	Admin membuka menu laporan dan tidak ada riwayat transaksi pada rentang waktu di pilih	Sistem menampilkan laporan transaksi sesuai dengan rentang waktu yang dipilih	Sistem menampilkan tidak adanya laporan transaksi	Sesuai
8	Melakukan <i>booking</i> studio band	Member membuka menu <i>booking</i> dan memilih tanggal dan jam yang diinginkan	Member membuka menu <i>booking</i> dan memilih tanggal dan jam yang diinginkan tetapi telah di <i>booking</i>	Sistem akan menampilkan halaman utama <i>mobile</i>	Sistem menampilkan notifikasi kesalahan	Sesuai

Tabel 5.8. Hasil Pengujian (lanjutan)

9	Mengelola data profile	Member mengklik menu akun dan dapat menjalankan fungsi <i>edit</i>	Member membuka edit akun dan mengkosongkan salah satu form	Sistem menampilkan kembali halaman akun	Sistem menampilkan notifikasi kesalahan input data	Sesuai
10	Melakukan transaksi pembayaran menggunakan <i>QR-Code</i>	Member membuka menu akun dan memilih <i>QR-Code</i> untuk melakukan pembayaran	Member membuka <i>QR-Code</i> untuk pembayaran tetapi saldo <i>member</i> tidak cukup	Sistem menampilkan halaman status transaksi telah selesai	Sistem menampilkan notifikasi saldo tidak cukup	Sesuai
11	Melihat status ruangan	Member melihat menu home dan langsung melihat status ruangan untuk hari	Member melihat menu home dan tidak bisa melihat status ruangan	Sistem menampilkan status ruangan	Sistem tidak menampilkan status ruangan	Sesuai

		dan jam saat ini				
--	--	---------------------	--	--	--	--

Tabel 5.8. Hasil Pengujian (lanjutan)

12	Melihat riwayat booking	Member membuka history pada menu home	Member membuka menu history tetap tidak ada data	Sistem menampilkan riwayat booking member	Sistem tidak menampilkan riwayat booking member	Sesuai
----	-------------------------	---------------------------------------	--	---	---	--------

