

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 pembuatan akta autentik notaris yang diusulkan pada notaris dan PPAT Rismadona, SH. Pada implementasi ini dijelaskan tentang hal-hal yang berhubungan dengan alur perangkat lunak dan spesifikasi yang dibutuhkan pada saat penelitian, lalu pada pengujian menjelaskan tentang fungsional yang ada pada aplikasi apakah berjalan sesuai perancangan atau tidak. Tahap implementasi dan tahap pengujian sistem dilakukan pada aplikasi *web*.

5.1 Implementasi Sistem

Aplikasi pembuatan akta autentik notaris ini difungsikan sebagai aplikasi yang mengelola dan menyajikan data yang berkaitan dengan akta pendirian perusahaan. Pada sistem ini terdapat tiga 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 sistem.

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

1. Komputer dengan *processor* AMD A8-6410 APU CPU 2.0 GHz.
2. Memori Komputer (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 8.1 Pro
2. *Web browser* yang digunakan adalah google chrome versi 74.0.

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

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 3.2.2. *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

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 mengendalikan semua perintah yang dideklarasikan dan mengirimkannya kepada *controller* serta *method* yang sesuai. Kode program *routing* aplikasi dapat dilihat pada Gambar 5.1

```
<?php
Auth::routes();
Route::middleware('auth')->group(function () {
    Route::get('/home', 'HomeController@index')->name('home');
    Route::get('/', 'HomeController@index');
    Route::resource('/users', 'UserController');
    Route::get('user/profile', 'UserController@profile')->
>name('user.profile');
    Route::get('user/edit/profile', 'UserController@editProfile')->
>name('user.editProfile');
    Route::patch('user/store/profile', 'UserController@storeProfile')->
>name('user.storeProfile');
    Route::resource('/templates', 'TemplateController');
    Route::resource('/akta', 'AktaController');
    Route::get('/akta/revisi/{id}', 'AktaController@revisi')->
>name('akta.revisi');
    Route::post('storeDisposisi', 'AktaController@storeDisposisi')->
>name('akta.storeDisposisi');
    Route::get('/akta/revisi/{id}', 'AktaController@revisi')->
>name('akta.revisi');
    Route::get('/akta/kirim/{id}', 'AktaController@kirim')->
>name('akta.kirim');
    Route::get('/akta/proses/{id}', 'AktaController@proses')->
>name('akta.proses');
```

Gambar 5. 1. Kode Program *Routing* Aplikasi

```

Route::get('/akta/close/{id}', 'AktaController@close')-
>name('akta.close');
Route::get('get/word/{id}', 'AktaGeneratorController@word')-
>name('getWord');
    Route::resource('/pihakTerlibat', 'PihakTerlibatController');

Route::get('/pihakTerlibat/create/{id}', 'PihakTerlibatController@create')-
->name('pihakTerlibat.create');
    Route::resource('/aktaUpload', 'AktaUploadController');
    Route::get('get/upload', 'AktaUploadController@getUpload')-
>name('getUpload');
    Route::resource('saksi', 'SaksiController');
    Route::resource('detailSaksi', 'DetailSaksiController');

Route::get('detailSaksi/create/{id}', 'DetailSaksiController@create')-
>name('detailSaksi.create');

Route::post('/getNotification', 'NotificationController@getNotification')-
->name('getNotification');
    Route::get('/getFile/{type}/{id}', 'DownloadController@getFile')-
>name('getFile');
    });

```

Gambar 5.1. Kode Program *Routing* Aplikasi(lanjutan)

5.1.1.2 Kode Program *Controller* Detail Saksi

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

```

<?php

namespace App\Http\Controllers;

use App\Models\DetailSaksi;
use App\Models\PihakTerlibat;
use App\Models\Saksi;
use Illuminate\Http\Request;

class DetailSaksiController extends Controller
{
    public function create($ids){
        $saksi = Saksi::pluck('nama', 'id');
        return view('detail_saksi.create', compact('saksi', 'ids'));
    }

    public function store(Request $request){
        DetailSaksi::create($request->all());
        return redirect(route('akta.show', ['id'=>$request->akta_id]));
    }

    public function edit($id){
        $saksi = Saksi::pluck('nama', 'id');
        $detailSaksi = DetailSaksi::find($id);
    }
}

```

Gambar 5.2. Kode Program *AktaController*

```

        return view('detail_saksi.edit', compact('saksi', 'detailSaksi'));
    }

    public function update(Request $request, $id) {
        $detail = DetailSaksi::find($id);
        $detail->update($request->all());
        return redirect(route('akta.show', ['id' => $detail->akta_id]));
    }

    public function destroy($id) {
        $pihak = DetailSaksi::find($id);
        $akta_id = $pihak->akta_id;
        $pihak->delete();
        return redirect(route('akta.show', ['id' => $akta_id]));
    }
}

```

Gambar 5.2. Kode Program *DetailSaksiController*(lanjutan)

5.1.1.3 Kode Program Model Akta

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* Akta untuk memanggil entitas akta pada *database*. Kode program *model* Akta dapat dilihat pada Gambar 5.3.

```

<?php
namespace App\Models;
use Eloquent as Model;
class Akta extends Model
{
    public $table = 'akta';

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

    public $fillable = [
        'nama_instansi',
        'alamat',
        'tanggal',
        'jam',
        'nomor_akta',
        'id_template',
        'id_notaris',
        'id_pemohon',
        'status',
        'modal',
        'helai',
        'nominal',
        'foto_rek_koran'
    ];
}

```

Gambar 5. 3. Kode Program Model Akta

```

protected $casts = [
    'id' => 'integer',
    'nama_instansi' => 'string',
    'alamat' => 'string',
    'nomor_akta' => 'integer',
    'id_template' => 'integer',
    'id_notaris' => 'integer',
    'id_pemohon' => 'integer',
    'status' => 'integer',
    'modal' => 'string',
    'helai' => 'string',
    'nominal' => 'string',
    'tanggal' => 'date',
    'jam' => 'time',
    'foto_rek_koran' => 'string',
];
public static $rules = [
];
public function user()
{
    return $this-
>hasOne(\App\Models\User::class,'id','id_pemohon');
}
public function notaris()
{
    return $this-
>hasOne(\App\Models\User::class,'id','id_notaris');
}

public function template()
{
    return $this-
>hasOne(\App\Models\Template::class,'id','id_template');
}
public function pihakTerlibats()
{
    return $this->hasMany(\App\Models\PihakTerlibat::class)-
>orderBy('id','ASC');
}

public function detailSaksis()
{
    return $this->hasMany(\App\Models\DetailSaksi::class)-
>orderBy('id','ASC');
}
public function logs()
{
    return $this->hasMany(\App\Models\Log::class)-
>orderBy('id','DESC');
}
}

```

Gambar 5. 2. Kode Program Model Akta(lanjutan)

5.1.1.4 Kode Program View Akta

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* akta *index.blade.php* yang

berfungsi untuk menampilkan halaman “akta”. Kode program *view* akta dapat dilihat pada Gambar 5.4.

```
@extends('layouts.app')

@section('content')
    <section class="content-header">
        <h1 class="pull-left">Akta</h1>
        <h1 class="pull-right">
            <a class="btn btn-primary pull-right" style="margin-top:
-10px;margin-bottom: 5px" href="{!! route('akta.create') !!}">+
Tambah</a>
        </h1>
    </section>
    <div class="content">
        <div class="clearfix"></div>
        <div class="box box-primary">
            <div class="box-body">
                @include('akta.table')
            </div>
        </div>
        <div class="text-center">

    </div>
    </div>
</section>
@section('css')
    <link rel="stylesheet" type="text/css" href="{!!
asset('js/DataTables/datatables.min.css') !!}" />
</section>

@section('scripts')
    <script type="text/javascript" src="{!!
asset('js/DataTables/datatables.min.js') !!}"></script>
    <script>
        $(document).ready(function () {
            var table = $('table').dataTable({
                responsive: true
            });
        });
    </script>
</section>
```

Gambar 5.4. Kode Program View Akta

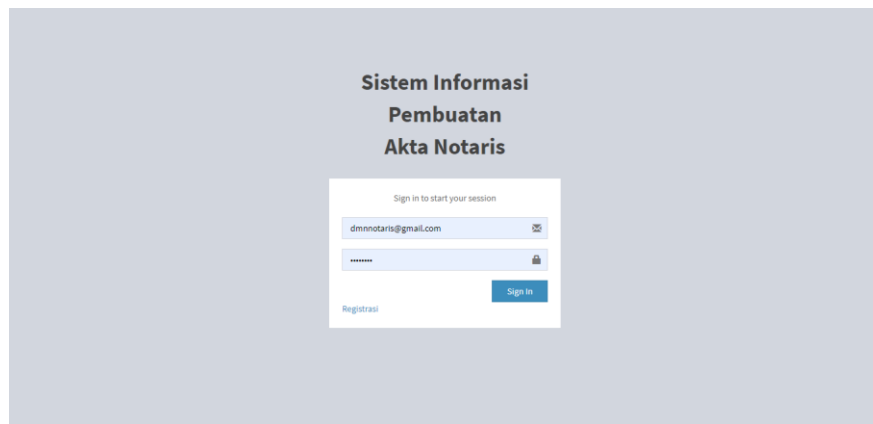
5.1.2 Implementasi Antarmuka Aplikasi Web

Aplikasi *web* yang dibangun dapat diakses oleh *user* menggunakan *web browser*. Berikut dijelaskan tampilan aplikasi *web* pembuatan akta autentik notaris 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 untuk menjaga agar aplikasi ini hanya dapat digunakan oleh *user* yang terlibat dalam sistem pembuatan

akta autentik notaris. Berikut tampilan halaman *login* dapat dilihat pada Gambar 5.5.

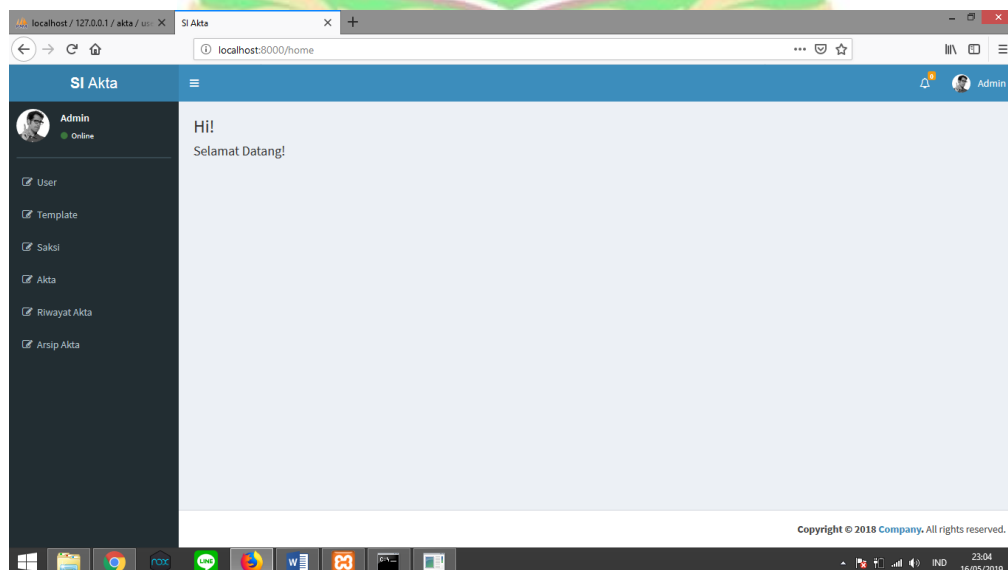


Gambar 5. 5. Tampilan Halaman Login

Berdasarkan Gambar 5.5 *user* terlebih dahulu harus membuat akun sebelum bisa melakukan *login*. Setelah membuat akun *user* dapat masuk kedalam aplikasi dengan men-*input*-kan *username* dan *password* yang telah di *input*-kan sebelumnya lalu menekan tombol “Sign In”. 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 admin. Admin memiliki wewenang dalam mengelola data akun, akta, template, saksi dan arsip akta. 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 lima menu utama, berikut keterangan dari menu-menu tersebut:

1. *User*

Menu *user* berfungsi sebagai menu untuk mengelola semua akun yang ada di dalam sistem.

2. *Template*

Menu *template* berfungsi sebagai menu untuk mengelola *template* akta autentik notaris.

3. Saksi

Menu saksi berfungsi untuk mengelola data saksi.

4. Akta

Menu akta berfungsi untuk mengelola akta dan pengajuan permohonan akta .

5. Riwayat Akta

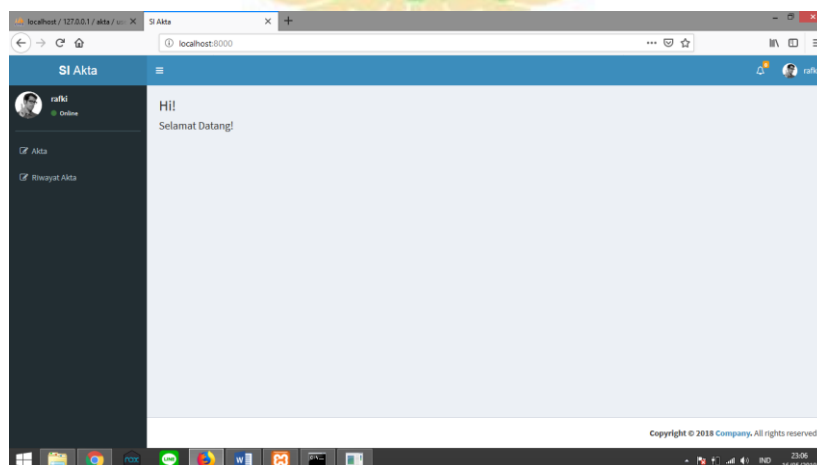
Menu riwayat akta berfungsi untuk melihat history akta

6. Arsip Akta

Menu arsip akta berfungsi untuk mengelola penyimpanan akta autentik notaris.

5.1.2.3 Halaman Pemohon

Halaman pemohon diberikan kepada seorang *user* dengan level hak akses pemohon. Pemohon hanya dapat memilih menu akta. Tampilan halaman pemohon dapat dilihat pada Gambar 5.7.



Gambar 5. 7. Tampilan Halaman Pemohon

Berdasarkan Gambar 5.7 dapat dijelaskan bahwa halaman pemohon hanya memiliki satu menu utama, berikut keterangan dari menu tersebut:

1. Akta

Menu akta berfungsi sebagai menu untuk pengajuan pembuatan akta autentik notaris.

2. Riwayat Akta

Menu riwayat akta berfungsi untuk melihat history akta

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 pegawai notaris Rismadona, SH.

5.2.1 Fokus Pengujian

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

Tabel 5.1. Fokus Pengujian

No.	Item Uji	Detail Pengujian
1	Pengajuan pembuatan akta	Lihat, <i>create</i> , <i>edit</i>
2	Mengelola Akta	Lihat, <i>create</i> , <i>edit</i> , <i>delete</i>
3	Mengelola akun	Lihat, <i>create</i> , <i>edit</i> , <i>delete</i>
4	Memperbaiki status akta	<i>create</i>

Tabel 5.2. Fokus Pengujian(lanjutan)

5	Mengupload akta	<i>Upload</i>
6	Mendownload akta	<i>Download</i>
7	Mengecek permohonan pengajuan akta	Lihat
8	Mengirim notifikasi	Kirim data
9	<i>History</i> pembuatan akta	Lihat
10.	Mengelola <i>template</i>	Lihat, <i>create, edit, delete</i>
11	Mengecek status akta	Lihat
12	Mengelola saksi	Lihat, <i>create, edit, delete</i>

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 pengajuan pembuatan akta, menambah *template* akta dan mengecek status akta.

5.2.2.1 Pengujian Pengajuan Pembuatan Akta

Pada pengujian ini dilakukan aktifitas pengujian pembuatan akta. Hasil pengujian pengajuan pembuatan akta dapat dilihat pada Tabel 5.2 dan Tabel 5.3

Tabel 5.1. Pengajuan Pembuatan akta (Benar)

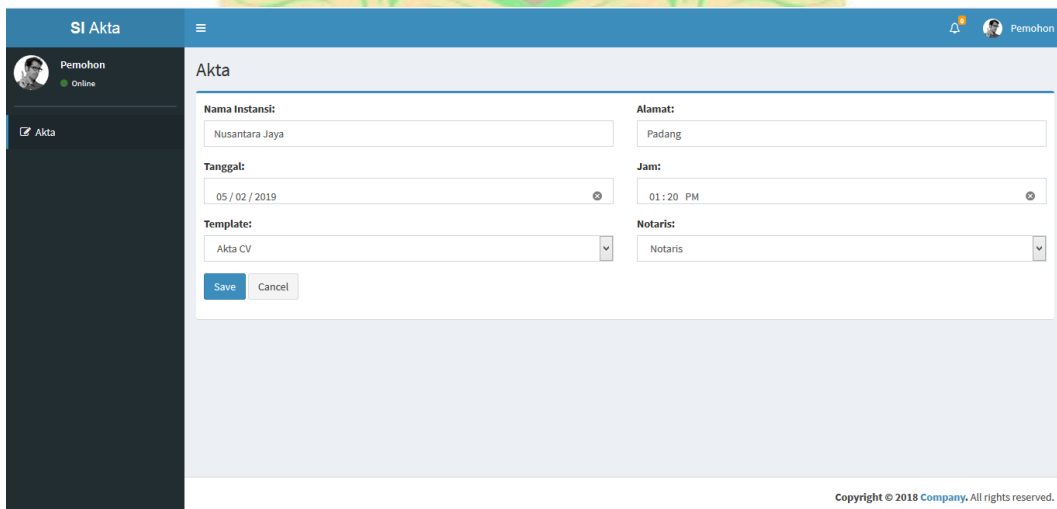
Kasus dan Hasil Uji (Benar)	
Data masukan	Seluruh data yang dibutuhkan dalam halaman akta
Yang diharapkan	Sistem menampilkan data akta yang telah diinputkan
Pengamatan	Dapat menginput data pada halaman akta serta menekan tombol “save”
Hasil	Sesuai

Tabel 5.2. Pengajuan Pembuatan Akta (Alternatif)

Kasus dan Hasil Uji (Alternatif)	
Data masukan	User menekan tombol “simpan” sebelum mengisi form
Yang diharapkan	Sistem menampilkan notifikasi kesalahan
Pengamatan	User tidak bisa mengajukan pembuatan akta
Hasil	Sesuai

A. Pengajuan Pembuatan Akta (Benar)

Pengujian aplikasi dilakukan dengan menginput data untuk pembuatan akta pada halaman akta dan menekan tombol “save”. Jika berhasil maka sistem menampilkan halaman lihat data akta. Hasil pengujian pengajuan pembuatan akta (benar) dapat dilihat pada Gambar 5.8 dan Gambar 5.9.

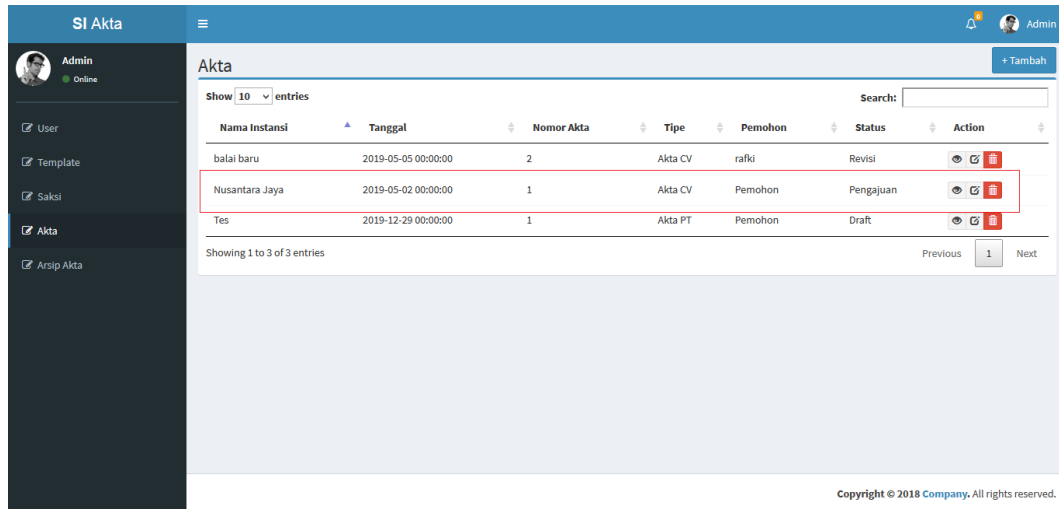


The screenshot shows a web interface for 'SI Akta'. On the left, there is a sidebar with a user profile 'Pemohon' (Online) and a menu item 'Akta'. The main content area is titled 'Akta' and contains a form with the following fields:

- Nama Instansi:** Nusantara Jaya
- Alamat:** Padang
- Tanggal:** 05 / 02 / 2019
- Jam:** 01 : 20 PM
- Template:** Akta CV
- Notaris:** Notaris

At the bottom of the form, there are two buttons: 'Save' and 'Cancel'. The footer of the page contains the text: 'Copyright © 2018 Company. All rights reserved.'

Gambar 5. 8. Pengajuan Pembuatan Akta (Benar)

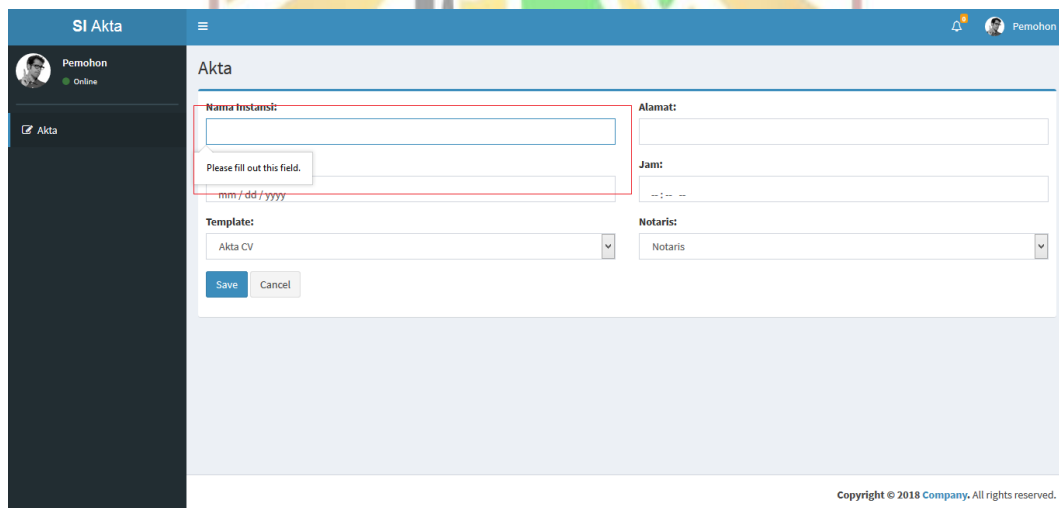


Gambar 5. 9. Pengajuan Pembuatan Akta (Benar)

Berdasarkan gambar 5.8 dan gambar 5.9 dapat dilihat bahwa pengajuan pembuatan akta berhasil dilakukan.

B. Pengajuan Pembutan Akta (Alternatif)

Pengujian aplikasi dilakukan dengan menekan tombol “save” sebelum mengisi *form*, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian pengajuan pembuatan akta (alternatif) dapat dilihat pada Gambar 5.10.



Gambar 5. 10. Pengajuan Pembuatan Akta (Alternatif)

Berdasarkan Gambar 5.10 terlihat bahwa sistem menampilkan notifikasi kesalahan ketika tidak ada data akta yang ditambahkan.

5.2.2.2 Mengelola *Template* Akta(tambah)

Pada pengujian ini dilakukan menambah *template* akta. Hasil pengujian tambah *template* dapat dilihat pada Tabel 5.4 dan Tabel 5.5.

Tabel 5.3. Menambah *Template* (Benar)

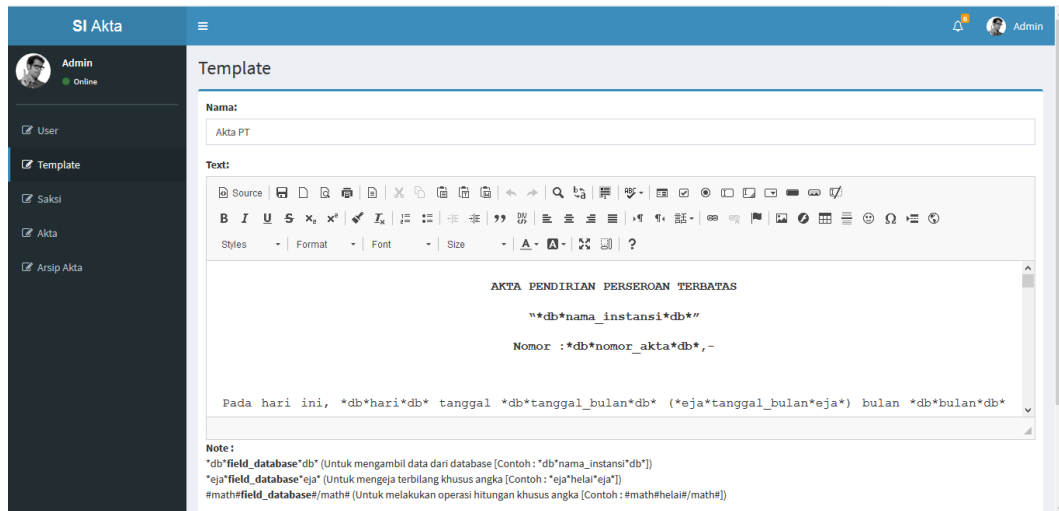
Kasus dan Hasil Uji (Benar)	
Data masukan	Seluruh data yang dibutuhkan dalam halaman <i>template</i>
Yang diharapkan	Sistem menampilkan halaman <i>template</i>
Pengamatan	Menginput data pada halaman <i>template</i> serta menekan tombol “ <i>save</i> ”
Hasil	Sesuai

Tabel 5.4. Menambah *Template* (Alternatif)

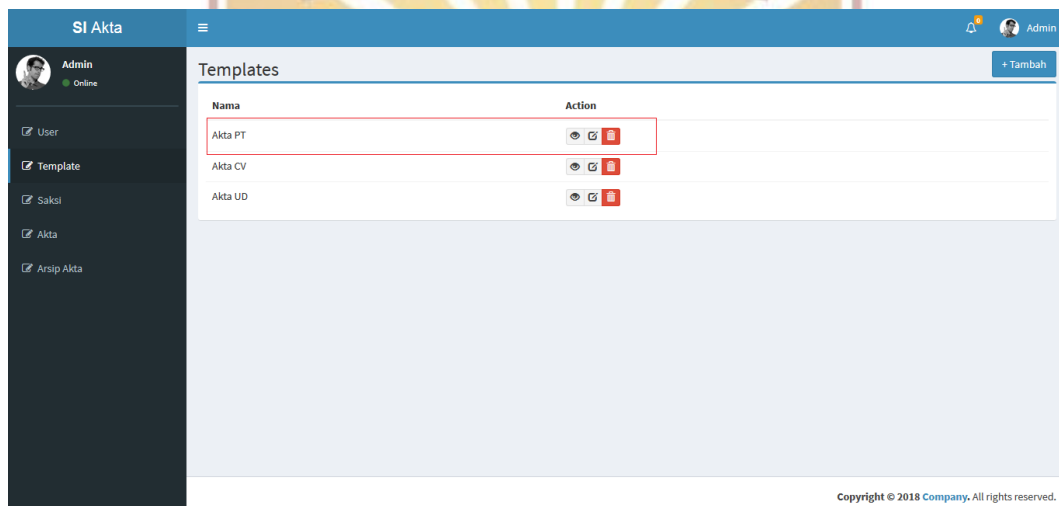
Kasus dan Hasil Uji (Alternatif)	
Data masukan	<i>User</i> menekan tombol “ <i>save</i> ” sebelum mengisi <i>form</i>
Yang diharapkan	Sistem menampilkan notifikasi kesalahan
Pengamatan	<i>User</i> tidak bisa menambahkan <i>template</i>
Hasil	Sesuai

A. Menambah *Template* (Benar)

Pengujian aplikasi dilakukan dengan nama *template* dan *template* pada halaman *template* dan menekan tombol “*save*”. Jika berhasil maka sistem menampilkan halaman *template*. Hasil pengujian menambah *template* (benar) dapat dilihat pada Gambar 5.11 dan Gambar 5.12.



Gambar 5. 11. Menambah Template (Benar)

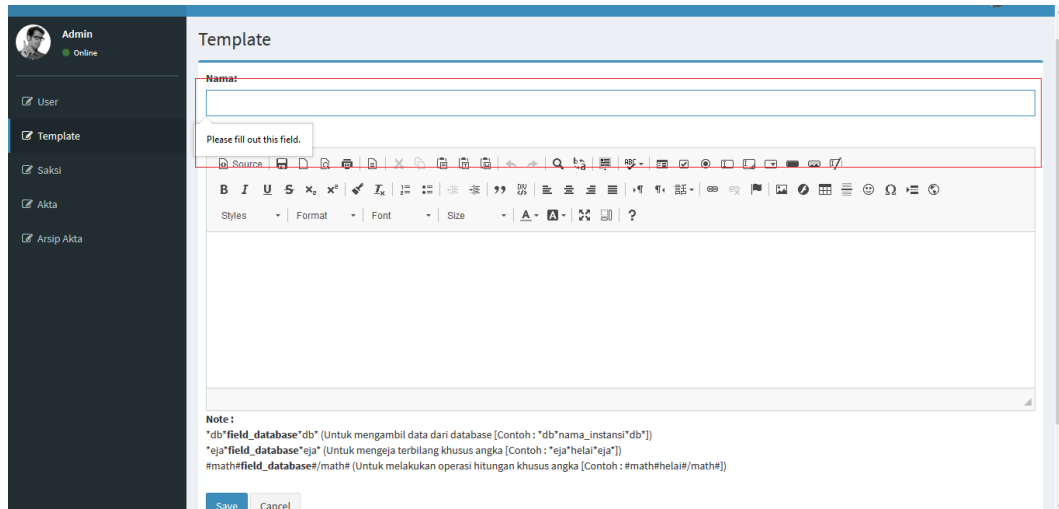


Gambar 5. 12. Menambah Template (Benar)

Berdasarkan Gambar 5.11 dan Gambar 5.12 terlihat bahwa pengujian menambah *template* berhasil dilakukan.

B. Menambah *Template* (Alternatif)

Pengujian aplikasi dilakukan dengan menekan tombol “*save*” sebelum mengisi *form*, kemudian sistem menampilkan notifikasi kesalahan. Hasil pengujian menambah *template* (alternatif) dapat dilihat pada Gambar 5.13.



Gambar 5.13. Menambah Template (Alternatif)

5.2.2.3 Pengecekan Status Akta

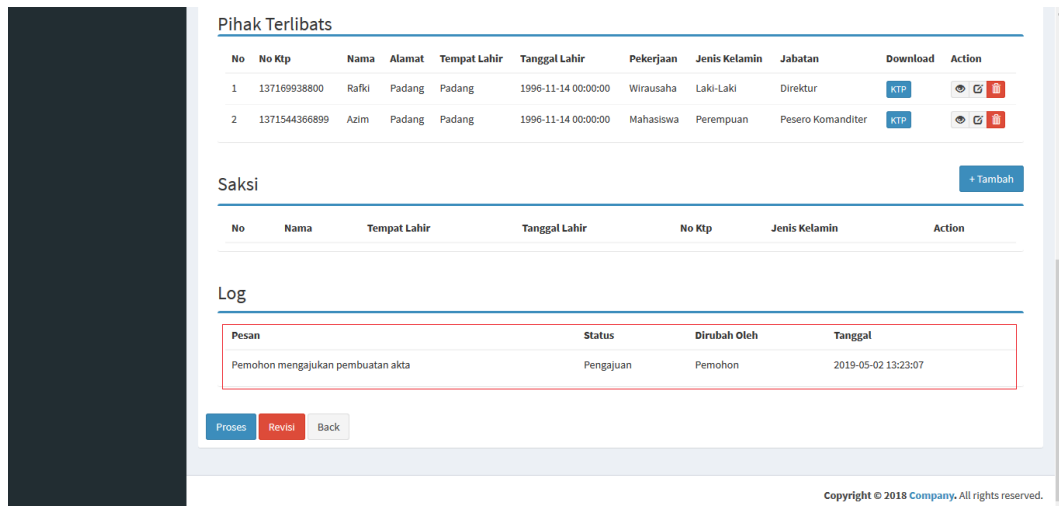
Pada pengujian ini dilakukan aktifitas pengecekan status akta. Hasil pengecekan status akta dapat dilihat pada Tabel 5.6.

Tabel 5.6. Pengujian Pengecekan Status Akta (Benar)

Kasus dan Hasil Uji (Benar)	
Data masukan	Seluruh data yang dibutuhkan dalam halaman lihat data akta
Yang diharapkan	Sistem menampilkan halaman lihat data akta
Pengamatan	Dapat melihat status akta yang telah di ajukan
Hasil	Sesuai

A. Pengujian Pengecekan Status Akta (Benar)

Pengujian aplikasi dilakukan melihat bagian Log halaman lihat data akta. Hasil pengujian pengecekan status akta (benar) dapat dilihat pada Gambar 5.14.



Gambar 5. 14. Pengecekan Status Akta (Benar)

Berdasarkan Gambar 5.14 terlihat bahwa pengujian pengecekan status akta berhasil dilakukan.

5.3 Kesimpulan Hasil Pengujian

Pengujian yang dilakukan terhadap aplikasi yang dibangun berfokus pada kesesuaian fungsional sistem yang diuji secara manual. Berdasarkan pengujian yang telah dilakukan diperoleh sebuah hasil antara perancangan dengan keluaran sistem, selain itu juga tidak ditemukan kegagalan dalam setiap proses pada masing-masing fungsional. Dapat disimpulkan bahwa aplikasi web untuk sistem informasi pembuatan akta autentik notaris pada notaris dan PPAT Rismadona, SH 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	Mengelola data <i>user</i>	Admin membuka menu data <i>user</i> , admin bisa menjalankan fungsi tambah, edit, hapus dan cari.	Admin membuka menu data <i>user</i> , admin tidak bisa menambah data jika formnya belum diisi.	Sistem menampilkan seluruh data <i>user</i> , menyimpan dan berhasil menjalankan fungsional.	Sistem menampilkan notifikasi kesalahan pada saat tambah data <i>user</i> .	Sesuai

Tabel 5.7. Hasil Pengujian(lanjutan)

2	Mengelola akta	Admin dan notaris membuka menu akta, admin dan notaris bisa menjalankan fungsi tambah, edit, hapus dan cari.	Admin dan notaris membuka menu akta, admin dan notaris tidak bisa menambah data jika formnya belum diisi.	Sistem menampilkan data akta yang baru ditambahkan.	Sistem menampilkan notifikasi kesalahan pada saat <i>input</i> data akta.	Sesuai
3	Mengelola saksi	Admin dan notaris membuka menu saksi, admin dan notaris bisa menjalankan fungsi tambah, edit, <i>detail</i> dan hapus	Admin dan notaris membuka menu saksi, admin dan notaris tidak bisa menambah data jika formnya belum diisi.	Sistem menampilkan data saksi yang baru ditambahkan.	Sistem menampilkan notifikasi kesalahan pada saat <i>input</i> data saksi.	Sesuai
4	Mengelola <i>template</i> akta	Admin dan notaris membuka menu <i>template</i> akta, admin bisa menjalankan fungsi tambah, edit, <i>detail</i> dan hapus	Admin dan notaris membuka menu <i>template</i> , admin dan notaris tidak bisa menambah data jika formnya belum diisi.	Sistem menampilkan data <i>template</i> yang baru ditambahkan.	Sistem menampilkan notifikasi kesalahan pada saat <i>input</i> <i>template</i> .	Sesuai
5	Perbarui status akta	Admin dan notaris membuka menu akta dan memillih lihat akta, admin dan notaris bisa memperbarui status akta setelah di periksa datanya	Admin dan notaris membuka menu lihat akta, admin dan notaris tidak bisa memperbarui status akta.	Sistem menampilkan status akta yang telah di perbarui.	Sistem menampilkan notifikasi kesalahan pada saat perbarui status akta.	Sesuai

Tabel 5.7. Hasil Pengujian(lanjutan)

6	Mengupload akta	Admin dan notaris membuka menu arsip akta, admin dan notaris bisa menjalankan fungsi tambah, edit, <i>detail</i> , hapus dan cari	Admin dan notaris membuka menu arsip akta, admin dan notaris tidak bisa menambahkan data jika formnya belum diisi.	Sistem menampilkan data arsip akta yang baru ditambahkan.	Sistem menampilkan notifikasi kesalahan pada saat <i>input</i> data arsip akta.	Sesuai
7	Download akta	Admin dan notaris membuka menu akta dan lihat akta, admin dan notaris bisa menjalankan fungsi mengunduh akta.	Admin dan notaris membuka menu lihat akta, admin dan notaris tidak bisa mengunduh akta.	Sistem menampilkan akta yang telah terunduh.	Sistem menampilkan notifikasi kesalahan pada saat mengunduh akta.	Sesuai
8	Mengecek permohonan pembuatan akta	Admin dan notaris membuka menu akta dan melihat <i>input-an</i> akta baru.	Admin dan notaris membuka menu akta dan tidak ada <i>input-an</i> akta baru.	Sistem menampilkan data akta yang baru ditambahkan.	Sistem tidak menampilkan data akta yang baru ditambahkan.	Sesuai
9	Mengirim notifikasi	Admin dan notaris membuka menu akta dan memilih lihat akta selanjutnta memeriksa data akta dan mengirim notifikasi sesuai kesalahan data	Admin dan notaris membuka menu lihat akta, admin dan notaris tidak bisa mengirim notifikasi	Sistem menampilkan pesan notifikasi yang baru di tambahkan.	Sistem tidak menampilkan notifikasi yang baru ditambahkan	Sesuai

Tabel 5.7. Hasil Pengujian(lanjutan)

10	Melihat history pembuatan akta	Admin dan notaris membuka menu riwayat akta dan memillih lihat akta, admin dan notaris bisa melihat histroy pembuatan akta.	Admin dan notaris membuka menu lihat akta, admin dan notaris tidak bisa melihat history pembuatan akta.	Sistem menampilkan data history pembuatan akta.	Sistem tidak menampilkan data history pembuatan akta.	Sesuai
11	Mengajukan pembuatan akta	Pemohon membuka menu akta, pemohon bisa menjalankan fungsi tambah, edit, <i>detail</i> , dan cari.	Pemohon membuka menu akta, pemohon tidak bisa menambahkan data jika formnya belum diisi.	Sistem menampilkan data akta yang baru ditambahkan.	Sistem menampilkan notifikasi kesalahan pada saat <i>input</i> data akta.	Sesuai
12	Mengecek status akta	Admin, notaris dan pemohon membuka menu akta, admin, notaris dan pemohon bisa melihat status akta.	Admin, notaris dan pemohon membuka menu akta, admin, notaris dan pemohon tidak bisa melihat status akta.	Sistem menampilkan status akta.	Sistem tidak menampilkan status akta.	Sesuai