

BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini membahas tentang implementasi kebutuhan pembangunan aplikasi dari hasil perancangan aplikasi yang dibahas pada bab sebelumnya. Setelah dilakukan tahap implementasi, maka dilanjutkan ke tahap pengujian sistem untuk meneliti kekurangan dan kesalahan pada sistem. Implementasi dan pengujian dilakukan pada aplikasi *web*.

5.1 Implementasi Sistem

Aplikasi pengelolaan pustaka elektronik ini difungsikan sebagai sistem yang mengelola dan menyajikan data yang berkaitan dengan pengelolaan pustaka. Pada sistem ini terdapat tiga aktor yang terlibat dan lima belas fungsional yang dapat dijalankan. Fungsional tersebut mewakili setiap tugas dari *user* dan diharapkan dapat memberikan efisiensi dan efektifitas terhadap semua proses pada sistem.

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

1. Komputer dengan *processor* Intel(R) Core(TM) i5-3337U CPU 1.80GHz
2. *Random Access Memory*(RAM) 4 GB
3. *Harddisk* komputer dengan kapasitas 500 GB

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

1. Sistem Operasi Windows 7 Pro
2. *Web browser* yang digunakan google chrome versi 1.26. 67
3. *Web Server* dengan modul PHP 7.4.20
4. *Web Browser* yang mendukung HTML 5

Aplikasi *web* dibangun menggunakan bahasa pemrograman PHP (*Pearl Hypertext Preprocessor*) dan *framework* Laravel. *Framework* Laravel yang digunakan adalah Laravel versi 7.X, dan *web server apache* XAMPP versi 2.4.48. *Web server* berfungsi untuk menjalankan bahasa pemrograman PHP pada *browser* dan mengolah data pada *database*. *Database* yang digunakan sebagai tempat penyimpanan data adalah *database* MySQL

5.1.1 Pengkodean Program

Berdasarkan arsitektur aplikasi MVC yang digunakan, pada bagian ini dijelaskan kode program yang dibuat. Bagian *view*, *model*, dan *controller* dijelaskan secara terpisah berdasarkan fungsional 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*

Router berfungsi menangani *request* aplikasi kemudian mengarahkan aplikasi untuk memanggil halaman/*resource* tertentu. *Routing* adalah bagian yang mengatur arah perpindahan permintaan (*request*) dari *user*. *Route* dapat *handle* semua perintah yang dideklarasikan dan mengirimkannya kepada *controller* serta *method* yang sesuai. Kode program *routing* aplikasi dapat dilihat pada Gambar 5.1.

```
<?php

use App\Http\Controllers\GuestController;
use Illuminate\Support\Facades\Route;
use Illuminate\Support\Facades\Auth;

Route::get('/', 'GuestController@index');
Route::get('/member', function () {
    return view('guest.checkin');
});
Route::get('/member/{no}/checkin', 'GuestController@checkin');
Route::get('/guest', 'GuestController@guest');
Route::post('/guest', 'GuestController@guestPost');

Auth::routes();

Route::get('/logout', 'HomeController@logout');
Route::get('/admin', 'HomeController@index')->name('home');
Route::get('/admin/report', 'HomeController@report');

Route::get('/admin/ddcs', 'DdcController@index');
Route::get('/admin/ddcs/{num}', 'DdcController@select');
Route::get('/admin/ddcs/{num}/print', 'DdcController@print');
Route::get('/admin/ddc/edit/{num}', 'DdcController@edit');
Route::post('/admin/ddc/edit/{num}', 'DdcController@update');
Route::get('/admin/ddc/detail/{num}', 'DdcController@searchBook');
Route::get('/admin/ddc/print/{num}', 'DdcController@printDdc');
Route::get('/admin/ddc/search', 'DdcController@search');

Route::get('/admin/bookkeepings', 'BookkeepingController@index');
Route::get('/admin/bookkeepings/print', 'BookkeepingController@print');
Route::get('/admin/bookkeeping/new', 'BookkeepingController@create');
Route::post('/admin/bookkeeping/new', 'BookkeepingController@store');
Route::get('/admin/bookkeeping/edit/{no}',
'BookkeepingController@edit');

Route::post('/admin/bookkeeping/edit/{no}',
'BookkeepingController@update');
```

Gambar 5.1 Kode Program *Routing Aplikasi*

```

Route::get('/admin/bookkeeping/book/{no}',
'BookkeepingController@listBook');
Route::get('/admin/bookkeeping/print/{no}',
'BookkeepingController@printBook');

Route::get('/admin/books', 'BookController@index');
Route::get('/admin/books/search', 'BookController@search');
Route::get('/admin/books/print', 'BookController@printAll');
Route::get('/admin/book/{id}/detail', 'BookController@show');
Route::get('/admin/book/{id}/print', 'BookController@print');

Route::get('/admin/book/new', 'BookController@create');
Route::get('/admin/book/new/ddc/{ddc}', 'BookController@createWithDdc');
Route::get('/admin/book/new/bookkeeping/{bookkeeping}',
'BookController@createWithBookkeeping');
Route::post('/admin/book/new', 'BookController@store');
Route::get('/admin/book/edit/{no}', 'BookController@edit');
Route::post('/admin/book/edit/{no}', 'BookController@update');

Route::get('/admin/members/registered', 'MemberController@index');
Route::get('/admin/members/unregistered',
'MemberController@unregistered');
Route::get('/admin/members/{stat}/search', 'MemberController@search');

Route::get('/admin/member/{no}/detail', 'MemberController@show');
Route::get('/admin/member/{no}/print', 'MemberController@print');
Route::get('/admin/member/{no}/accept', 'MemberController@accept');
Route::get('/admin/member/{no}/delete', 'MemberController@destroy');
Route::get('/admin/member/{no}/edit', 'MemberController@edit');
Route::post('/admin/member/{no}/edit', 'MemberController@update');

Route::get('/admin/visitors', 'VisitorController@index');
Route::get('/admin/visitors/search', 'VisitorController@search');
Route::get('/admin/visitors/print', 'VisitorController@print');

Route::get('/admin/borrows', 'BorrowController@index');
Route::get('/admin/borrows/search', 'BorrowController@search');
Route::get('/admin/borrows/print', 'BorrowController@print');

Route::get('/admin/borrow/{id}/return',
'BorrowController@returnBorrow');
Route::get('/admin/borrow/{id}/fine', 'BorrowController@fineMaker');
Route::get('/admin/borrow/{id}/pay', 'BorrowController@finePay');
Route::get('/admin/borrow/new', function () {
    return view('admin.borrow.new');
});

Route::get('/admin/returns', function () {
    return view('admin.borrow.return');
});

Route::get('/admin/borrow/make/{no}/{id}', 'BorrowController@create');

```

Gambar 5.1 Kode Program *Routing* Aplikasi(Lanjutan)

5.1.1.2 Kode Program *Controller*

Controller merupakan bagian yang mengatur seluruh alur dan *method* pada aplikasi sistem informasi dengan menggunakan arsitektur MVC. Salah satu *controller*

yang dibangun adalah *BookkeepingController*. *BookkeepingController* berisi kode program untuk mengolah data Pembukuan. Kode *BookkeepingController* dapat dilihat pada Gambar 5.2.

```

<?php

namespace App\Http\Controllers;

use App\Bookkeeping;
use App\Book;
use Illuminate\Support\Facades\DB;
use Illuminate\Http\Request;
use Illuminate\Support\Facades\Validator;

class BookkeepingController extends Controller
{
    public function __construct()
    {
        $this->middleware('auth');
    }

    public function index()
    {
        $bookkeepingsRes = Bookkeeping::orderBy('tanggal', 'desc')
            ->paginate(10);

        foreach ($bookkeepingsRes as $bookkeeping) {
            $bookCount = Book::select(DB::raw('count(*) as total'))
                ->where('no_induk', '=', $bookkeeping->no_induk)
                ->first();

            $bookkeeping->count = $bookCount->total;
        }

        return view('admin.bookkeeping.table', compact('bookkeepingsRes'));

        public function print()
        {
            $bookkeepingsRes = Bookkeeping::get();

            foreach ($bookkeepingsRes as $bookkeeping) {
                $bookCount = Book::select(DB::raw('count(*) as total'))
                    ->where('no_induk', '=', $bookkeeping->no_induk)
                    ->first();

                $bookkeeping->count = $bookCount->total;
            }

            return view('admin.bookkeeping.print', compact('bookkeepingsRes'));

            public function create()
            {
                $add = true;
                return view('admin.bookkeeping.form', compact('add'));
            }
        }
    }
}

```

Gambar 5.2 Kode Program *Controller* Aplikasi

```

public function listBook($bookkeeping)
{
    $link = str_replace("&", "/", $bookkeeping);
    $booksRes = Book::where('no_induk', $link)->paginate(10);

    $bookkeepingInfo = Bookkeeping::find($link);

    session()->forget('forms.title');
    session()->forget('forms.author');
    session()->put('forms.ddc', "10");

    return view('admin.book.table', compact('booksRes',
'bookkeepingInfo'));
}

public function store()
{
    $validator = Validator::make(request()->all(), [
        'no_induk' => array(
            'required',
            'unique:bookkeepings',
            'regex:/^[\\0-9A-Z.]+$/u'
        ),
        'tanggal' => 'required|max:255',
        'sumber' => 'required|max:255',
    ]);
    if ($validator->fails()) {
        return redirect()->back()->withErrors($validator)-
>withInput();
    }

    $bookkeeping = request('no_induk');

    $bookkeepingRes = new Bookkeeping;
    $bookkeepingRes->no_induk = request('no_induk');
    $bookkeepingRes->tanggal = request('tanggal');
    $bookkeepingRes->sumber = request('sumber');
    $bookkeepingRes->save();

    return redirect("admin/bookkeepings")->with('success',
"Pembukuan
$bookkeeping berhasil dibuat");
}

public function edit($bookkeeping)
{
    $link = str_replace("&", "/", $bookkeeping);
    $bookkeepingRes = Bookkeeping::find($link);
    $edit = true;
    return view('admin.bookkeeping.form', compact('bookkeepingRes',
'edit'));
}

public function update($bookkeeping)
{
    $validator = Validator::make(request()->all(), [
        'no_induk' => array(
            'required',
            'unique:bookkeepings',
            'regex:/^[\\0-9A-Z.]+$/u'
        ),
    ],

```

Gambar 5.2 Kode Program *Controller* Aplikasi(Lanjutan)

```

'tanggal' => 'required|max:255',
    'sumber' => 'required|max:255',
]);
if ($validator->fails()) {
    return redirect()->back()->withErrors($validator)-
>withInput();
}

$link = str_replace("&", "/", $bookkeeping);

$bookkeepingRes = Bookkeeping::find($link);
$bookkeepingRes->no_induk = request('no_induk');
$bookkeepingRes->tanggal = request('tanggal');
$bookkeepingRes->sumber = request('sumber');
$bookkeepingRes->save();

return redirect("admin/bookkeepings")->with('success',
"Pembukuan $link berhasil diperbarui");
}

public function printBook($bookkeeping)
{
    $link = str_replace("&", "/", $bookkeeping);
    $booksRes = Book::where('no_induk', $link)->get();

    $bookkeepingInfo = Bookkeeping::find($link);

    return view('admin.book.printAll', compact('booksRes',
'bookkeepingInfo'));
}
}

```

Gambar 5.2 Kode Program *Controller* Aplikasi(Lanjutan)

5.1.1.3 Kode Program Model

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

```

<?php

namespace App;

use Illuminate\Database\Eloquent\Model;

class Borrow extends Model
{
    public $timestamps = false;
    protected $primaryKey = 'borrow_id';
    protected $fillable = [
        'member_no',
        'book_id',
        'status_denda',
        'admin_username',
        'tanggal_peminjaman',

```

Gambar5.3 Kode Program *Model* Peminjaman

```

        'tanggal_pengembalian',
        'tanggal_pembayaran'
    ];
    public function memberDetail()
    {
        return $this->belongsTo('App\Member', 'member_no');
    }

    public function bookDetail()
    {
        return $this->belongsTo('App\Book', 'book_id');
    }

    public static function getBook()
    {
        return static::leftJoin(
            'books',
            'books.book_id',
            '=',
            'borrows.book_id'
        );
    }
}

```

Gambar 5.3 Kode Program Model Peminjaman(Lanjutan)

5.1.1.4 Kode Program View

View merupakan antarmuka aplikasi yang berinteraksi langsung dengan *user*. Data dari *database* dan perintah dari *controller* kemudian ditampilkan pada *view*. Salah satu *view* yang dibangun adalah *view member/form.php* yang berfungsi untuk menampilkan halaman “Formulir Member”. *View data admin* ini berisi *script* untuk memanggil menampilkan halaman untuk mengubah atau menambahkan data anggota perpustakaan. Kode program data *admin* dapat dilihat pada Gambar 5.4.

```

@extends('layouts.app')

@section('content')
    <div class="container">
        <div class="row justify-content-center">
            <div class="col-md-12">
                <div class="card">
                    <div class="card-body">
                        @if (count($errors) > 0)
                            @foreach ($errors->all() as $error)
                                <div class="alert alert-danger alert-
dismissible fade show" role="alert">
                                    {{ $error }}
                                    <button type="button" class="close"
data-dismiss="alert" aria-label="Close">
                                        <span aria-
hidden="true">&times;</span>
                                </div>
                            @endforeach
                        @endif
                    </div>
                    <form method="POST"
action="/admin/member/{{ $memberRes->member_no }}/edit"
enctype="multipart/form-data">
                        @csrf
                    </div class="row">

```

Gambar 5.4 Kode Program View anggota perpustakaan

```

<div class="col-md-6">
  <div class="form-group">
    <label>Nama</label>
    <input type="text" class="form-
control" name="nama" value="{{ $memberRes->nama }}"
required autofocus
oninvalid="this.setCustomValidity('Data dibutuhkan') "
oninput="this.setCustomValid
ity('') ">
  </div>
</div>
<div class="col-md-3">
  <div class="form-group">
    <label>Tempat Lahir</label>
    <input type="text" class="form-
control" name="tempat_lahir"
value="{{ $memberRes-
>tempat_lahir }}" required autofocus
oninvalid="this.setCustomVal
idity('Data dibutuhkan') "
oninput="this.setCustomValid
ity('') ">
  </div>
</div>
<div class="col-md-3">
  <div class="form-group">
    <label>Tanggal Lahir</label>
    <input type="date" class="form-
control" name="tanggal_lahir"
value="{{ $memberRes-
>tanggal_lahir }}" required autofocus
oninvalid="this.setCustomVal
idity('Data dibutuhkan') "
oninput="this.setCustomValid
ity('') ">
  </div>
</div>
<div class="col-md-12">
  <div class="form-group">
    <label>Alamat</label>
    <input type="text" class="form-
control" name="alamat"
value="{{ $memberRes-
>alamat }}" required autofocus
oninvalid="this.setCustomVal
idity('Data dibutuhkan') "
oninput="this.setCustomValid
ity('') ">
  </div>
</div>
<div class="col-md-6">
  <div class="form-group">
    <label>Pekerjaan</label>
    <select id="bookkeepingForm"
class="form-control" name="pekerjaan">
      @foreach ($jobsRes as $job)
        <option value="{{ $job-
>job_id }}" @if (old('pekerjaan') == $job->job_id || $memberRes->job_id
== $job->job_id) selected="selected" @endif>
          {{ $job-
>pekerjaan }}</option>
      @endforeach
    </select>
  </div>
</div>
<div class="col-md-6">
  <div class="form-group">

```

Gambar 5.4 Kode Program *View Member*(Lanjutan)


```

<label>Instansi</label>
<input type="text" class="form-
control" name="nama_instansi"
value="{{ $memberRes-
>nama_instansi }}" required autofocus
oninvalid="this.setCustomVal
idty('Data dibutuhkan') "
oninput="this.setCustomValid
ity('')">
</div>
</div>
<div class="col-md-6">
<div class="form-group">
<label>Nomor Telepon</label>
<input type="text" class="form-
control" name="telepon_no"
value="{{ $memberRes-
>telepon_no }}" required autofocus
oninvalid="this.setCustomVal
idty('Data dibutuhkan') "
oninput="this.setCustomValid
ity('')">
</div>
</div>
<div class="col-md-6">
<div class="form-group">
<label>Nomor Identitas</label>
<input type="text" class="form-
control" name="identitas_no"
value="{{ $memberRes-
>identitas_no }}" required autofocus
oninvalid="this.setCustomVal
idty('Data dibutuhkan') "
oninput="this.setCustomValid
ity('')">
</div>
</div>
<div class="col-md-6">
<label>Pas Foto</label>
<input type="file" name="foto_file"
class="form-control">
</div>
<div class="col-md-6">
<label>Identitas</label>
<input type="file"
name="identitas_file" class="form-control">
</div>
</div>
<div class="form-group pull-right">
<button class="btn btn-success">
<i class="fa fa-check"></i>
Simpan
</button>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
@endsection

```

Gambar 5.4 Kode Program View Anggota Perpustakaan(Lanjutan)

5.1.2 Implementasi Antarmuka Aplikasi Web

Pada bagian ini dijelaskan tampilan aplikasi *web* dalam pembangunan aplikasi pengelolaan pustaka elektronik pada Dinas Perpustakaan dan Kearsipan Kota Padang. Aplikasi *web* ini dapat diakses oleh pegawai perpustakaan dan pengunjung. Halaman awal yang ditampilkan saat mengakses aplikasi web ini adalah halaman kunjungan(*Check-in*). Setelah itu barulah halaman *login*. Jika *user*(pegawai) berhasil *login*, maka akan ditampilkan halaman utama *admin*. Berikut dijelaskan tampilan aplikasi *web* dari sistem yang dibangun sesuai dengan implementasi sistem. Untuk penjelasan antarmuka lain dapat dilihat pada lampiran H

5.1.2.1 Halaman *Check-in*

Halaman *Check-in* adalah halaman yang pertama kali ditampilkan ketika pengunjung akan mengisi buku tamu. Halaman ini bertujuan untuk mencatat data pengunjung yang berkunjung pada hari itu, baik itu anggota perpustakaan ataupun bukan anggota perpustakaan. Tampilan Halaman *Check-in* dapat dilihat pada Gambar 5.5

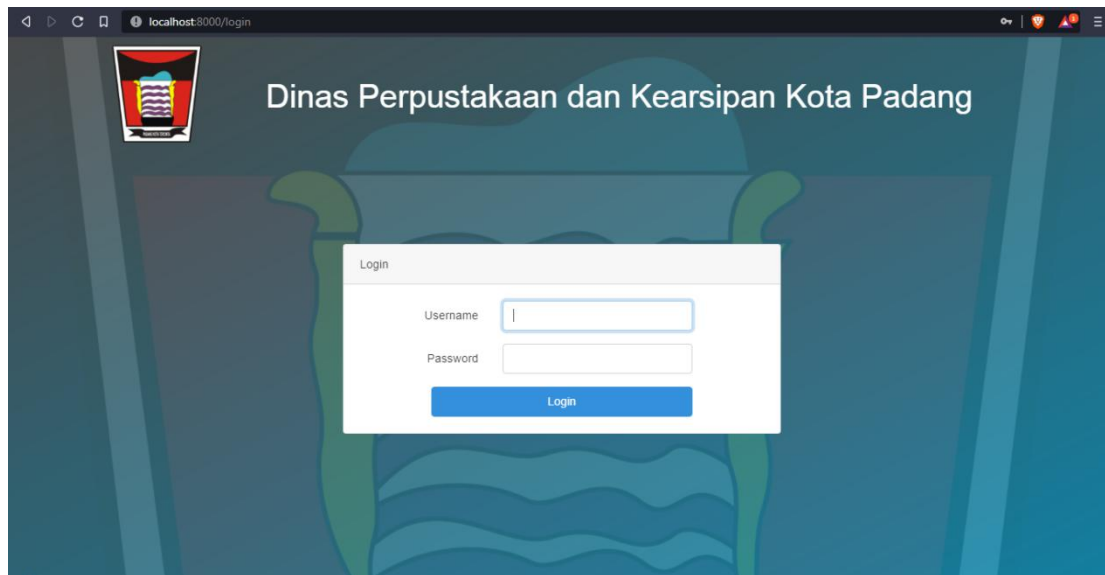


Gambar 5.5 Tampilan Halaman *Check-in*

5.1.2.2 Halaman Login Admin

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 pengelolaan pustaka dan memiliki hak akses.

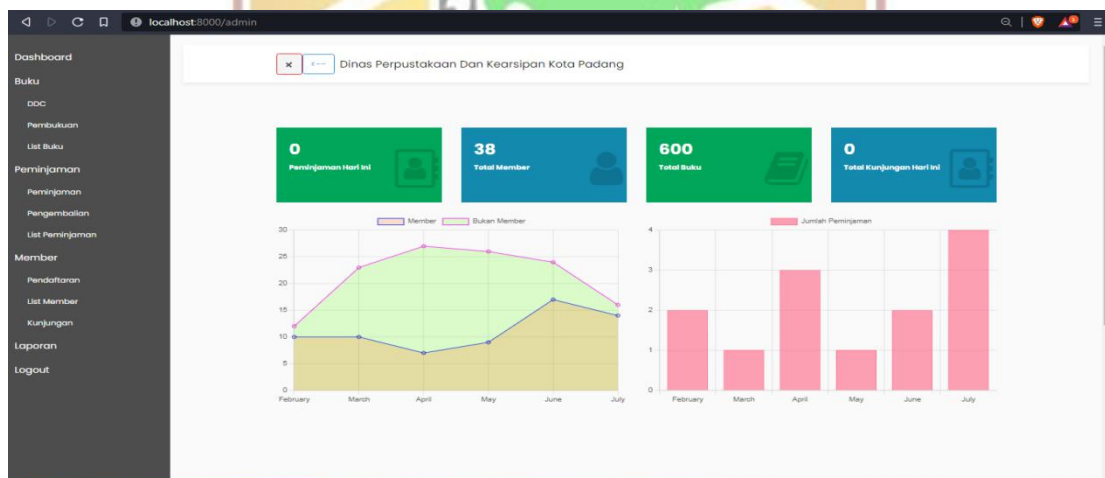
Aplikasi ini memiliki satu level hak akses *user*, yaitu pegawai perpustakaan. Berikut tampilan halaman *login* dapat dilihat pada Gambar 5.6.



Gambar 5.6 Tampilan Halaman *login*

5.1.2.3 Halaman Admin

Halaman admin diberikan kepada seorang *user* dengan level hak akses sebagai Pegawai perpustakaan. Pegawai memiliki wewenang dalam mengelola data buku, data pembukuan, data anggota perpustakaan, data kunjungan dan data laporan. Tampilan halaman pegawai dapat dilihat pada Gambar 5.7



Gambar 5.7 Halaman Pegawai Perpustakaan

Berdasarkan Gambar 5.7 dapat dijelaskan bahwa halaman pegawai perpustakaan memiliki

enam menu utama dan satu menu *logout* yang diuraikan sebagai berikut:

1. Menu *Dashboard*

Menu *dashboard* merupakan halaman utama yang menampilkan informasi data kunjungan, jumlah anggota perpustakaan dan buku yang sering dipinjam.

2. Menu *Buku*

Menu buku berfungsi sebagai menu untuk mengelola data DDC, pembukuan dan DDC. *User* dapat melihat dan mengedit data yang terpilih.

3. Menu Peminjaman

Menu peminjaman berfungsi sebagai menu untuk mengelola data peminjaman dan pengembalian buku. *User* dapat melakukan peminjaman buku yang ingin dipinjam oleh anggota perpustakaan dan *user* juga dapat melakukan pengembalian buku yang dikembalikan oleh anggota perpustakaan. *User* dapat melakukan peminjaman dan pengembalian dengan menggunakan QR-Code maupun dengan input manual.

4. Menu anggota perpustakaan

Menu anggota perpustakaan berfungsi sebagai menu untuk mengelola anggota perpustakaan, pendaftaran dan kunjungan. Pada sub-menu anggota perpustakaan, *user* hanya dapat melihat dan mengubah data terpilih. Pada sub-menu pendaftaran *user* dapat melihat, menghapus, mengedit dan verifikasi data terpilih, sedangkan pada sub-menu kunjungan *user* dapat melihat kunjungan.

5. Menu Laporan

Menu laporan berfungsi untuk menampilkan dan mencetak laporan pengunjung dan laporan peminjaman buku.

7. Menu *Logout*

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

5.2 Pengujian Sistem

Tahap pengujian sistem merupakan proses untuk memeriksa apakah aplikasi yang dihasilkan telah berjalan sesuai dengan perancangan sistem yang telah ditetapkan. Pengujian ini dilakukan dengan menggunakan metode *black box testing*, yaitu metode pengujian yang berfokus pada pemeriksaan ketersediaan fungsional yang sudah dirancang pada aplikasi. 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, pegawai serta salah satu pengunjung dari Dinas Perpustakaan dan Kearsipan Kota Padang.

5.2.1 Fokus Pengujian

Fokus pengujian aplikasi ini menggunakan data uji berdasarkan data yang telah didapat dari aplikasi *web*. Pada pengujian ini terdapat lima belas item uji aplikasi *web*. Fokus pengujian dapat dilihat pada tabel 5.1 berikut

Tabel 5.1 Fokus Pengujian

No	Item Uji	Jenis Aplikasi	Detail Pengujian
1	<i>Login/Logout</i>	<i>Web</i>	Verifikasi data <i>Login</i>
2	Mengelola data anggota perpustakaan	<i>Web</i>	Lihat, edit
3	Melihat dan mencetak kartu anggota perpustakaan	<i>Web</i>	Lihat, cetak
4	Melihat dan mencetak laporan kunjungan	<i>Web</i>	Lihat, cetak
5	Mengelola data peminjaman dan pengembalian buku	<i>Web</i>	Lihat, tambah
6	Mengelola data pembukuan	<i>Web</i>	Lihat, edit, tambah, cetak
7	Melihat dan mencetak laporan pembukuan dan list buku	<i>Web</i>	Lihat, cetak
8	Melihat dan mencetak laporan peminjaman dan pengembalian buku	<i>Web</i>	Lihat, cetak
9	Melakukan proses peminjaman dan pengembalian buku dengan QR-Code	<i>Web</i>	Tambah, scan
10	Anggota perpustakaan mengisi buku tamu dengan menggunakan QR-Code ataupun dengan input manual	<i>Web</i>	Tambah, scan
11	Pengunjung bukan tamu mengisi buku tamu dengan input manual	<i>Web</i>	Tambah
12	Melihat daftar pengunjung	<i>Web</i>	Lihat
13	Melihat grafik pengunjung dan grafik buku yang sering dipinjam	<i>Web</i>	Lihat
14	Mendaftar menjadi anggota perpustakaan	<i>Web</i>	Lihat, tambah
15	Mencari buku pada list buku	<i>Web</i>	Lihat

Tabel 5.1 Fokus Pengujian(Lanjutan)

5.2.2 Kasus Hasil Pengujian

Pada bagian ini ditampilkan hasil dari pengujian sistem. Pengujian yang dilakukan berpedoman pada fokus pengujian yang telah ditentukan sebelumnya. Pengujian dilakukan berdasarkan fitur-fitur pada sistem yang dibangun dengan memperhatikan masukan ke sistem dan keluaran yang dihasilkan sistem. Di bagian ini dijelaskan tiga kasus pengujian, yaitu pengujian registrasi anggota perpustakaan baru, cetak kartu anggota dan menambah data peminjaman buku

5.2.2.1 Pengujian Registrasi Anggota Perpustakaan Baru

Pada pengujian ini dilakukan input data calon anggota pada aplikasi *web*. Hasil pengujian registrasi anggota perpustakaan dapat dilihat pada Tabel 5.2 dan Tabel 5.3.

Tabel 5.2 Pengujian Registrasi Anggota Perpustakaan Baru(Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Semua data yang dibutuhkan dalam form registrasi “anggota perpustakaan baru”
Yang diharapkan	Data tersimpan dan sistem menampilkan halaman formulir pendaftaran
Pengamatan	Data berhasil tersimpan dan sistem menampilkan halaman formulir pendaftaran
Hasil	Sesuai

Pengujian dilakukan dengan menekan tombol “Daftar” pada halaman daftar anggota perpustakaan, lalu akan muncul form input “Daftar Anggota Perpustakaan Baru”. Setelah itu masukkan data sesuai dengan form dan menekan tombol “Daftar”. Jika berhasil, sistem akan menampilkan halaman formulir pendaftaran. Tampilan hasil pengujian registrasi anggota perpustakaan(benar) dapat dilihat pada Gambar 5.8 dan 5.9.

DINAS PERPUSTAKAAN DAN KEARSIPAN KOTA PADANG

BERANDA CARI BUKU DAFTAR MEMBER

Nama
Nora Savira

Tempat / Tanggal Lahir
Padang 08/05/1998

Alamat
jalan pangambiran

Pekerjaan
4.Mahasiswa

Nama Instansi
UNP

Nomor Telepon
081913457896

Nomor Identitas
6781965781999

Pas Foto
Choose File No file chosen
Max. 2048 kB

Kartu Identitas
Choose File No file chosen

Saya setuju dengan persyaratan member Perpustakaan Kota Padang

Daftar

Gambar 5.8 *Input* Registrasi Anggota Perpustakaan Baru (Benar)

Pemerintah Kota Padang
Dinas Perpustakaan dan Kearsipan
Kota Padang

Formulir Pendaftaran Anggota

No Pendaftaran: REG.00000009 No. Anggota:

Pengembalian formulir tanggal:

Nama
Nora Savira

Tempat / Tanggal Lahir
Padang / 1998-05-08

Alamat Tetap
jalan pangambiran

No. Telp / Hp
081913457896

Pendidikan / Pekerjaan
Mahasiswa

Nama Instansi
Unp

Mendaftar sebagai anggota dan menggunakan fasilitas perpustakaan dan saya berjanji akan menaati segala peraturan yang berlaku di Dinas Perpustakaan dan Kearsipan Kota Padang.

Yang Menjanjikan Yang Memohon

(.....) (Nora Savira)

Persyaratan Anggota

- Berdomisili kota Padang
- Mengisi formulir Pendaftaran
- Melampirkan (photocopy) kartu identitas yang masih berlaku
- Kartu keanggotaan hanya berlaku bagi pemilik
- Melampirkan surat keterangan
- Mengganti buku yang hilang ketika dipinjam

Tata Tertib Pengunjung

- Mengisi formulir tamu
- Mendepok barang bawaan pada loket yang telah disediakan dan harap dikunci
- Dilarang membawa paket, tas, topi, dan helm
- Tidak dibenarkan makan, minum, dan merokok
- Tidak dibenarkan memotong, memijat, memotoblo dan merobek koleksi perpustakaan
- Melakukan kembali buku yang telah dibaca sesuai dengan kode klasifikasinya

Print 1 sheet of paper

Destination Microsoft XPS Docum

Pages All

Color Color

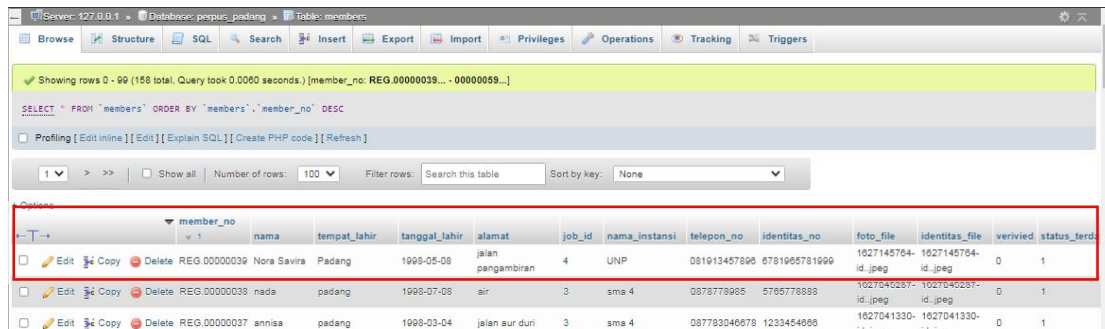
More settings

Print Cancel

Gambar 5.9 *Output* Pengujian Registrasi Anggota Perpustakaan Baru (Benar)

Berdasarkan Gambar 5.8 dan Gambar 5.9 dapat dilihat bahwa registrasi anggota baru berhasil dilakukan. Data tersimpan ke *database* dan sistem menampilkan formulir pendaftaran yang nantinya akan dicetak dan diserahkan kepada pegawai perpustakaan untuk mengambil kartu anggota perpustakaan.

Untuk melakukan pembuktian bahwa registrasi anggota perpustakaan sudah sesuai, maka pengujian dilakukan dengan mengakses langsung ke *database*. Pembuktian dengan *query database* dapat dilihat pada Gambar 5.10

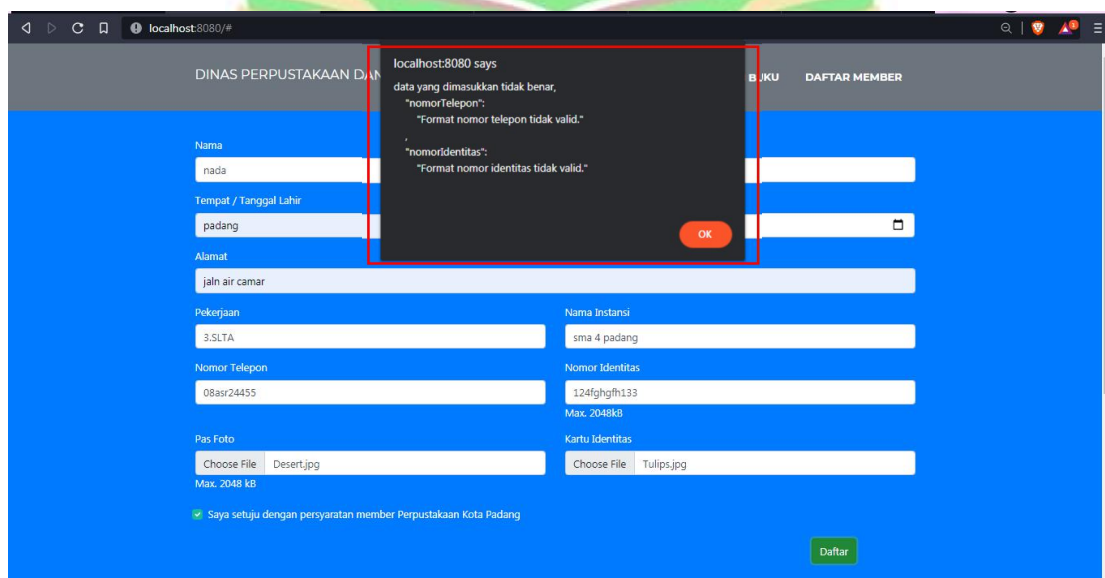


Gambar 5.10 Pengecekan Pengujian Registrasi Anggota Perpustakaan Baru (Benar)

Tabel 5.3 Pengujian Registrasi Anggota Perpustakaan Baru (Alternatif)

Kasus dan Hasil Uji (Alternatif)	
Data masukan	User menekan tombol “Daftar” saat form registrasi “anggota perpustakaan baru” masih ada yang kosong atau salah format data
Yang diharapkan	Sistem menampilkan notifikasi kesalahan
Pengamatan	User tidak bisa melakukan pendaftaran dan sistem menampilkan notifikasi kesalahan
Hasil	Sesuai

Pengujian aplikasi dilakukan dengan menekan tombol “Daftar” sebelum mengisi form atau saat form registrasi anggota masih ada yang belum terisi atau salah format data, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian registrasi Anggota perpustakaan baru (alternatif) dapat dilihat pada Gambar 5.11.



Gambar 5.11 Output Pengujian Registrasi Anggota Perpustakaan Baru (Alternatif)

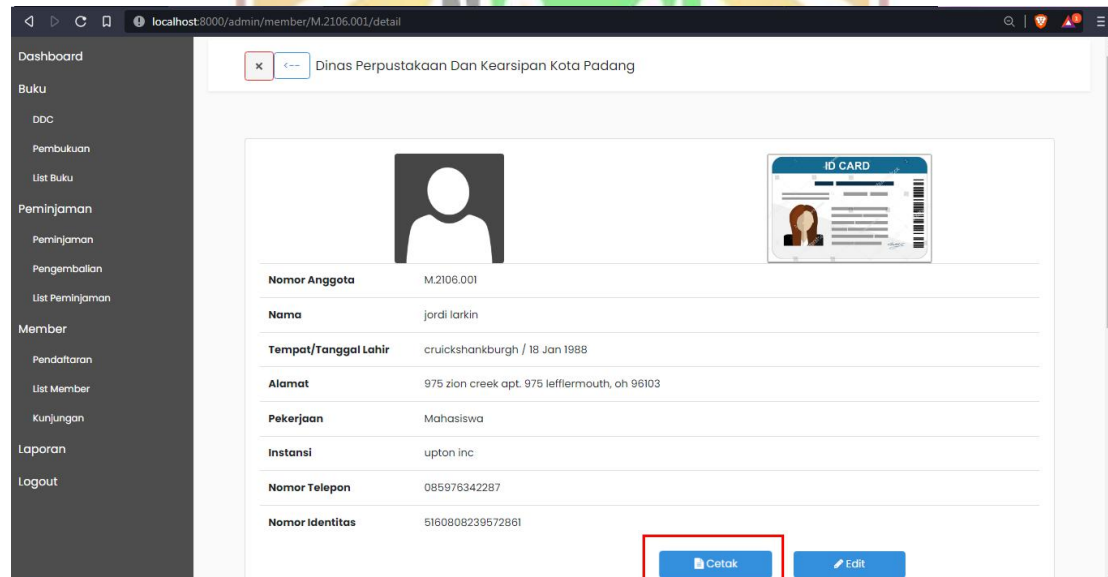
5.2.2.2 Pengujian Cetak Kartu Anggota Perpustakaan

Pada pengujian ini dilakukan cetak kartu anggota perpustakaan pada aplikasi *web*. Hasil pengujian cetak kartu anggota perpustakaan dapat dilihat pada Tabel 5.4.

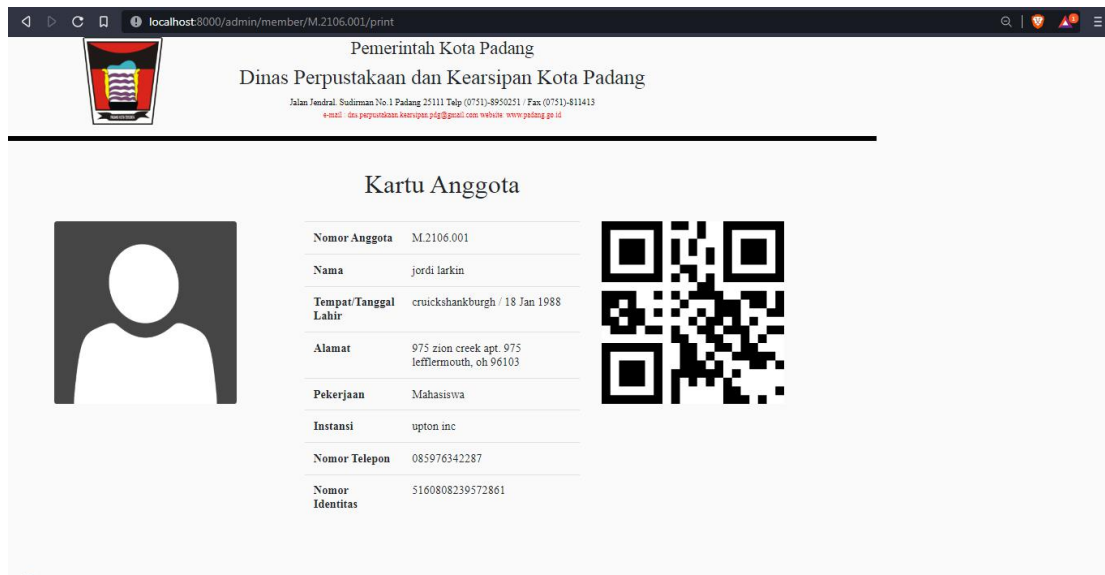
Tabel 5.4 Pengujian Cetak Kartu Anggota Perpustakaan(Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Nomor anggota perpustakaan atau nama anggota perpustakaan
Yang diharapkan	Data didapatkan, muncul detail anggota perpustakaan dan klik tombol cetak pada halaman detail anggota
Pengamatan	Data berhasil didapatkan dan kartu anggota berhasil dicetak
Hasil	Sesuai

Pengujian dilakukan dengan menekan tombol “Detail” pada list anggota perpustakaan, lalu akan muncul halaman detail anggota Perpustakaan. Setelah itu menekan tombol “Cetak”. Jika berhasil, sistem akan menampilkan halaman cetak kartu anggota. Tampilan hasil pengujian cetak kartu anggota perpustakaan (benar) dapat dilihat pada Gambar 5.12 dan Gambar 5.13.



Gambar 5.12 Input Cetak Kartu Anggota Perpustakaan (Benar)



Gambar 5.13 *Output* Pengujian Cetak Kartu Anggota Perpustakaan (Benar)

Berdasarkan Gambar 5.12 dan Gambar 5.13 dapat dilihat bahwa cetak kartu anggota baru berhasil dilakukan dan sistem menampilkan keluaran yang benar.

5.2.2.3 Pengujian Menambah Data Peminjaman Buku

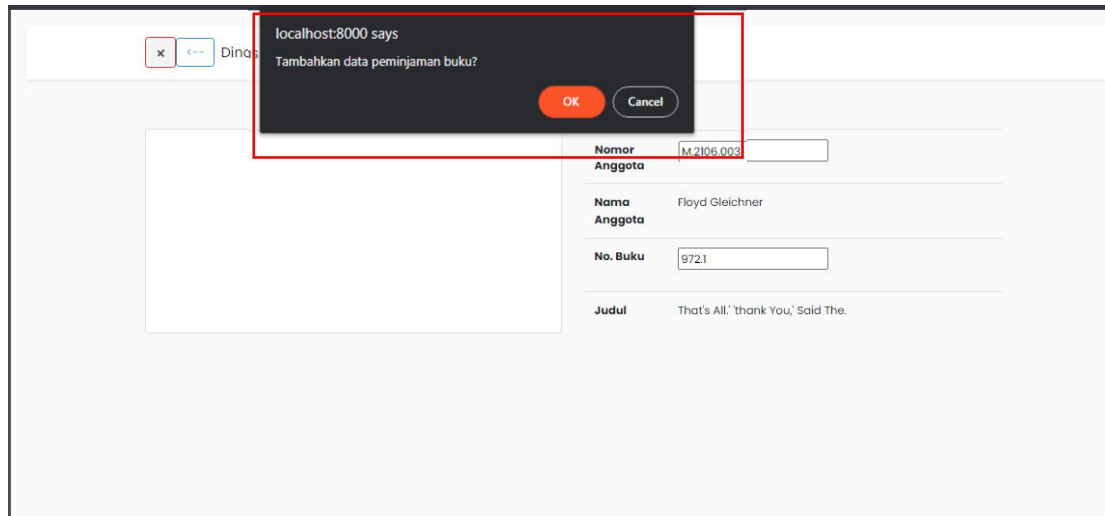
Pada pengujian ini dilakukan dengan scan QR-Code kartu anggota dan kartu buku atau dengan cara input manual nomor anggota dan nomor buku yang dipinjam pada aplikasi *web*. Hasil pengujian menambah data peminjaman buku dapat dilihat pada Tabel 5.5 dan Tabel 5.6.

Tabel 5.5 Pengujian Menambah Data Peminjaman Buku (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Scan QR-Code nomor anggota peminjam dan nomor buku
Yang diharapkan	Data ditemukan dan berhasil melakukan peminjaman, sistem menampilkan halaman detail anggota perpustakaan
Pengamatan	Data peminjaman buku berhasil tersimpan, sistem menampilkan halaman detail anggota perpustakaan dan menampilkan notifikasi “peminjaman buku berhasil ditambahkan”.
Hasil	Sesuai

Pengujian dilakukan dengan menekan menu peminjaman pada halaman tampilan utama, lalu akan muncul halaman scan QR-Code. Setelah itu scan QR-Code kartu anggota dan kartu buku. Jika berhasil, sistem akan menampilkan nomor anggota perpustakaan, nama anggota, nomor buku dan judul buku yang dipinjam. Tampilan

hasil menambah data peminjaman buku(benar) dapat dilihat pada Gambar 5.14 dan 5.15.



localhost:8000 says
Tambahkan data peminjaman buku?

OK Cancel

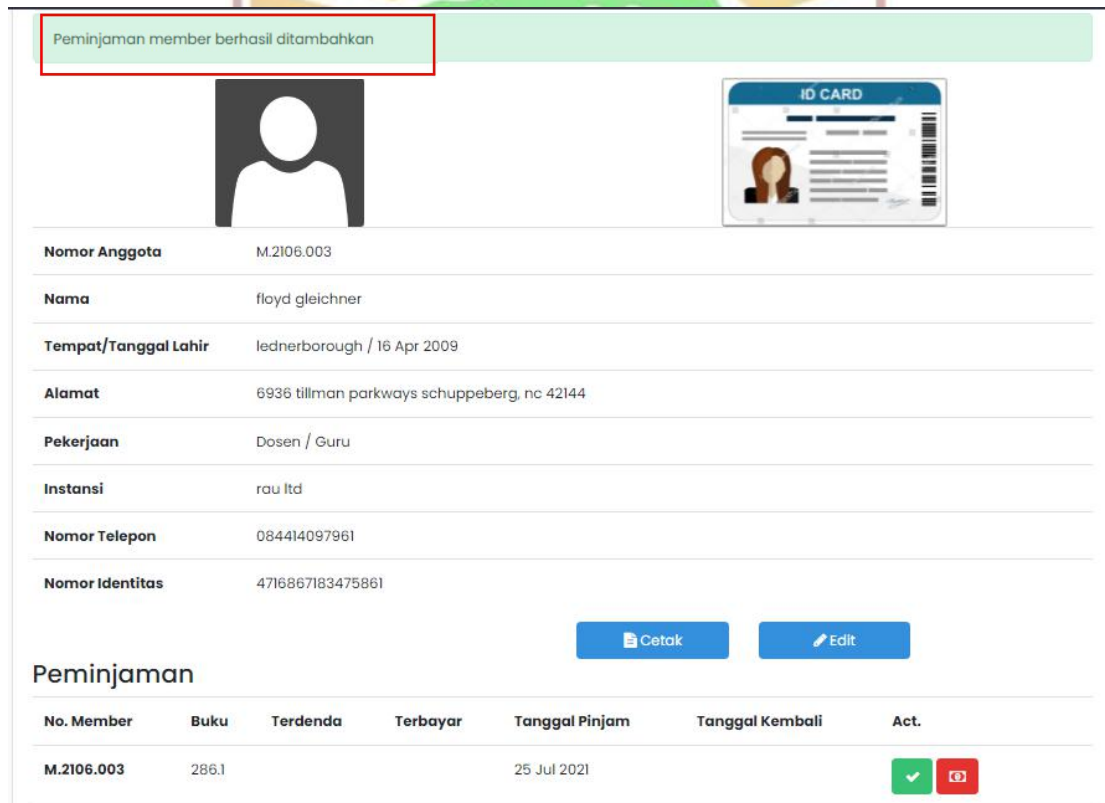
Nomor Anggota M.2106.003

Nama Anggota Floyd Gleichner

No. Buku 9721

Judul That's All, 'thank You,' Said The.

Gambar 5.14 Input Menambah Data Peminjaman Buku (Benar)



Peminjaman member berhasil ditambahkan

ID CARD

Nomor Anggota M.2106.003

Nama floyd gleichner

Tempat/Tanggal Lahir lednerborough / 16 Apr 2009

Alamat 6936 tillman parkways schuppeberg, nc 42144

Pekerjaan Dosen / Guru

Instansi rau ltd

Nomor Telepon 084414097961

Nomor Identitas 4716867183475861

Cetak Edit

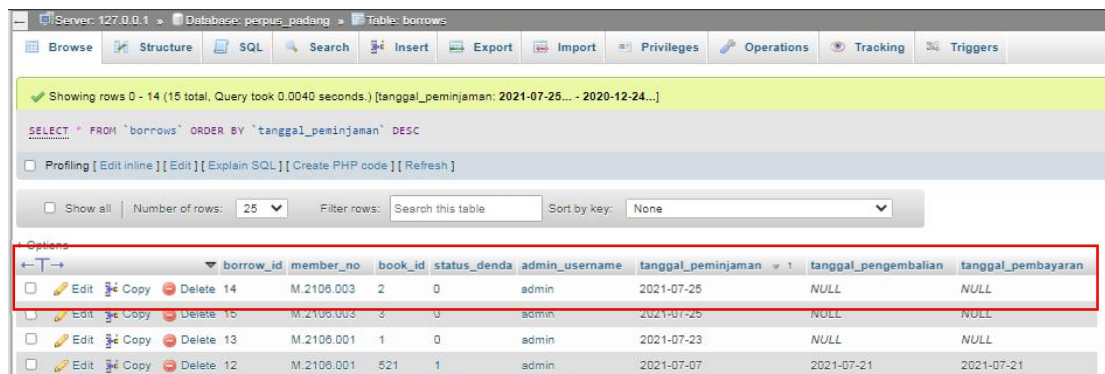
Peminjaman

No. Member	Buku	Terdenda	Terbayar	Tanggal Pinjam	Tanggal Kembali	Act.
M.2106.003	286.1			25 Jul 2021		<input checked="" type="checkbox"/> <input type="checkbox"/>

Gambar 5.15 Output Pengujian Menambah Data Peminjaman Buku(Benar)

Berdasarkan Gambar 5.14 dan Gambar 5.15 dapat dilihat bahwa peminjaman buku berhasil dilakukan. Data tersimpan ke *database* dan sistem menampilkan notifikasi berhasil melakukan peminjaman .

Untuk melakukan pembuktian bahwa menambah data peminjaman buku sudah sesuai, maka pengujian dilakukan dengan mengakses langsung ke *database*. Pembuktian dengan *query database* dapat dilihat pada Gambar 5.16

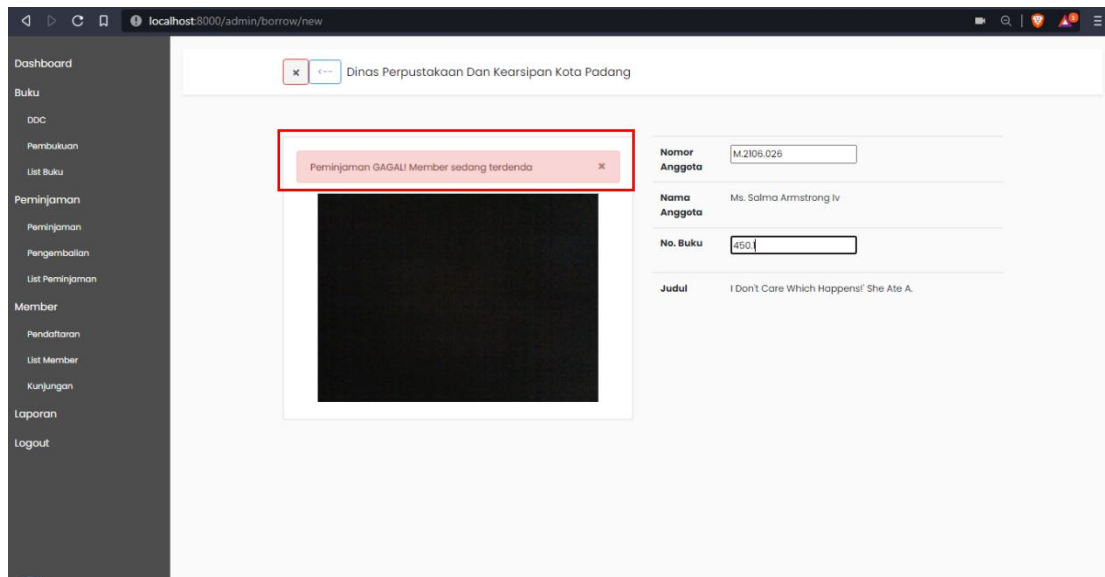


Gambar 5.16 Pengecekan Pengujian Menambah Data Peminjaman Buku (Benar)

Tabel 5.6 Pengujian Menambah Data Peminjaman Buku (Alternatif)

Kasus dan Hasil Uji (Alternatif)	
Data masukan	Scan QR-Code kartu anggota dan kartu buku
Yang diharapkan	Sistem menampilkan notifikasi tidak bisa melakukan peminjaman karena terdenda
Pengamatan	User tidak bisa melakukan peminjaman dan sistem menampilkan notifikasi gagal meminjam
Hasil	Sesuai

Pengujian dilakukan dengan menekan menu peminjaman pada halaman tampilan utama, lalu akan muncul halaman scan QR-Code. Setelah itu scan QR-Code kartu anggota dan kartu buku. Saat melakukan peminjaman muncul notifikasi peminjaman tidak berhasil karena peminjam sedang terdenda. Hasil pengujian menambah data peminjaman buku (alternatif) dapat dilihat pada Gambar 5.17.



Gambar 5.17 Output Pengujian Menambah Data Peminjaman Buku(Alternatif)

5.3 Kesimpulan Hasil Pengujian

Tahap pengujian yang dilakukan terhadap aplikasi yang dibangun, fokus pada ketersediaan dan kesesuaian fungsional sistem yang diuji secara manual. Setelah melakukan pengujian, didapatkan kesesuaian hasil antara perancangan dengan keluaran sistem, selain itu selama pengujian, juga tidak ditemukan kegagalan dalam setiap proses pada masing-masing fungsional. Dapat disimpulkan bahwa pembangunan aplikasi Pengelolaan Pustaka Elektronik Pada Dinas Perpustakaan dan Kearsipan Kota Padang telah berjalan sesuai dengan fungsional yang dirancang. Hasil pengujian secara lengkap dijelaskan pada lampiran I. Kesimpulan dari hasil pengujian sistem dapat dilihat pada Tabel 5.7

Tabel 5.7 Hasil Pengujian

No	Fungsional	Skenario Pengujian		Hasil Yang Diharapkan		Hasil
		Benar	Alternatif	Benar	Alternatif	
1	<i>Login / logout</i>	Pegawai menginputkan <i>username</i> dan <i>password</i>	Pegawai menginputkan <i>username</i> dan <i>password</i> yang salah	Berhasil <i>logout</i> dan <i>login</i> , sistem menampilkan halaman utama aplikasi	Sistem kembali ke halaman <i>login</i> dan menampilkan notifikasi <i>username</i> dan <i>password</i> salah.	Sesuai

Tabel 5.7 Hasil Pengujian(Lanjutan)

2	Mengelola data anggota perpustakaan	Pegawai membuka menu list anggota perpustakaan atau pendaftaran, pegawai bisa menjalankan fungsi verifikasi, mengubah, lihat, cari dan hapus	Pegawai melakukan input yang salah atau <i>primary duplikat</i>	sistem memperbarui data dalam <i>database</i> berdasarkan aksi yang dipilih oleh <i>user</i> .	Sistem menampilkan notifikasi kesalahan	Sesuai
3	Mencetak kartu anggota perpustakaan	Melakukan pencetakan kartu anggota beserta QR-Code yang digunakan untuk <i>login</i> anggota	-	Sistem menampilkan tampilan kartu anggota dan menjalankan fungsi untuk mencetak halaman kartu anggota	-	Sesuai
4	Pegawai melihat dan mencetak laporan kunjungan	Melakukan pencetakan laporan tamu berdasarkan tanggal kunjungan atau pekerjaan	-	Sistem menampilkan tampilan laporan kunjungan dan menjalankan fungsi untuk mencetak halaman	-	Sesuai

Tabel 5.7 Hasil Pengujian(Lanjutan)

				laporan		
5	Mengelola peminjaman dan pengembalian buku	Pegawai membuka menu peminjaman atau pengembalian, admin bisa menjalankan fungsi menambah peminjaman, pengembalian, pemberian denda dan pembayaran denda	Melakukan input data yang salah	Sistem memasukkan atau memperbarui data dalam <i>database</i> berdasarkan aksi yang dipilih oleh <i>user</i>	Sistem menampilkan notifikasi kesalahan	Sesuai
6	Mengelola data pembukuan perpustakaan	Pegawai membuka menu buku/pembukuan serta menjalankan fungsi menambah buku, pembukuan, mengupdate dan melihat	Melakukan input yang salah	Sistem memasukkan atau memperbarui data dalam <i>database</i> berdasarkan aksi yang dipilih oleh <i>user</i>	Sistem menampilkan notifikasi kesalahan	Sesuai
7	Melihat dan mencetak laporan pembukuan	Pegawai melakukan pencetakan laporan buku	-	Sistem menampilkan tampilan laporan buku/	-	Sesuai

Tabel 5.7 Hasil Pengujian(Lanjutan)

	dan list buku	atau pembukuan		pembukuan dan menjalankan fungsi untuk mencetak halaman laporan		
8	Melihat dan mencetak laporan peminjaman dan pengembalian buku	Pegawai melakukan pencetakan laporan peminjaman dan pengembalian buku	-	Sistem menampilkan tampilan laporan peminjaman dan pengembalian dan menjalankan fungsi untuk mencetak halaman laporan	-	Sesuai
9	Peminjaman dan pengembalian buku dengan scan QR-Code	Pegawai perpustakaan membuka menu peminjaman / pengembalian serta menjalankan fungsi menambah peminjaman, pengembalian, pemberian	Melakukan inputan data yang salah	Sistem memasukkan atau memperbaiki data dalam <i>database</i> berdasarkan aksi yang dipilih oleh use	Sistem menampilkan notifikasi kesalahan	Sesuai

Tabel 5.7 Hasil Pengujian(Lanjutan)

		denda, dan pembayaran denda				
10	Mengisi buku tamu dengan QR-Code atau input manual	Pegawai melakukan input atau scan kartu anggota pada halaman <i>check-in</i> anggota perpustakaan	Melakukan input data yang salah	Sistem memasukkan data dalam <i>database</i> berdasarkan aksi yang dipilih oleh <i>user</i>	Sistem menampilkan notifikasi kesalahan	Sesuai
11	Pengunjung bukan anggota mengisi buku tamu dengan cara inputkan data yang diminta	Melakukan input pada halaman <i>check-in</i> tamu bukan anggota	Melakukan input data yang salah	Sistem memasukkan data dalam <i>database</i> berdasarkan aksi yang dipilih oleh <i>user</i>	Sistem menampilkan notifikasi kesalahan	Sesuai
12	Pengunjung melihat daftar pengunjung perpustakaan	Pengunjung melihat halaman <i>check-in</i> dan langsung melihat daftar pengunjung hari ini dan jam saat ini	-	Sistem menampilkan daftar pengunjung perpustakaan hari ini dan jam saat ini	-	Sesuai
13	Pengunjung melihat grafik buku yang sering	Pengunjung melihat halaman <i>check-in</i> dan	-	Sistem menampilkan grafik buku yang sering		

Tabel 5.7 Hasil Pengujian(Lanjutan)

	dipinjam dan pengunjung bulanan	langsung melihat grafik buku yang sering dipinjam dan grafik pengunjung bulanan		dipinjam dan grafik pengunjung bulanan		
14	Pengunjung melakukan pendaftaran menjadi anggota perpustakaan	Pengunjung membuka halaman pendaftaran, mengisi form pendaftaran dan menekan tombol daftar	Pengunjung membuka halaman pendaftaran, mengisi form pendaftaran dan menekan tombol pendaftaran saat inputan data salah dan kosong	sistem memasukkan data dalam <i>database</i> dan sistem menampilkan halaman formulir pendaftaran	Sistem menampilkan notifikasi kesalahan	Sesuai
15	Pengunjung mencari buku pada list buku	Melakukan akses halaman buku dan memasukkan kata kunci dari judul atau penulis buku yang dicari		Menampilkan hasil pencarian data berdasarkan kata kunci	-	Sesuai