

## BAB IV

### IMPLEMENTASI DAN PEMBAHASAN SISTEM

#### 4.1 Implementasi Sistem

Tujuan dari tahap implementasi adalah menyiapkan semua kegiatan penerapan sistem sesuai dengan rancangan yang telah ditentukan. Setelah itu akan dilakukan tahapan pengujian system guna meminimalisir segala kemungkinan kesalahan yang terjadi.

##### 4.1.1 Implementasi Basis Data

Pembuatan basis data dan tabel adalah langkah awal dalam mengimplementasikan sebuah sistem. Berikut adalah basis data yang digunakan untuk aplikasi nomor register legalisasi dokumen adminduk.:

##### 1. Tabel User

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	<u>userId</u>	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Indeks Lainnya
2	<u>email</u>	varchar(128)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Lainnya
3	<u>password</u>	varchar(128)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Lainnya
4	<u>name</u>	varchar(128)			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Lainnya
5	<u>mobile</u>	varchar(20)			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Lainnya
6	<u>statusId</u>	tinyint(4)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Lainnya
7	<u>isDeleted</u>	tinyint(4)			Tidak	0		Ubah Hapus Kunci Utama Unik Indeks Lainnya
8	<u>createdBy</u>	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Lainnya
9	<u>createdDtm</u>	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Lainnya
10	<u>updatedBy</u>	int(11)			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Lainnya
11	<u>updatedDtm</u>	datetime			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Lainnya

Gambar 4.1. Implementasi Tabel User

Gambar 4.1 merupakan hasil implementasi tabel user yang digunakan untuk menyimpan data user.

## 2. Tabel Biodata Adminduk

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	biodatmid	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
2	NIK	varchar(16)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
3	NAMA_LGKP	varchar(50)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
4	JENIS_KLMIN	varchar(10)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
5	TMPT_LHR	varchar(100)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
6	TGL_LHR	date			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
7	NO_AKTA_LHR	varchar(16)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
8	GOL_DRH	varchar(10)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
9	AGAMA	varchar(20)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
10	STAT_KWN	varchar(25)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
11	NO_AKTA_KWN	varchar(16)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
12	TGL_KWN	date			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
13	NO_AKTA_CRAI	varchar(16)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
14	TGL_CRAI	date			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
15	STAT_HKEL	varchar(25)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
16	PDDK_AKH	varchar(10)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
17	JENIS_PKRJUN	varchar(25)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
18	NAMA_LGKP_IBU	varchar(50)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
19	NAMA_LGKP_AYAH	varchar(50)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
20	NO_KK	varchar(16)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
21	terhapus	tinyint(1)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
22	dibuatOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
23	waktuDibuat	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
24	diubahOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
25	waktuDiubah	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns

Gambar 4.2. Implementasi Tabel Biodata Adminduk

Gambar 4.2 merupakan hasil implementasi tabel biodata adminduk yang digunakan untuk menyimpan data pemohon yang ingin melegalisasi dokumen adminduknya.

## 3. Tabel Jenis Dokumen

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	jenisdokid	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
2	kode	varchar(3)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
3	nama	varchar(50)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
4	terhapus	tinyint(1)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
5	dibuatOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
6	waktuDibuat	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
7	diubahOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
8	waktuDiubah	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns

Gambar 4.3. Implementasi Tabel Jenis Dokumen

Gambar 4.3 merupakan hasil implementasi tabel jenis dokumen yang digunakan untuk menyimpan data jenis dokumen adminduk yang akan dilegalisir.

#### 4. Tabel Pejabat

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	pejabatid	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
2	nip	varchar(25)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
3	nama	varchar(50)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
4	jabatan	varchar(50)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
5	terhapus	tinyint(1)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
6	dibuatOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
7	waktuDibuat	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
8	diubahOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
9	waktuDiubah	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns

Gambar 4.4. Implementasi Tabel Pejabat

Gambar 4.4 merupakan hasil implementasi tabel pejabat yang digunakan untuk menyimpan data pejabat yang melegalisir dokumen adminduk.

#### 5. Tabel Register Legalisir

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	reglegid	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
2	pejabatid	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
3	jenisdokid	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
4	nik	varchar(16)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
5	no_reg	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
6	tanggal	date			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
7	terhapus	tinyint(1)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
8	dibuatOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
9	waktuDibuat	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
10	diubahOleh	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
11	waktuDiubah	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns

Gambar 4.5. Implementasi Tabel Register Legalisir

Gambar 4.5 merupakan hasil implementasi tabel register legalisir yang digunakan untuk menyimpan data register legalisir.

#### 6. Tabel Status User

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	userid	int(11)			Tidak	Tidak ada	AUTO_INCREMENT	Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
2	email	varchar(128)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
3	password	varchar(128)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
4	name	varchar(128)			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
5	mobile	varchar(20)			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
6	statusid	tinyint(4)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
7	isDeleted	tinyint(4)			Tidak	0		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
8	createdBy	int(11)			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
9	createdDtm	datetime			Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
10	updatedBy	int(11)			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns
11	updatedDtm	datetime			Ya	NULL		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Distinct values Add to central columns

#### Gambar 4.6. Implementasi Tabel Status User

Gambar 4.6 merupakan hasil implementasi tabel status user yang digunakan untuk menyimpan status hak akses user.

#### 4.1.2 Relasi Antar Tabel

#### Gambar 4.7. Implementasi Tabel Status User

Gambar 4.7 merupakan hasil implementasi relasi antar tabel yang digunakan untuk aplikasi nomor register legalisasi dokumen adminduk.

#### 4.1.3 Fungsi Menambah Data User

Fungsi di bawah ini digunakan untuk menambah data user, dan akan digunakan pada menu Users dengan menggunakan hak akses admin.

```
function addUser()
{
    if($this->isAdmin() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->model('user_model');
        $data['status'] = $this->user_model->getUserStatus();

        $this->global['pageTitle']='Adminduk : Add New User';

        $this->loadViews("addUser",    $this->global,    $data,
NULL);
    }
}
```

#### Gambar 4.8. Fungsi Menampilkan Form Tambah User

Gambar 4.8 merupakan fungsi untuk menampilkan form tambah user di halaman Users, pada hak akses Admin.

```

function addNewUser()
{
    if($this->isAdmin() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->library('form_validation');

        $this->form_validation->set_rules('fname','Full
Name','trim|required|max_length[128]|xss_clean');
        $this->form_validation-
>set_rules('email','Email','trim|required|valid_email|xss_cl
ean|max_length[128]');
        $this->form_validation-
>set_rules('password','Password','required|max_length[20]');
        $this->form_validation-
>set_rules('cpassword','Confirm
Password','trim|required|matches[password]|max_length[20]');
        $this->form_validation-
>set_rules('status','Status','trim|required|numeric');
        $this->form_validation-
>set_rules('mobile','Mobile
Number','required|min_length[10]|xss_clean');

        if($this->form_validation->run() == FALSE)
        {
            $this->addUser();
        }
        else
        {
            $name    =    ucwords(strtolower($this->input-
>post('fname')));
            $email = $this->input->post('email');
            $password = $this->input->post('password');
            $statusId = $this->input->post('status');
            $mobile = $this->input->post('mobile');

            $userInfo    =    array('email'=>$email,
'password'=>getHashedPassword($password),
'statusId'=>$statusId, 'name'=> $name,
                                'mobile'=>$mobile,
'createdBy'=>$this->vendorId,      'createdDtm'=>date('Y-m-d
H:i:s'));

            $this->load->model('user_model');
            $result    =    $this->user_model-
>addNewUser($userInfo);

            if($result > 0)

```

```

        {
            $this->session->set_flashdata('success',
'New User created successfully');
        }
        else
        {
            $this->session->set_flashdata('error',
'User creation failed');
        }

        redirect('addUser');
    }
}

```

Gambar 4.9. Fungsi Validasi dan Inisialisasi Input Data User

Gambar 4.9 merupakan fungsi untuk validasi data yang diinputkan dan inisialisasi data untuk dikirim ke database. Fungsi ini berfungsi di halaman Users, pada hak akses Admin.

```

function addNewUser($userInfo)
{
    $this->db->trans_start();
    $this->db->insert('tbl_users', $userInfo);

    $insert_id = $this->db->insert_id();

    $this->db->trans_complete();

    return $insert_id;
}

```

Gambar 4.10. Fungsi Untuk Menyimpan Data User Baru ke Database

Gambar 4.10 merupakan fungsi untuk menyimpan data user baru ke database. Fungsi ini terdapat di halaman Users, pada hak akses Admin.

#### 4.1.4 Fungsi Menghapus Data User

Fungsi di bawah ini digunakan untuk menghapus data user, dan akan digunakan pada menu Users dengan menggunakan hak akses admin. Fungsi hapus

ini tidak benar-benar menghapus data user dari database, tetapi hanya merubah status dari data user. Status 0 untuk data user aktif, sedangkan 1 untuk data user dengan status terhapus.

```
function deleteUser()
{
    if($this->isAdmin() == TRUE)
    {
        echo(json_encode(array('status'=>'access')));
    }
    else
    {
        $userId = $this->input->post('userId');
        $userInfo = array('isDeleted'=>1, 'updatedBy'=>$this->vendorId,
        'updatedDtm'=>date('Y-m-d H:i:s'));
        $result = $this->user_model->deleteUser($userId, $userInfo);
        if ($result > 0) {
            echo(json_encode(array('status'=>TRUE))); }
        else {
            echo(json_encode(array('status'=>FALSE))); }
        }
    }
}
```

Gambar 4.11. Fungsi Untuk Menghapus Data Terpilih

Gambar 4.11 merupakan fungsi untuk menghapus data user yang dipilih oleh admin. Fungsi ini dapat di akses oleh admin pada halaman Users.

```
function deleteUser($userId, $userInfo)
{
    $this->db->where('userId', $userId);
    $this->db->update('tbl_users', $userInfo);

    return $this->db->affected_rows();
}
```

Gambar 4.12. Fungsi Untuk Menghapus Data User di Database

Gambar 4.12 merupakan fungsi untuk menghapus data user dari database. Fungsi ini dapat di akses oleh admin pada halaman Users.

#### 4.1.5 Fungsi Mengedit Data User

Fungsi di bawah ini digunakan untuk mengedit data user, dan akan digunakan pada menu Users dengan menggunakan hak akses admin.

```
function editOldUsr($userId = NULL)
{
    if($this->isAdmin() == TRUE || $userId == 1)
    {
        $this->loadThis();
    }
    else
    {
        if($userId == null)
        {
            redirect('userListing');
        }

        $data['status'] = $this->user_model->getUserStatus();
        $data['userInfo'] = $this->user_model->getUserInfo($userId);

        $this->global['pageTitle'] = 'Adminduk : Edit User';

        $this->loadViews("editOldUsr", $this->global, $data, NULL);
    }
}
```

Gambar 4.13. Fungsi Untuk Menampilkan Form Ubah Data User Lama

Gambar 4.13 merupakan fungsi untuk menampilkan form ubah data user lama. Fungsi ini dapat di akses oleh admin pada halaman Users.

```
function editUser()
{
    if($this->isAdmin() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->library('form_validation');
    }
}
```



```

        $userId = $this->input->post('userId');

        $this->form_validation->set_rules('fname', 'Full
Name', 'trim|required|max_length[128]|xss_clean');
        $this->form_validation-
>set_rules('email', 'Email', 'trim|required|valid_email|xss_c
lean|max_length[128]');
        $this->form_validation-
>set_rules('password', 'Password', 'matches[cpassword]|max_le
ngth[20]');
        $this->form_validation-
>set_rules('cpassword', 'Confirm
Password', 'matches[password]|max_length[20]');
        $this->form_validation-
>set_rules('status', 'Status', 'trim|required|numeric');
        $this->form_validation-
>set_rules('mobile', 'Mobile
Number', 'required|min_length[10]|xss_clean');

        if($this->form_validation->run() == FALSE)
        {
            $this->editOldUsr($userId);
        }
        else
        {
            $name = ucwords(strtolower($this->input-
>post('fname')));
            $email = $this->input->post('email');
            $password = $this->input->post('password');
            $statusId = $this->input->post('status');
            $mobile = $this->input->post('mobile');

            $userInfo = array();

            if(empty($password))
            {
                $userInfo = array('email'=>$email,
'statusId'=>$statusId, 'name'=>$name,
                                'mobile'=>$mobile,
'updatedBy'=>$this->vendorId, 'updatedDtm'=>date('Y-m-d
H:i:s'));
            }
            else
            {
                $userInfo = array('email'=>$email,
'password'=>getHashedPassword($password),
'statusId'=>$statusId,
                                'name'=>ucwords($name),
'mobile'=>$mobile, 'updatedBy'=>$this->vendorId,
                                'updatedDtm'=>date('Y-m-d H:i:s'));
            }
        }
    }
}

```

```

        $result = $this->user_model->editUser($userInfo, $userId);

        if($result == true)
        {
            $this->session->set_flashdata('success', 'User updated successfully');
        }
        else
        {
            $this->session->set_flashdata('error', 'User updation failed');
        }

        redirect('userListing');
    }
}

```

Gambar 4.14. Fungsi Untuk Menampilkan Form Ubah Data User Lama

Gambar 4.14 merupakan fungsi untuk validasi data yang diinputkan dan inisialisasi data untuk dikirim ke database. Fungsi ini berfungsi di halaman Users, pada hak akses Admin.

#### Fungsi untuk menyimpan perubahan data di database

```

function editUser($userInfo, $userId)
{
    $this->db->where('userId', $userId);
    $this->db->update('tbl_users', $userInfo);

    return TRUE;
}

```

Gambar 4.15. Fungsi Untuk Menyimpan Perubahan Data User di Database

Gambar 4.15 merupakan fungsi untuk menyimpan perubahan data user di database. Fungsi ini berfungsi di halaman Users, pada hak akses Admin.

#### 4.1.6 Meregister Dokumen Adminduk

Fungsi di bawah ini digunakan untuk meregister dokumen adminduk. Meregister dokumen adminduk meliputi proses input data dokumen. Fungsi ini terdapat pada menu Register Legalisasi dengan menggunakan hak akses user.

```
function editOldRlg($reglegId = NULL)
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        // if($reglegId == null)
        // {
        //     redirect('DaftarRegLeg');
        // }

        $this->load->model('regleg_model');
        $data['cbxPejabat'] = $this->CbxEjabat();
        $data['cbxJenis'] = $this->CbxEjenis();
        $data['reglegInfo'] = $this->regleg_model->getRegLegInfo($reglegId);

        $this->global['pageTitle'] = 'Adminduk : Ubah
        Register Legalisir';

        $this->loadViews("regleg/editOldRlg", $this->global, $data, NULL);
    }
}
```

Gambar 4.16. Fungsi Untuk Menampilkan Informasi di Form Register Legalisir

Gambar 4.16 merupakan fungsi yang digunakan untuk menampilkan informasi pada form register legalisir yang akan diubah. Informasi berupa jenis dokumen dan pejabat legalisir yang ditampilkan juga menggunakan *combo box*.

```
function editRegLeg()
{
    if($this->isForAll() == TRUE)
    {
```

```

        $this->loadThis();
    }
    else
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('pejabatId','Pejabat','trim|required|max_length[
50]|xss_clean');
        $this->form_validation-
>set_rules('jenisdokId','Jenis
Dokumen','trim|required|max_length[50]|xss_clean');
        $this->form_validation->set_rules('nik_', 'Nomor
Induk
Kependudukan','trim|required|max_length[16]|xss_clean');

        if($this->form_validation->run() == FALSE)
        {
            $this->editOldRlg();
        }
        else
        {
            $pejabat = $this->input->post('pejabatId');
            $jenis = $this->input->post('jenisdokId');
            $nik = $this->input->post('nik_');
            $tanggal = $this->input->post('tanggal');
            $y__ = date('Y', strtotime($tanggal));

            $DATANOBARU          =          $this->regleg_model-
>getnewnoreg($y__, $jenis);
            $GETNOREGBARU = '';
            if(count($DATANOBARU)==0)    $GETNOREGBARU  =
1;
            else    $GETNOREGBARU    =    $DATANOBARU[0]-
>noregakhir+1;

            if($jenis=='4'){
                $dtlweek          =          $this->regleg_model-
>getreglegInfoByNIK($nik);
                echo '<pre>';
                print_r($dtlweek);
                echo '</pre>';

                if(strtotime($tanggal)>strtotime($dtlweek[0]-
>tanggal)+604800){
                    $GETNOREGBARU = $GETNOREGBARU;
                }
                else{
                    $GETNOREGBARU          =          $dtlweek[0]-
>no_reg;

```

```

        }

        echo $GETNOREGBARU;
    }

    $reglegInfo = array('tanggal'=>$tanggal,
'no_reg'=>$GETNOREGBARU,      'pejabatId'=>$pejabat,
'jenisId'=>$jenis,      'nik'=>$nik,      'dibuatOleh'=>$this->vendorId, 'waktuDibuat'=>date('Y-m-d H:i:s'));

    $result = $this->regleg_model->insertRegLeg($reglegInfo);
    redirect('viewTemplate/'.$result);
}
}
}

```

Gambar 4.17. Fungsi Untuk Merubah Data Register Legalisir

Gambar 4.17 merupakan fungsi yang digunakan untuk merubah data register legalisir.

```

/**
 * Fungsi ini digunakan untuk memperbaharui informasi
register legalisir
 * @param array $reglegInfo : Ini informasi register
legalisir yang telah diperbaharui
 * @param number $reglegId : Ini register legalisir id
 */
function insertRegLeg($reglegInfo)
{

    $this->db->trans_start();
    $this->db->insert('tbl_regleg', $reglegInfo);

    $insert_id = $this->db->insert_id();

    $this->db->trans_complete();

    return $insert_id;
}

```

Gambar 4.18. Fungsi Untuk Memperbarui Informasi Register Legalisir

Gambar 4.18 merupakan fungsi yang digunakan untuk memperbarui informasi register legalisir.

#### 4.1.7 Fungsi Mencetak Nomor Register Legalisir

Fungsi di bawah ini digunakan untuk mencetak nomor legalisir. Fungsi ini terdapat pada menu Register Legalisasi dengan menggunakan hak akses user.

```
function cetakTunggal(){
    date_default_timezone_set("Asia/Jakarta");
    $tgl1 = $tgl2 = $searchText = '';

    $id= $this->uri->segment('2');
    if($id!=''){
        $reglegData = $this->regleg_model->cetakTunggal($id);

        $pdf = new FPDF('l','cm',array(15,12));
        $pdf->AddPage();
        $pdf->SetFont('Arial','B',10);
        $pdf->MultiCell(13,0.5,'DINAS KEPENDUDUKAN DAN
PENCATATAN SIPIL '.$n'. ' KABUPATEN BANTUL',0,'C');
        $pdf->SetFont('Arial','',8);
        $pdf->MultiCell(13,0.5,'Tanggal : '.date('d-m-
Y'). ' '.$n'. ' Jam : '.date('H:i'). ' ',0,'C');
        $pdf->SetFont('Arial','B',10);
        $pdf->Ln();
        $pdf->MultiCell(13,0.5,'NO. KARTU KELUARGA :
'.$reglegData[0]->NO_KK,0,'C');
        $pdf->MultiCell(13,0.5,'Nomor Register
Legalisasi Dokumen Adminduk : ',0,'C');
        $pdf->SetFont('Arial','B',15);
        $pdf->MultiCell(13,0.5,$reglegData[0]-
>kode.$reglegData[0]-
>no_reg.'/'.date('Y',strtotime($reglegData[0]-
>rlgtgl)),0,'C');
        $pdf->SetFont('Arial','',8);
        $pdf->Ln();
        $pdf->MultiCell(13,0.5,'Disahkan Oleh',0,'C');
        $pdf->MultiCell(13,0.5,$reglegData[0]-
>jabatan,0,'C');
        $pdf->MultiCell(13,0.5,$reglegData[0]-
>nmpjb."n". 'NIP. '.$reglegData[0]->nip,0,'C');
        $pdf->Ln();
        $pdf->SetFont('Arial','',8);
        $pdf->MultiCell(13,0.5,'Silahkan tulis nomor di
atas pada dokumen adminduk dengan '.$n'. ' tanggal yang
sama, yang telah dilegalisir di Disdukcapil Bantul',0,'C');
        $pdf->MultiCell(13,0.5,'--TERIMA KASIH--
',0,'C');
```

```

        $pdf->Output();
    }
}

```

Gambar 4.19. Fungsi Untuk Mengatur Tampilan Nomor Register Legalisir

Gambar 4.19 merupakan fungsi yang digunakan untuk mengatur tampilan nomor register legalisir yang akan dicetak.

```

function cetakTunggal($id)
{
    $this->db->select('*', jdk.nama as nmjdk , pjb.nama
as nmpjb, rlg.tanggal as rlgtgl');
    $this->db->from('tbl_regleg as rlg');
    $this->db->from('tbl_bioadm as bio', 'bio.bioadmId
= rlg.bioadmId');
    $this->db->join('tbl_pejabat          as          pjb',
'pjb.pejabatId = rlg.pejabatId');
    $this->db->join('tbl_jenisdok          as          jdk',
'jdk.jenisdokId = rlg.jenisdokId');
    $this->db->where('rlg.reglegId', $id);
    $this->db->where('rlg.terhapus', 0);
    $query = $this->db->get();

    $result = $query->result();

    return $result;
}

```

Gambar 4.20. Fungsi Memanggil Data Untuk Nomor Register Legalisir

Gambar 4.20 merupakan fungsi yang digunakan untuk memanggil data nomor register legalisir dari database yang akan dicetak.

#### 4.1.8 Fungsi Mencetak Laporan Legalisir

Fungsi di bawah ini digunakan untuk mencetak laporan legalisir. Laporan dapat di urutkan berdasarkan tanggal dan jenis dokumen adminduk. Fungsi ini terdapat pada menu Register Legalisasi dengan menggunakan hak akses user.

```

function cetakpdf(){
    //2 (tglawal) 3 (tglakhir) 4 (searchText)
    $tgl1 = $tgl2 = $searchText = $jenisdok = '';

    if($this->uri->segment('2')!='-') $tgl1 = date('d-
m-Y',$this->uri->segment('2'));
    else $tgl1 = '';

    if($this->uri->segment('3')!='-') $tgl2 = date('d-
m-Y',$this->uri->segment('3'));
    else $tgl2 = '';

    if($this->uri->segment('4')!='-') $jenisdok =
$this->uri->segment('4');
    else $jenisdok = '';

    if($this->uri->segment('5')!='-') $searchText =
$this->uri->segment('5');
    else $searchText = '';

    $pdf = new FPDF('l','mm','A5');
    // membuat halaman baru
    $pdf->AddPage();
    // setting jenis font yang akan digunakan
    $pdf->SetFont('Arial','B',16);
    // mencetak string
    $pdf->Cell(190,7,'LAPORAN REGISTER
LEGALISIR',0,1,'C');
    $pdf->SetFont('Arial','B',12);

    $reglegData = $this->regleg_model-
>cetakBasedPost($searchText,$tgl1,$tgl2,$jenisdok);

    if($tgl1!='' && $tgl2!=''){
        $pdf->Cell(190,6,'Tanggal '.$tgl1.' s/d
'.$tgl2,0,1,'C');
    }
    if($jenisdok!=''){
        $pdf->Cell(190,6,'Jenis Dokumen :
'.$reglegData[0]->nmjkd,0,1,'C');
    }

    // Memberikan space kebawah agar tidak terlalu
    rapat
    $pdf->Cell(10,7,'',0,1);
    $pdf->SetFont('Arial','B',10);
    $pdf->Cell(10,6,'No.',1,0);
    $pdf->Cell(35,6,'No Register',1,0);
    $pdf->Cell(35,6,'NIK',1,0);
    $pdf->Cell(45,6,'Jenis Dokumen',1,0);

```



```

$pdf->Cell(35,6,'Pejabat Legalisir',1,0);
$pdf->Cell(25,6,'Tanggal',1,1);
$pdf->SetFont('Arial','',10);

$i=1;
foreach ($reglegData as $row){
    $pdf->Cell(10,6,$i,1,0);
    $pdf->Cell(35,6,$row->kode.$row-
>no_reg.'/'.date('Y',strtotime($row->rlgtgl)),1,0);
    $pdf->Cell(35,6,$row->nik,1,0);
    $pdf->Cell(45,6,$row->nmjdk,1,0);
    $pdf->Cell(35,6,$row->nmpjb,1,0);
    $pdf->Cell(25,6,date('d-m-Y',strtotime($row-
>rlgtgl)),1,1);

    $i++;
}

$pdf->Output();
}

```

Gambar 4.21. Fungsi Untuk Mengatur Tampilan Laporan Register Legalisir

Gambar 4.21 merupakan fungsi yang digunakan untuk mengatur tampilan laporan legalisir yang akan dicetak.

```

function cetakBasedPost($searchText = '', $dt1 , $dt2,
$jenisdok)
{
    $this->db->select('*', jdk.nama as nmjdk , pjb.nama
as nmpjb, rlg.tanggal as rlgtgl');
    $this->db->from('tbl_regleg as rlg');
    $this->db->join('tbl_pejabat as pjb', 'pjb.pejabatId
= rlg.pejabatId');
    $this->db->join('tbl_jenisdok as jdk',
'jdk.jenisdokId = rlg.jenisdokId');

    if(!empty($searchText)) {
        $likeCriteria = "(no_dok LIKE
'%" . $searchText . "%')";
        $this->db->where($likeCriteria);
    }
    if(!empty($dt1) && !empty($dt2)){
        $dt1 = date('Y-m-d',strtotime($dt1));
        $dt2 = date('Y-m-d',strtotime($dt2));

        $likeCriteria = "rlg.tanggal between '" . $dt1 . "'
and '" . $dt2 . "' ";
    }
}

```

```

        $this->db->where($likeCriteria);
    }
    if(!empty($jenisdok)) {
        $likeCriteria
"rlg.jenisdokId='". $jenisdok.'";
        $this->db->where($likeCriteria);
    }

    $this->db->where('rlg.terhapus', 0);
    $query = $this->db->get();

    $result = $query->result();

    return $result;
}

```

Gambar 4.22. Fungsi Untuk Memanggil Data Register Legalisir

Gambar 4.22 merupakan fungsi yang digunakan untuk memanggil data register legalisir dari database yang akan dicetak.

#### 4.1.9 Fungsi Menambahkan Data Pejabat Legalisir

```

function addPejabat()
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->model('pejabat_model');

        $this->global['pageTitle'] = 'Adminduk : Buat
Pejabat Baru';

        $this->loadViews("pejabat/addPejabat", $this-
>global, '', NULL);
    }
}

```

Gambar 4.23. Fungsi Untuk Menampilkan Form tambah Pejabat

Gambar 4.23 merupakan fungsi yang digunakan untuk menampilkan form tambah pejabat yang terdapat pada menu Pejabat Legalisir.

```

function addNewPejabat()
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->library('form_validation');

        $this->form_validation->
>set_rules('nip', 'NIP', 'trim|required|max_length[25]|xss_clean');
        $this->form_validation->set_rules('nama', 'Nama
Pejabat', 'trim|required|max_length[50]|xss_clean');
        $this->form_validation->
>set_rules('jabatan', 'Jabatan', 'trim|required|max_length[50]
|xss_clean');
        if($this->form_validation->run() == FALSE)
        {
            $this->addPejabat();
        }
        else
        {
            $nip = $this->input->post('nip');
            $nama = $this->input->post('nama');
            $jabatan = $this->input->post('jabatan');
            $pejabatInfo = array('nip'=>$nip,
'nama'=>$nama, 'jabatan'=>$jabatan, 'dibuatOleh'=>$this-
>vendorId, 'waktuDibuat'=>date('Y-m-d H:i:s'));

            $this->load->model('pejabat_model');
            $result = $this->pejabat_model-
>addNewPejabat($pejabatInfo);
            if($result > 0)
                $this->session->set_flashdata('success',
'Pejabat Baru Berhasil Dibuat');
            else
                $this->session->set_flashdata('error',
'Pejabat Baru Gagal Dibuat');

            redirect('pejabat/addPejabat');
        }
    }
}

```

Gambar 4.24. Fungsi Untuk Menambah Pejabat Baru di Sistem

Gambar 4.24 merupakan fungsi yang digunakan untuk menambah pejabat baru ke dalam sistem.

```

/**
 * Fungsi ini untuk menambahkan pejabat baru
 * @return number $insert_id : Ini ID tambahan terakhir
 */
function addNewPejabat($pejabatInfo)
{
    $this->db->trans_start();
    $this->db->insert('tbl_pejabat', $pejabatInfo);

    $insert_id = $this->db->insert_id();

    $this->db->trans_complete();

    return $insert_id;
}

```

Gambar 4.25. Fungsi Untuk Menambah Pejabat Baru di Database

Gambar 4.25 merupakan fungsi yang digunakan untuk menambah data pejabat baru ke dalam database.

#### 4.1.10 Fungsi Menghapus Data Pejabat Legalisir

Fungsi di bawah ini digunakan untuk menghapus data pejabat legalisir, dan akan digunakan pada menu Pejabat Legalisir dengan menggunakan hak akses user. Fungsi hapus ini tidak benar-benar menghapus data pejabat legalisir dari database, tetapi hanya merubah status dari data pejabat legalisir. Status 0 untuk data pejabat legalisir aktif, sedangkan 1 untuk data pejabat legalisir dengan status terhapus.

```

function deletePejabat()
{
    if($this->isForAll() == TRUE)
    {
        echo(json_encode(array('status'=>'access')));
    }
    else
    {
        $pejabatId = $this->input->post('pejabatId');
    }
}

```

```

        $pejabatInfo
array('terhapus'=>1, 'diubahOleh'=>$this->vendorId,
'waktuDiubah'=>date('Y-m-d H:i:s'));

        $result = $this->pejabat_model-
>deletePejabat($pejabatId, $pejabatInfo);

        if ($result > 0) {
echo(json_encode(array('status'=>TRUE))); }
        else {
echo(json_encode(array('status'=>FALSE))); }
    }
}

```

Gambar 4.26. Fungsi Untuk Menghapus Data Pejabat Legalisir

Gambar 4.26 merupakan fungsi yang digunakan untuk menghapus data pejabat menggunakan pejabatId.

```

function deletePejabat($pejabatId, $pejabatInfo)
{
    $this->db->where('pejabatId', $pejabatId);
    $this->db->update('tbl_pejabat', $pejabatInfo);
    return $this->db->affected_rows();
}

```

Gambar 4.27. Fungsi Untuk Menghapus Informasi Pejabat Legalisir

Gambar 4.27 merupakan fungsi yang digunakan untuk menghapus informasi pejabat legalisir.

#### 4.1.11 Fungsi Mengedit Data Pejabat Legalisir

Fungsi di bawah ini digunakan untuk mengedit data pejabat legalisir, dan akan digunakan pada menu Pejabat Legalisir dengan menggunakan hak akses user.

```

function editOldPjb($pejabatId = NULL)
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
}

```

```

    }
    else
    {
        if($pejabatId == null)
        {
            redirect('DaftarPejabat');
        }

        $data['pejabatInfo'] = $this->pejabat_model-
>getPejabatInfo($pejabatId);

        $this->global['pageTitle'] = 'Adminduk : Ubah
Pejabat';

        $this->loadViews("pejabat/editOldPjb", $this-
>global, $data, NULL);
    }
}

```

Gambar 4.28. Fungsi Untuk Menampilkan Informasi Pejabat Legalisir

Gambar 4.28 merupakan fungsi yang digunakan untuk menampilkan informasi pejabat legalisir yang akan diedit.

```

function editPejabat()
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->library('form_validation');

        $pejabatId = $this->input->post('pejabatId');

        $this->form_validation-
>set_rules('nip', 'NIP', 'trim|required|max_length[25]|xss_cl
ean');
        $this->form_validation->set_rules('nama', 'Nama
Pejabat', 'trim|required|max_length[50]|xss_clean');
        $this->form_validation-
>set_rules('jabatan', 'Jabatan', 'trim|required|max_length[50
]|xss_clean');

        if($this->form_validation->run() == FALSE)
        {

```

```

        $this->editOldPjb($pejabatId);
    }
    else
    {
        $nip = $this->input->post('nip');
        $nama = $this->input->post('nama');
        $jabatan = $this->input->post('jabatan');

        $pejabatInfo = array('nip'=>$nip,
'nama'=>$nama, 'jabatan'=>$jabatan,
                                'diubahOleh'=>$this-
>vendorId, 'waktuDiubah'=>date('Y-m-d H:i:s'));

        $result = $this->pejabat_model-
>editPejabat($pejabatInfo, $pejabatId);
        if($result == true)
        {
            $this->session-
>set_flashdata('success', 'Data Pejabat Berhasil Diubah
!');
        }
        else
        {
            $this->session->set_flashdata('error',
'Data Pejabat Gagal Diubah');
        }
        redirect('DaftarPejabat');
    }
}

```

Gambar 4.29. Fungsi Untuk Merubah Data Pejabat Legalisir

Gambar 4.29 merupakan fungsi yang digunakan untuk merubah data pejabat legalisir.

```

function editPejabat($pejabatInfo, $pejabatId)
{
    $this->db->where('pejabatId', $pejabatId);
    $this->db->update('tbl_pejabat', $pejabatInfo);
    return TRUE;
}

```

Gambar 4.30. Fungsi Untuk Memperbarui Informasi Pejabat Legalisir

Gambar 4.30 merupakan fungsi yang digunakan untuk memperbarui informasi pejabat legalisir.

#### 4.1.12 Fungsi Menambahkan Jenis Dokumen Adminduk

Fungsi di bawah ini digunakan untuk menambah jenis dokumen adminduk, dan akan digunakan pada menu Jenis Dokumen Adminduk dengan menggunakan hak akses user.

```
function addJenisDok()
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->model('jenisdok_model');

        $this->global['pageTitle'] = 'Adminduk : Buat
Jenis Dokumen Baru';

        $this->loadViews("jenisdok/addJenisDok", $this-
>global, '', NULL);
    }
}
```

Gambar 4.31. Fungsi Untuk Menampilkan Form Tambah Jenis Dokumen

Gambar 4.31 merupakan fungsi yang digunakan untuk menampilkan form tambah jenis dokumen.

```
function addNewJenisDok()
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        $this->load->library('form_validation');

        $this->form_validation-
>set_rules('kode', 'Kode', 'trim|required|max_length[3]|xss_c
lean');
        $this->form_validation->set_rules('nama', 'Nama
Jenis Dokumen', 'trim|required|max_length[50]|xss_clean');
```



```

        if($this->form_validation->run() == FALSE)
        {
            $this->addJenisDok();
        }
        else
        {
            $kode = $this->input->post('kode');
            $nama = $this->input->post('nama');
            $jenisdokInfo =
array('kode'=>$kode, 'nama'=>$nama,      'dibuatOleh'=>$this-
>vendorId, 'waktuDibuat'=>date('Y-m-d H:i:s'));

            $this->load->model('jenisdok_model');
            $result = $this->jenisdok_model-
>addNewJenisDok($jenisdokInfo);

            if($result > 0)
                $this->session-
>set_flashdata('success', 'Jenis Dokumen Baru Berhasil
Dibuat');
            else
                $this->session->set_flashdata('error',
'Jenis Dokumen Baru Gagal Dibuat');

            redirect('jenisdok/addJenisDOK');
        }
    }
}

```

Gambar 4.32. Fungsi Untuk Menambah Jenis Dokumen Baru

Gambar 4.32 merupakan fungsi yang digunakan untuk menambah jenis dokumen baru ke dalam system.

```

function addNewJenisDok($jenisdokInfo)
{
    $this->db->trans_start();
    $this->db->insert('tbl_jenisdok', $jenisdokInfo);
    $insert_id = $this->db->insert_id();
    $this->db->trans_complete();

    return $insert_id;
}

```

Gambar 4.33. Fungsi Untuk Menambahkan Jenis Dokumen di Database

Gambar 4.33 merupakan fungsi yang digunakan untuk menambah jenis dokumen baru ke dalam database.

#### 4.1.13 Fungsi Menghapus Jenis Dokumen Adminduk

Fungsi di bawah ini digunakan untuk menghapus jenis dokumen, dan akan digunakan pada menu Jenis Dokumen dengan menggunakan hak akses user. Fungsi hapus ini tidak benar-benar menghapus jenis dokumen dari database, tetapi hanya merubah status dari data jenis dokumen. Status 0 untuk data jenis dokumen aktif, sedangkan 1 untuk data jenis dokumen dengan status terhapus.

```
function deleteJenisDok()
{
    if($this->isForAll() == TRUE)
    {
        echo(json_encode(array('status'=>'access')));
    }
    else
    {
        $jenisdokId = $this->input->post('jenisdokId');
        $jenisdokInfo =
array('terhapus'=>1, 'diubahOleh'=>$this->vendorId,
'waktuDiubah'=>date('Y-m-d H:i:s'));

        $result = $this->jenisdok_model-
>deleteJenisDok($jenisdokId, $jenisdokInfo);

        if ($result > 0) {
echo(json_encode(array('status'=>TRUE))); }
        else {
echo(json_encode(array('status'=>FALSE))); }
    }
}
```

Gambar 4.34. Fungsi Untuk Menghapus Jenis Dokumen

Gambar 4.34 merupakan fungsi yang digunakan untuk menghapus jenis dokumen menggunakan jenisdokId.

```
function deleteJenisDok($jenisdokId, $jenisdokInfo)
{
    $this->db->where('jenisdokId', $jenisdokId);
    $this->db->update('tbl_jenisdok', $jenisdokInfo);

    return $this->db->affected_rows();
}
```

#### Gambar 4.35. Fungsi Untuk Menghapus Informasi Jenis Dokumen

Gambar 4.35 merupakan fungsi yang digunakan untuk menghapus informasi jenis dokumen pada database.

#### 4.1.14 Fungsi Mengedit Jenis Dokumen Adminduk

Fungsi di bawah ini digunakan untuk mengedit data jenis dokumen, dan akan digunakan pada menu Jenis Dokumen dengan menggunakan hak akses user.

```
function editOldJdk($jenisdokId = NULL)
{
    if($this->isForAll() == TRUE)
    {
        $this->loadThis();
    }
    else
    {
        if($jenisdokId == null)
        {
            redirect('DaftarJenisDok');
        }

        $data['jenisdokInfo'] = $this->jenisdok_model->getJenisDokInfo($jenisdokId);

        $this->global['pageTitle'] = 'Adminduk : Ubah
        Jenis Dokumen';
        $this->loadViews("jenisdok/editOldJdk", $this->global, $data, NULL);
    }
}
```

#### Gambar 4.36. Fungsi Untuk Menampilkan Informasi Jenis Dokumen

Gambar 4.36 merupakan fungsi yang digunakan untuk menampilkan informasi jenis dokumen yang akan diedit..

```
function editJenisDok()
{
    if($this->isForAll() == TRUE)
    {
```

```

        $this->loadThis();
    }
    else
    {
        $this->load->library('form_validation');
        $jenisdokId = $this->input->post('jenisdokId');
        $this->form_validation-
>set_rules('kode','Kode','trim|required|max_length[3]|xss_c
lean');
        $this->form_validation->set_rules('nama','Nama
Jenis Dokumen','trim|required|max_length[50]|xss_clean');

        if($this->form_validation->run() == FALSE)
        {
            $this->editOldJdk($jenisdokId);
        }
        else
        {
            $kode = $this->input->post('kode');
            $nama = $this->input->post('nama');
            $jenisdokInfo = array('kode'=>$kode,
'nama'=>$nama,
'waktuDiubah'=>date('Y-m-d H:i:s'));

            $result = $this->jenisdok_model-
>editJenisDok($jenisdokInfo, $jenisdokId);

            if($result == true)
            {
                $this->session-
>set_flashdata('success', 'Data Jenis Dokumen Berhasil
Diubah !');
            }
            else
            {
                $this->session->set_flashdata('error',
'Data Jenis Dokumen Gagal Diubah');
            }

            redirect('DaftarJenisDok');
        }
    }
}

```

Gambar 4.37. Fungsi Untuk Merubah Data Jenis Dokumen

Gambar 4.37 merupakan fungsi yang digunakan untuk merubah data jenis dokumen..

```

function editJenisDok($jenisdokInfo, $jenisdokId)
{
    $this->db->where('jenisdokId', $jenisdokId);
    $this->db->update('tbl_jenisdok', $jenisdokInfo);

    return TRUE;
}

```

Gambar 4.38. Fungsi Untuk Memperbatui Informasi Jenis Dokumen

Gambar 4.38 merupakan fungsi yang digunakan untuk memperbarui informasi jenis dokumen..

#### 4.1.15 Koneksi Basis Data

Sebelum data yang ada dalam database digunakan, maka yang harus dilakukan adalah melakukan koneksi database. Berikut kode program yang digunakan untuk koneksi database.

```

$active_group = 'default';
$active_record = TRUE;

$db['default']['hostname'] = 'localhost';
$db['default']['username'] = 'root';
$db['default']['password'] = '';
$db['default']['database'] = 'db_legalisir';
$db['default']['dbdriver'] = 'mysqli';
$db['default']['dbprefix'] = '';
$db['default']['pconnect'] = TRUE;
$db['default']['db_debug'] = TRUE;
$db['default']['cache_on'] = FALSE;
$db['default']['cachedir'] = '';
$db['default']['char_set'] = 'utf8';
$db['default']['dbcollat'] = 'utf8_general_ci';
$db['default']['swap_pre'] = '';
$db['default']['autoinit'] = TRUE;
$db['default']['stricton'] = FALSE;

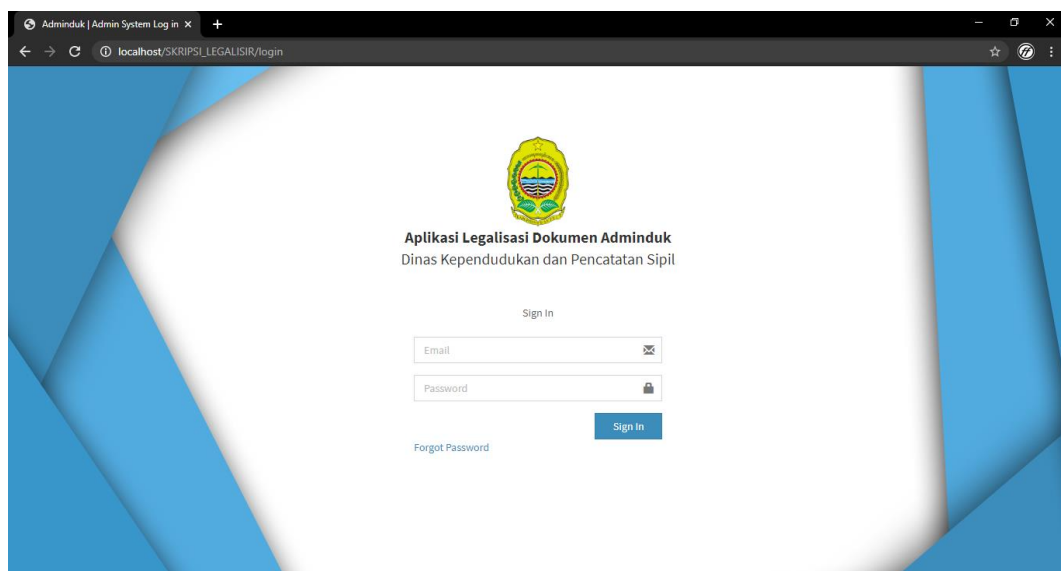
```

Gambar 4.37. Koneksi Database

## 4.2 Pembahasan Sistem

Terdapat 2 bagian pada aplikasi legalisasi ini, yaitu untuk admin dan user (petugas legalisir). Admin mempunyai hak akses menyeluruh,. Perbedaannya hak akses untuk user dengan admin yaitu pada admin dapat menambahkan hak akses baru untuk user (staf).

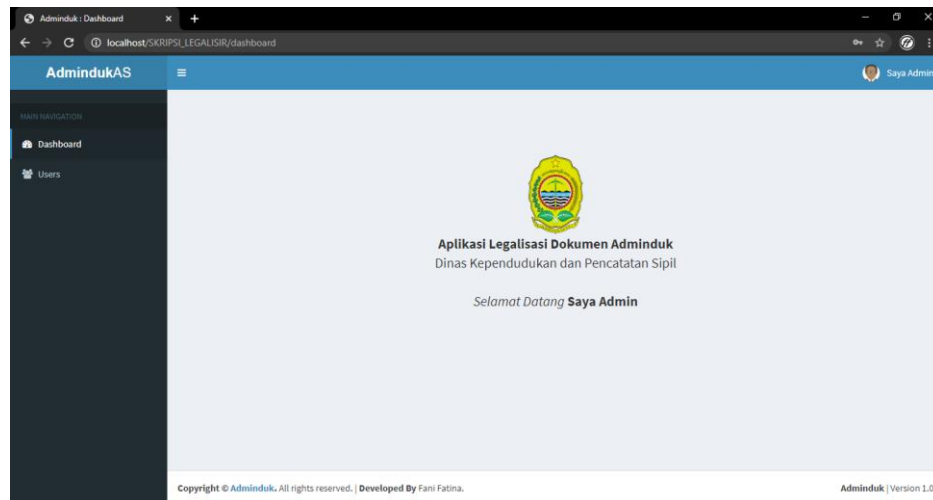
### 4.2.1 Halaman Login



Gambar 4.38. Halaman Login

Gambar 4.38 merupakan halaman untuk login yang dapat digunakan secara bersamaan dengan menggunakan hak akses admin maupun hak akses user.

## 4.2.2 Halaman Utama Admin



Gambar 4.39. Halaman Utama Admin

Gambar 4.39 merupakan halaman utama admin setelah proses login berhasil. Terdapat 2 menu pada bagian kiri yaitu Dashboard dan Users.

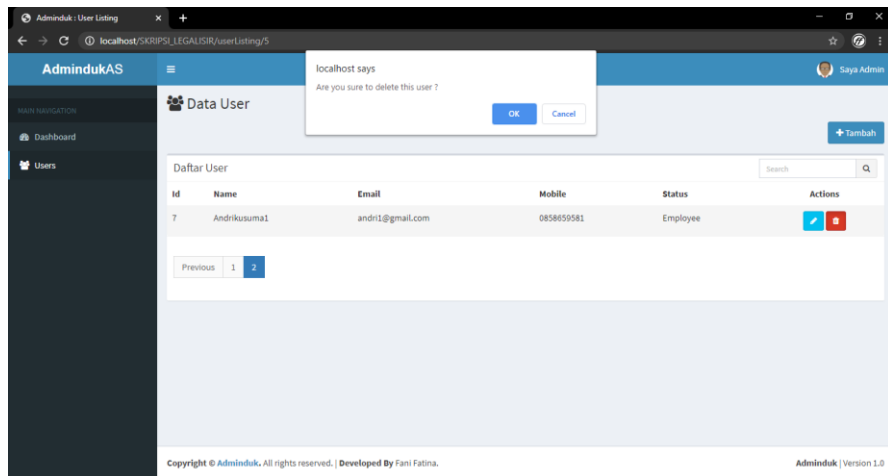
## 4.2.3 Halaman Form Tambah User

The screenshot shows the 'Add New User' form in the AdmindukAS application. The browser address bar shows 'localhost/SKRIPSI\_LEGALISIR/addUser'. The application has a blue header with 'AdmindukAS' and a user profile 'Saya Admin'. A dark sidebar on the left contains 'MAIN NAVIGATION' with 'Dashboard' and 'Users' options. The main content area is titled 'Data User' and contains a 'Formulir Data User Baru' with the following fields: 'Nama Lengkap', 'Alamat Email', 'Password', 'Konfirmasi Password', 'Nomor Handphone', and 'Status Employee'. There are 'Submit' and 'Reset' buttons at the bottom of the form. The footer includes 'Copyright © Adminduk. All rights reserved. | Developed By Fani Fatina.' and 'Adminduk | Version 1.0'.

Gambar 4.40. Halaman Form Tambah User

Gambar 4.39 merupakan halaman form tambah user baru. Menu ini hanya ada pada hak akses admin.

#### 4.2.4 Halaman Proses Hapus Data User



Gambar 4.41. Halaman Proses Hapus Data User

Gambar 4.41 merupakan halaman saat admin melakukan proses hapus data user. Pada daftar user di bagian kanan terdapat tombol untuk menghapus akun user.

#### 4.2.5 Halaman Form Edit Data User

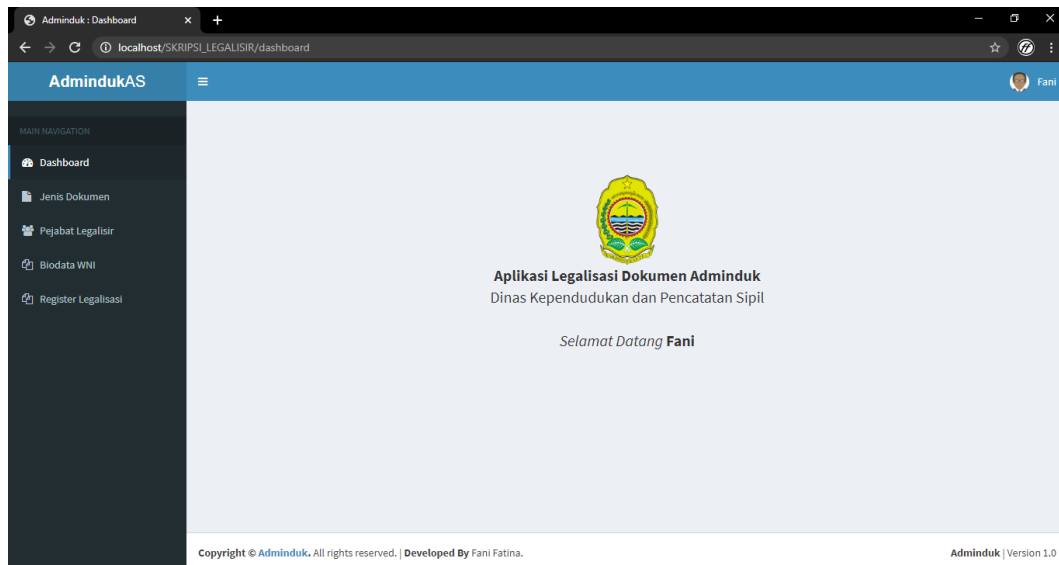
The screenshot shows the 'AdminindukAS' application interface for editing a user. The sidebar on the left has 'Dashboard' and 'Users' options. The main content area is titled 'Data User' and contains a form titled 'Formulir Data User Baru'. The form has fields for 'Nama Lengkap' (filled with 'Fani'), 'Alamat Email' (filled with 'user@user.com'), 'Password', 'Konfirmasi Password', 'Nomor Handphone' (filled with '0858659512'), and 'Status' (filled with 'Employee'). There are 'Submit' and 'Reset' buttons at the bottom of the form. The footer includes 'Copyright © Admininduk. All rights reserved. | Developed By Fani Fatma.' and 'Admininduk | Version 1.0'.

Gambar 4.42. Halaman Form Edit Data User

Gambar 4.42 merupakan halaman yang digunakan admin untuk mengedit data user yang sudah ada.



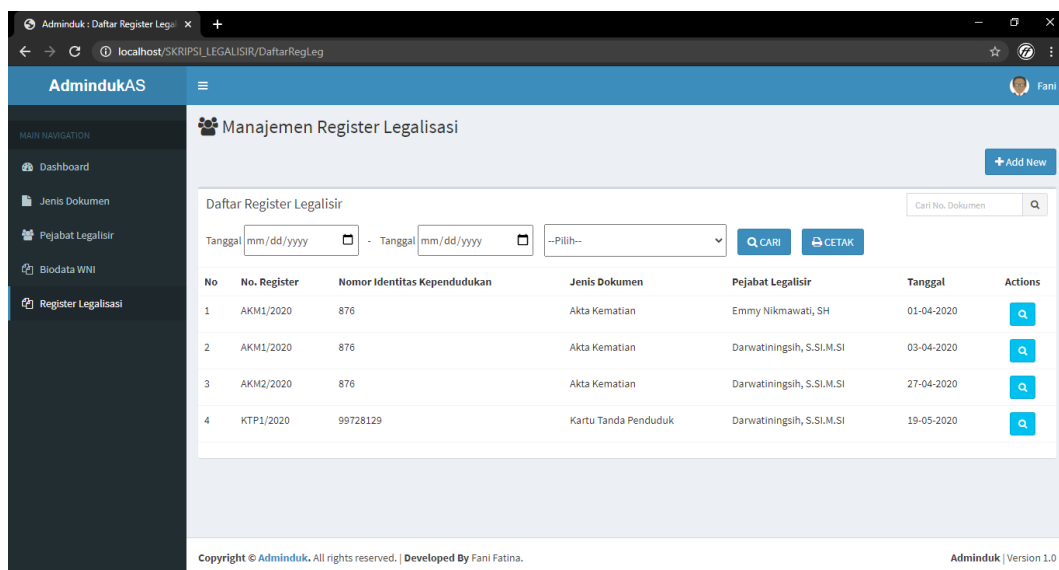
## 4.2.6 Halaman Utama User



Gambar 4.43. Halaman Utama User

Gambar 4.43 merupakan halaman utama user setelah proses login berhasil. Terdapat 5 menu pada bagian kiri yaitu Dashboard, Jenis Dokumen, Pejabat Legalisir, Biodata WNI dan Register Legalisir.

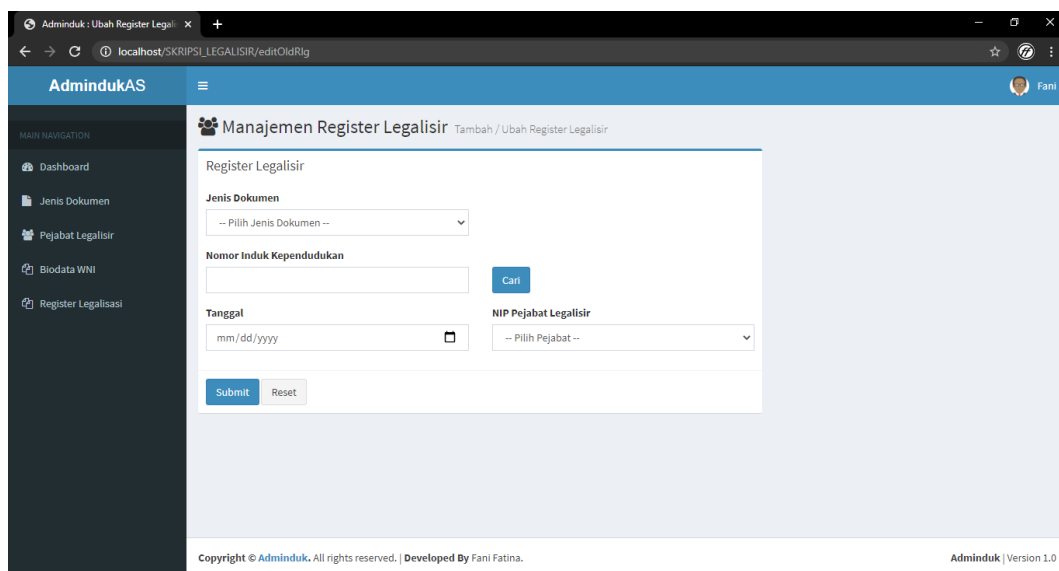
## 4.2.7 Halaman Menu Register Legalisir



Gambar 4.44. Halaman Menu Register Legalisir

Gambar 4.44 merupakan halaman menu register legalisir. Terdapat kolom untuk memfilter data register legalisir yang nantinya dapat di cetak sebagai bentuk laporan. Pada bagian bawah kolom untuk filter terdapat daftar register legalisir yang sudah tersimpan di database.

#### 4.2.8 Halaman Form Tambah Register Legalisir



The screenshot displays the 'Manajemen Register Legalisir' form within the AdmindukAS application. The form is titled 'Register Legalisir' and contains the following fields and controls:

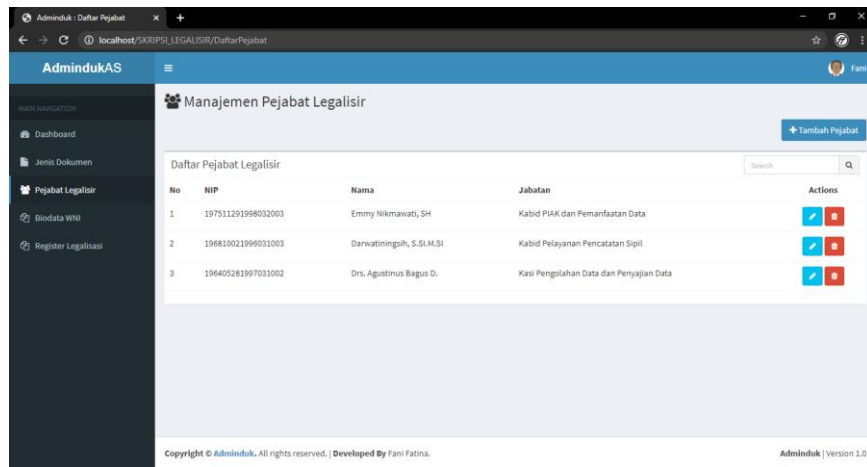
- Jenis Dokumen:** A dropdown menu with the placeholder text '-- Pilih Jenis Dokumen --'.
- Nomor Induk Kependudukan:** A text input field.
- Tanggal:** A date picker with the format 'mm/dd/yyyy'.
- NIP Pejabat Legalisir:** A dropdown menu with the placeholder text '-- Pilih Pejabat --'.
- Buttons:** A 'Cari' button next to the 'Nomor Induk Kependudukan' field, and 'Submit' and 'Reset' buttons at the bottom of the form.

The sidebar on the left shows the main navigation menu with options: Dashboard, Jenis Dokumen, Pejabat Legalisir, Biodata WNI, and Register Legalisasi. The footer of the page includes the text 'Copyright © Adminduk. All rights reserved. | Developed By Fani Fatina.' and 'Adminduk | Version 1.0'.

Gambar 4.45. Halaman Form Tambah Register Legalisir

Gambar 4.45 merupakan halaman yang digunakan user untuk menambahkan data dokumen adminduk untuk mendapatkan nomor register legalisir yang nantinya akan dicetak.

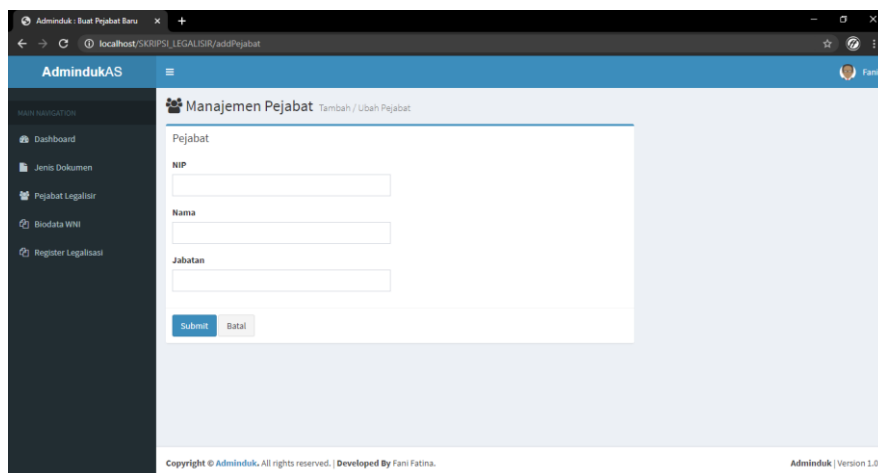
## 4.2.9 Halaman Menu Pejabat Legalisir



Gambar 4.46. Halaman Menu Pejabat Legalisir

Gambar 4.46 merupakan halaman menu pejabat legalisir yang berisi daftar pejabat legalisir. Pada bagian kanan di setiap data pejabat terdapat tombol untuk mengedit dan menghapus.

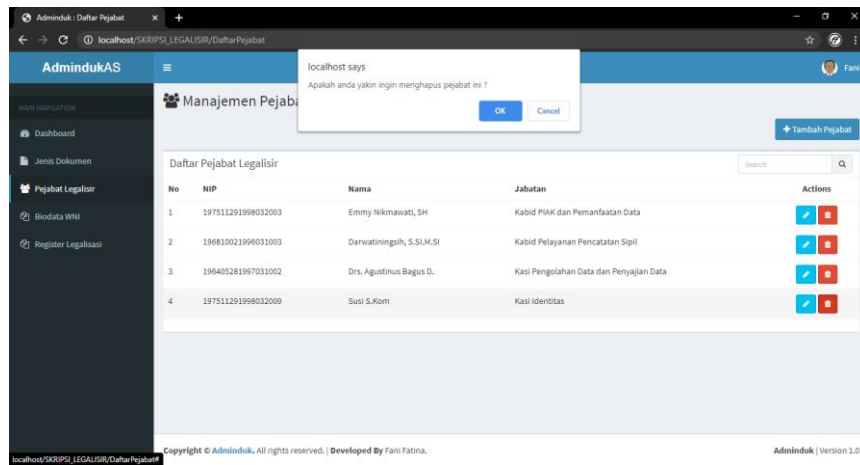
## 4.2.10 Halaman Form Tambah Pejabat Legalisir



Gambar 4.47. Halaman Form Tambah Pejabat Legalisir

Gambar 4.47 merupakan halaman yang digunakan oleh user untuk menambahkan data pejabat legalisir baru.

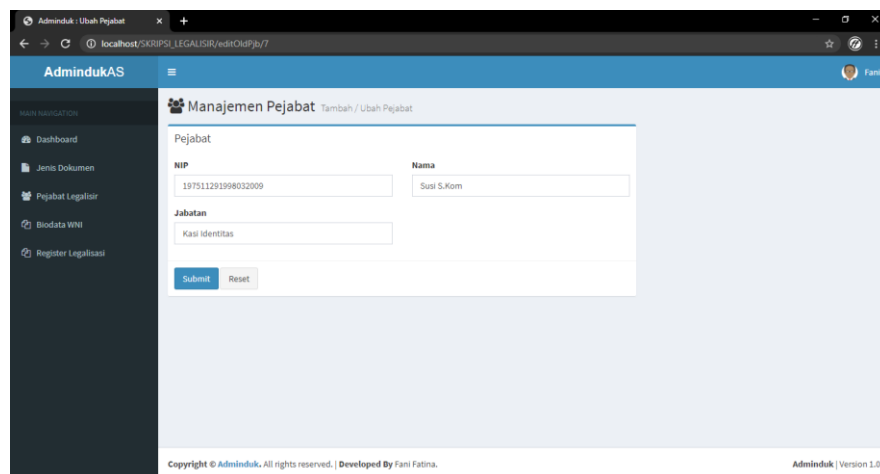
#### 4.2.11 Halaman Proses Hapus Data Pejabat Legalisir



Gambar 4.48. Halaman Proses Hapus Data Pejabat Legalisir

Gambar 4.48 merupakan halaman saat proses hapus data pejabat legalisir, yang sebelumnya user memilih data pejabat yang akan dihapus lalu menekam tombol hapus yang ada di sebelah kanan data pejabat legalisir.

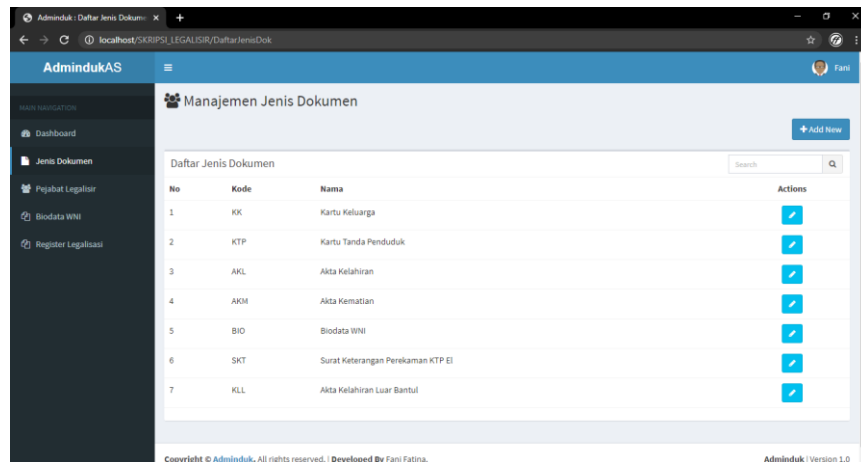
#### 4.2.12 Halaman Form Edit Pejabat Legalisir



Gambar 4.49. Halaman Form Edit Data Pejabat Legalisir

Gambar 4.49 merupakan halaman yang digunakan admin untuk mengedit data pejabat legalisir yang sudah ada.

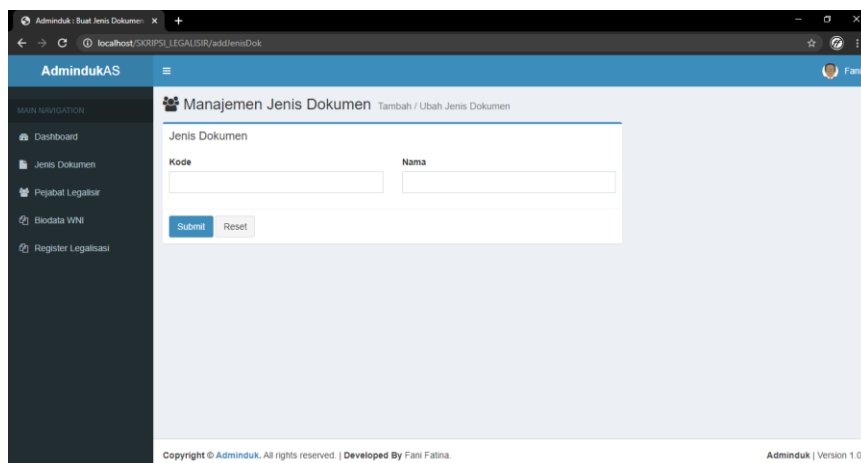
#### 4.2.13 Halaman Menu Jenis Dokumen



Gambar 4.50. Halaman Menu Jenis Dokumen

Gambar 4.50 merupakan halaman menu jenis dokumen yang berisi daftar jenis dokumen. Pada bagian kanan di setiap data pejabat etrdapat tombol untuk mengedit dan menghapus.

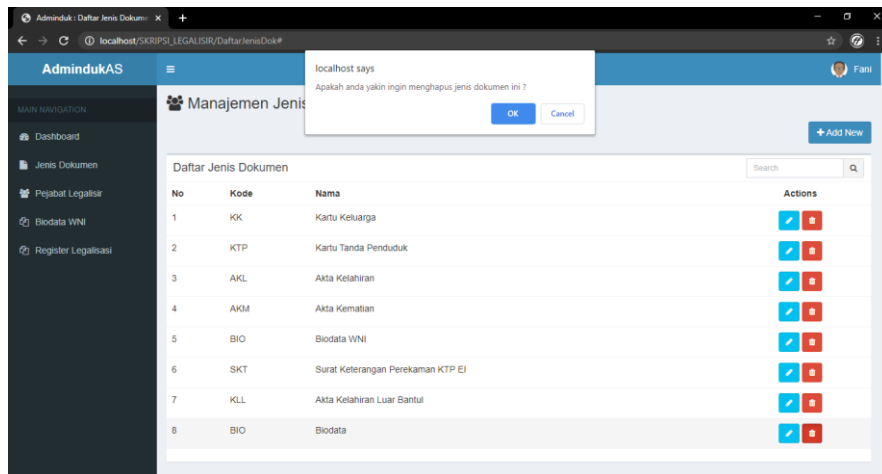
#### 4.2.14 Halaman Form Tambah Jenis Dokumen



Gambar 4.51. Halaman Form Tambah Jenis Dokumen

Gambar 4.51 merupakan halaman yang digunakan oleh user untuk menambahkan data jenis dokumen baru.

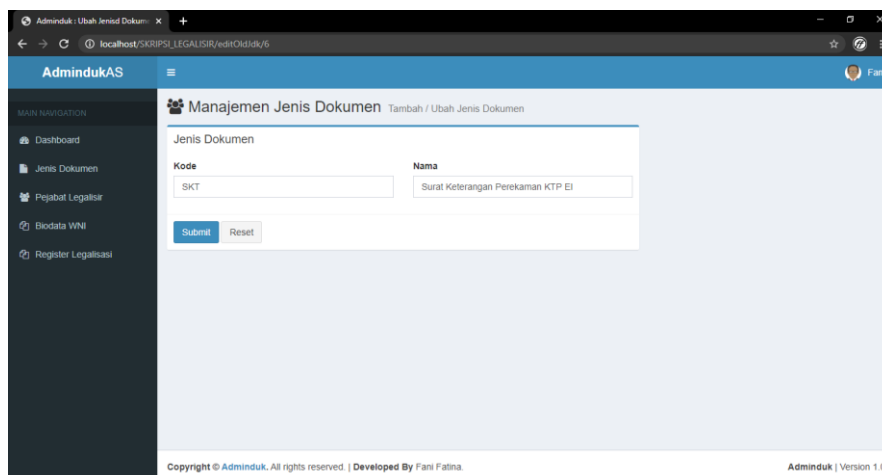
#### 4.2.15 Halaman Proses Hapus Jenis Dokumen



Gambar 4.52 Halaman Proses Hapus Jenis Dokumen

Gambar 4.52 merupakan halaman proses hapus jenis dokumen, yang sebelumnya user memilih jenis dokumen yang akan dihapus lalu menekam tombol hapus yang ada di sebelah kanan jenis dokumen.

#### 4.2.16 Halaman Form Edit Jenis Dokumen



Gambar 4.53. Halaman Form Edit Jenis Dokumen

Gambar 4.53 merupakan halaman yang digunakan admin untuk mengedit jenis dokumen yang sudah ada.