# BAB IV

**IMPLEMENTASI DAN PEMBAHASAN SISTEM**

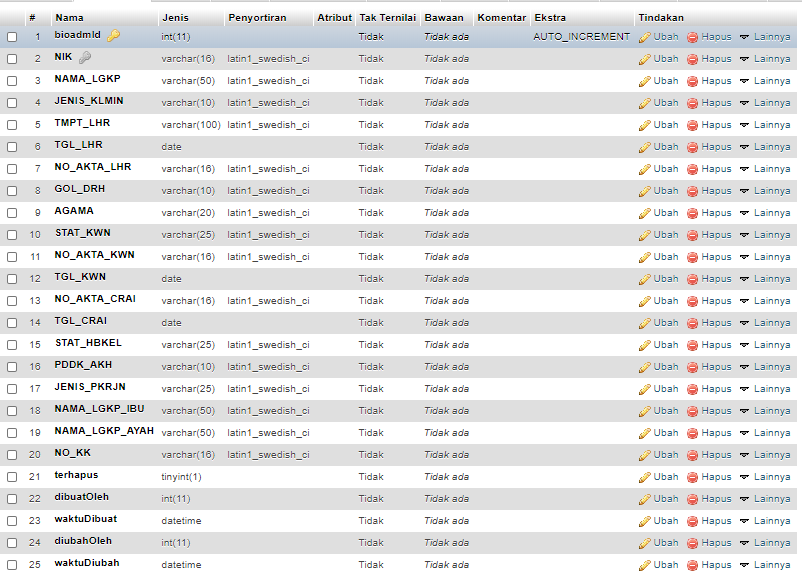
## Implementasi Sistem

Tahap implementasi bertujuan untuk menyiapkan kegiatan penerapan sistem sesuai dengan rancangan yang telah ditentukan. Selanjutnya akan dilakukan tahapan pengujian sistem guna meminamilisir kemungkinan kesalahan yang terjadi.

### Implementasi Basis Data

Pembuatan basis data dan tabel merupakan tahap awal dalam mengimplementasian sebuah sistem. Digitalisasi Arsip Pindah Datang Di Dinas Kependudukan dan Pencatatan Sipil Kabupaten Bantul Berbasis Webs terdiri dari 5 (lima) tabel :

* + - 1. **Tabel Biodata Adminduk (tbl\_bioadm)**



#### Gambar 4.1 Implementasi Tabel Biodata Adminduk (tbl\_bioadm)

Gambar 4.1 merupakan implementasi tabel biodata adminduk (tbl\_bioadm) yang digunakan untuk menyimpan data biodata adminduk.

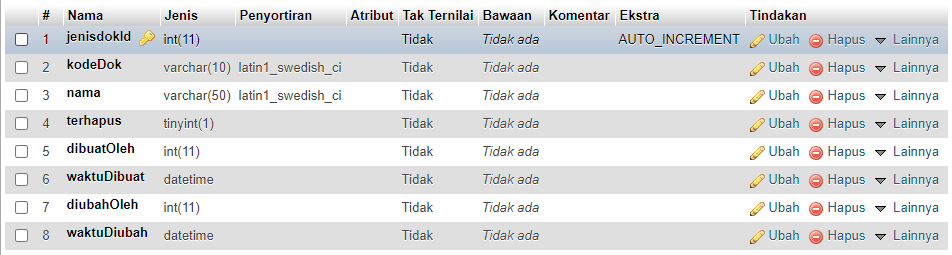
* + - 1. **Tabel Digitalisasi (tbl\_digitalisasi)**



#### Gambar 4.2 Implementasi Tabel tbl\_digitalisasi

Gambar 4.2 merupakan hasil implementasi dari tabel digitalisasi (tbl\_digitalisasi) yang digunakan untuk menyimpan data digitalisasi arsip yang sudah discan.

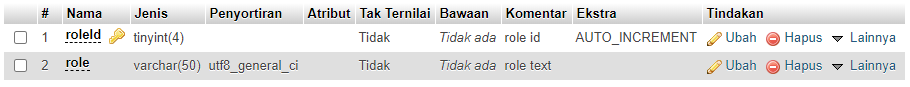
* + - 1. **Tabel Jenis Dokumen (tbl\_jenisdok)**



#### Gambar 4.3 Implementasi Tabel Jenis Dokumen (tbl\_jenisdok)

Gambar 4.3 merupakaan implementasi dari tabel jenis dokumen (tbl\_jenisdok) yang digunakan untuk menyiman data jenis dokumen.

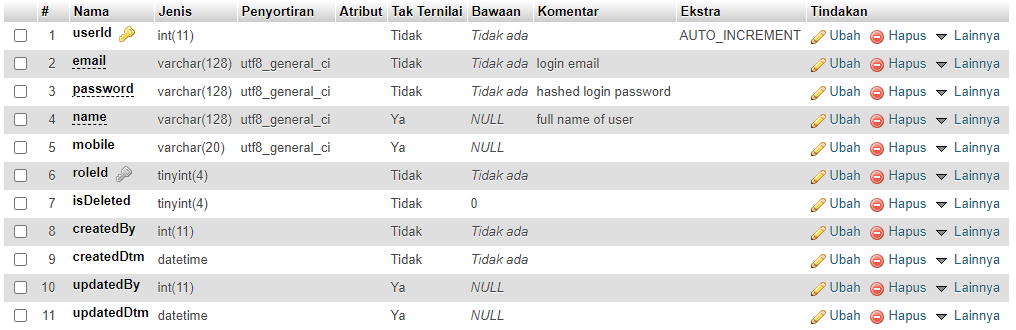
* + - 1. **Tabel Role (tbl\_roles)**



#### Gambar 4.4 Implementasi Tabel Role (tbl\_roles)

Gambar 4.4 merupakan implementasi dari tabel role (tbl\_roles) yang digunakan untuk menyimpan status user, sebagai admin atau sebagai petugas.

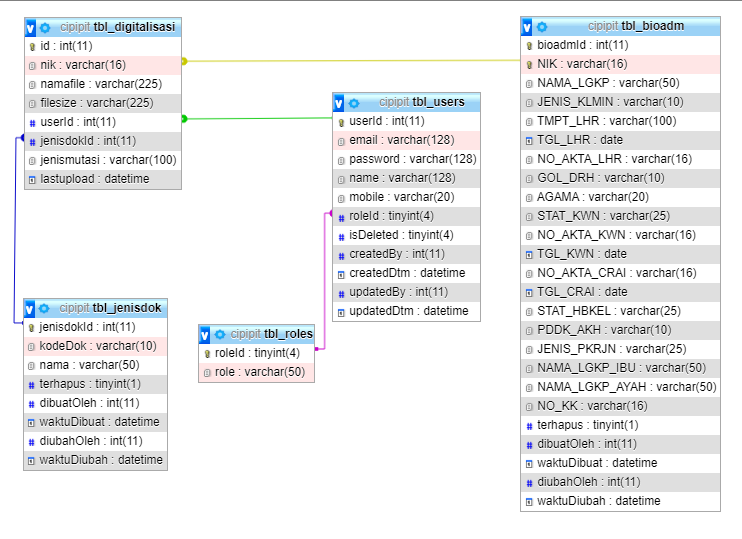
* + - 1. **Tabel User (tbl\_users)**



#### Gambar 4.5 Implementasi User (tbl\_users)

Gambar 4.5 merupakan implementasi dari tabel user (tbl\_users) yang digunakan untuk menyimpan data user.

### Relasi Antar Tabel



#### Gambar 4.6 Implementasi Relasi Tabel

Gambar 4.6 merupakan implementasi relasi antar tabel pada sistem bantu Digitalisasi Arsip Pindah Datang Di Dinas Kependudukan dan Pencatatan Sipil Kabupaten Bantul Berbasis Web.

### Fungsi Menambah Data Petugas

Fungsi berikut digunakan untuk menambahkan data user. Data user hanya dapat ditambahkan oleh admin yang merupakan super user. Fungsi ini digunakan pada halaman Users.

(Controller : user.php)

function addUser()

{

if($this->isAdmin() == TRUE)

{

$this->loadThis();

}

else

{

$this->load->model('user\_model');

$data['roles'] = $this->user\_model->getUserRoles();

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

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

}

}

#### Gambar 4.7 Fungsi Menampilkan Form Tambah User

Gambar 4.7 merupakan fungsi untuk menampilkan form tambah user. Untuk menambahkan data user baru admin harus mengisi terlebih dahulu form tambah user pada halaman Users.

(Application/controllers/user.php)

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\_clean|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('role','Role','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');

$roleId = $this->input->post('role');

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

$userInfo = array('email'=>$email, 'password'=>getHashedPassword($password), 'roleId'=>$roleId, '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 4.8 Fungsi Validasi dan Inisialisasi Input Data User

Gambar 4.8 merupakan fungsi untuk validasi data input dan inisialisasi data untuk dikirim ke database. Fungsi ini akan melakukan validasi input data dan inisialisasi data yang petugas yang telah diisikan. Jika data yang diisikan lengkap data akan tersimpan namun apabila data yang diisikan kurang valid maka data penyimpanan data tidak dapat dilanjutkan.

(Model : user\_model.php)

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 4.9 Fungsi Menambahkan Data User ke Database

Gambar 4.9 merupakan fungsi untuk menambahkan data user baru ke dalam database. Setelah data yang diisikan pada form tambah data sudah divalidasi dan diinisialisasi, selanjutnya data akan diproses simpan data dengan fungsi di atas.

### Fungsi Mengedit Data Petugas

Fungsi berikut merupakan fungsi yang digunakan untuk mengedit data petugas. Untuk mengedit data petugas menggunakan akses dari admin. Fungsi ini digunakan pada menu Users.

(Controller user.php)

function editOldUsr($userId = NULL)

{

if($this->isAdmin() == TRUE || $userId == 1)

{

$this->loadThis();

}

else

{

if($userId == null)

{

redirect('userListing');

}

$data['roles'] = $this->user\_model->getUserRoles();

$data['userInfo'] = $this->user\_model->getUserInfo($userId);

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

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

}

}

#### Gambar 4.10 Fungsi Menampilkan Form Ubah Data User

Gambar 4.10 merupakan fungsi untuk menampilkan form ubah data user lama. Fungsi ini hanya dapat diakses oleh admin pada halaman Users.

(application/controllers/user.php)

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\_clean|max\_length[128]');

$this->form\_validation->set\_rules('password','Password','matches[cpassword]|max\_length[20]');

$this->form\_validation->set\_rules('cpassword','Confirm Password','matches[password]|max\_length[20]');

$this->form\_validation->set\_rules('role','Role','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');

$roleId = $this->input->post('role');

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

$userInfo = array();

if(empty($password))

{

$userInfo = array('email'=>$email, 'roleId'=>$roleId, 'name'=>$name,

'mobile'=>$mobile, 'updatedBy'=>$this->vendorId, 'updatedDtm'=>date('Y-m-d H:i:s'));

}

else

{

$userInfo = array('email'=>$email, 'password'=>getHashedPassword($password), 'roleId'=>$roleId,

'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 4.11 Fungsi Validasi Data Input dan Inisialisasi Data

Gambar 4.11 merupakan fungsi untuk validasi data input dan inisialisasi data untuk dikirim ke database. Fungsi ini digunakan pada akses admin di halaman Users.

(Models user\_model.php)

function editUser($userInfo, $userId)

{

$this->db->where('userId', $userId);

$this->db->update('tbl\_users', $userInfo);

return TRUE;

}

#### Gambar 4.12 Fungsi Menyimpan Data Input ke Database

Gambar 4.12 merupakan fungsi untuk menyimpan data input user ke database. Data yang sudah berhasil divalidasi dan diinisialisasi kemudian disimpan ke dalam database menggunakan fungsi di atas.

### Fungsi Menghapus Data Petugas

Fungsi berikut merupanan fungsi yang digunakan untuk menghapus data user. Hapus data users hanya dapat dilakukan oleh admin. Fungsi hapus ini hanya mengubah status users dari aktif (status 0) menjadi terhapus (status 1)

(Controllers user.php)

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 4.13 Fungsi Untuk Menghapus Data User

Gambar 4.13 merupakan fungsi untuk menghapus data users terpilih. Fungsi ini digunakan pada halaman users dan hanya dapat diakses oleh admin.

(Models user\_model.php)

function deleteUser($userId, $userInfo)

{

$this->db->where('userId', $userId);

$this->db->update('tbl\_users', $userInfo);

return $this->db->affected\_rows();

}

#### Gambar 4.14 Fungsi Untuk Menghapus Data User di Database

Gambar 4.14 merupakan fungsi untuk menghapus data user di database. Fungsi ini digunakan pada halaman users dan hanya dapat diakses oleh admin.

### Fungsi Melihat Laporan dan Mencetak Laporan

Fungsi berikut adalah fungsi yang digunakan untuk melihat laporan dan mencetak laporan. Laporan dapat dilihat berdasarkan jenis pindah atau disajikan semua data pindah datang yang sudah dilakukan digitalisasi arsipnya. Untuk fungsi ini dapat diakses oleh admin dan petugas.

(Models BioAdm\_model.php)

function DaftarBioAdmJoinDigital($searchText = '', $dt1 , $dt2, $jenismutasi, $page, $segment)

{

$this->db->select('\*');

$this->db->from('tbl\_bioadm as bio');

$this->db->join('tbl\_digitalisasi as dgt', 'dgt.nik = bio.NIK');

$this->db->group\_by("dgt.nik");

if(!empty($searchText)) {

$likeCriteria = "(bio.NIK LIKE '%".$searchText."%' or bio.NAMA\_LGKP LIKE '%".$searchText."%' or bio.TMPT\_LHR LIKE '%".$searchText."%')";

$this->db->where($likeCriteria);

}

if(!empty($dt1) && !empty($dt2)){

$dt1 = date('Y-m-d',strtotime($dt1));

$dt2 = date('Y-m-d',strtotime($dt2));

$likeCriteria = "dgt.lastupload between '".$dt1."' and '".$dt2."' ";

$this->db->where($likeCriteria);

}

if(!empty($jenismutasi)){

$likeCriteria = "dgt.jenismutasi='".$jenismutasi."'";

$this->db->where($likeCriteria);

}

$this->db->where('bio.terhapus', 0);

$this->db->limit($page, $segment);

$query = $this->db->get();

$result = $query->result();

return $result;

}

#### Gambar 4. 15 Fngsi Untuk Mengambil Data Laporan

Gambar 4.15 merupakan fungsi untuk mengambil data laporan dari database. Fungsi ini akan menampilkan laporan berdasarkan tanggal dan jenis pindah.

(Controllers Laporan.php)

function lapcetakpdf(){

//2 (tglawal) 3 (tglakhir) 4 (searchText)

$this->load->model('laporan\_model');

$tgl1 = $tgl2 = $searchText = '';

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')!='-') $jmutasi = $this->uri->segment('4');

else $jmutasi = '';

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

else $searchText = '';

$pdf = new FPDF('l','mm','A5');

$pdf->AddPage();

$pdf->SetFont('Arial','B',16);

$pdf->Cell(190,7,'LAPORAN MUTASI',0,1,'C');

$pdf->SetFont('Arial','B',12);

if($tgl1!='' && $tgl2!=''){

$pdf->Cell(190,6,'Tanggal '.$tgl1.' s/d '.$tgl2,0,1,'C');

}

if($jmutasi!=''){

$pdf->Cell(190,6,'Jenis Mutasi : '.ucwords($jmutasi),0,1,'');

}

$pdf->Cell(10,7,'',0,1);

$pdf->SetFont('Arial','B',10);

$pdf->Cell(10,6,'No.',1,0);

$pdf->Cell(35,6,'NIK',1,0);

$pdf->Cell(45,6,'Nama',1,0);

$pdf->Cell(35,6,'Jenis Kelamin',1,0);

$pdf->Cell(25,6,'Tempat Lahir',1,0);

$pdf->Cell(25,6,'Tanggal Lahir',1,1);

$pdf->SetFont('Arial','',10);

$laporanData = $this->laporan\_model->cetakBasedPost($searchText,$tgl1,$tgl2,$jmutasi);

$i=1;

foreach ($laporanData as $row){

if($row->JENIS\_KLMIN=='L') $jk = 'Laki-Laki'; else if($row->JENIS\_KLMIN=='P') $jk = 'Perempuan';

if($row->TGL\_LHR!='0000-00-00') $dtx= date('d-m-Y',strtotime($row->TGL\_LHR)); else $dtx= '-';

$pdf->Cell(10,6,$i,1,0);

$pdf->Cell(35,6,$row->NIK,1,0);

$pdf->Cell(45,6,$row->NAMA\_LGKP,1,0);

$pdf->Cell(35,6,$jk,1,0);

$pdf->Cell(25,6,$row->TMPT\_LHR,1,0);

$pdf->Cell(25,6,$dtx,1,1);

$i++;

}

$pdf->Output();

}

#### Gambar 4.16 Fungsi Untuk Mengatur Tampilan Laporan

Gambar 4.16 merupakan fungsi untuk mengatur tampilan laporan, kemudian laporan akan ditampilkan dalam bentuk pdf.

(Models laporan\_model.php)

function cetakBasedPost($searchText = '', $dt1 , $dt2, $jenismutasi)

{

$this->db->select('\*');

$this->db->from('tbl\_bioadm as bio');

$this->db->join('tbl\_digitalisasi as dgt', 'dgt.nik = bio.NIK');

$this->db->group\_by("dgt.nik");

if(!empty($searchText)) {

$likeCriteria = "(bio.NIK LIKE '%".$searchText."%' or bio.NAMA\_LGKP LIKE '%".$searchText."%' or bio.TMPT\_LHR LIKE '%".$searchText."%')";

$this->db->where($likeCriteria);

}

if(!empty($dt1) && !empty($dt2)){

$dt1 = date('Y-m-d',strtotime($dt1));

$dt2 = date('Y-m-d',strtotime($dt2));

$likeCriteria = "dgt.lastupload between '".$dt1."' and '".$dt2."' ";

$this->db->where($likeCriteria);

}

if(!empty($jenismutasi)){

$likeCriteria = "dgt.jenismutasi='".$jenismutasi."'";

$this->db->where($likeCriteria);

}

$this->db->where('bio.terhapus', 0);

$query = $this->db->get();

$result = $query->result();

return $result;

}

#### Gambar 4.17 Fungsi Untuk Mengambil Data Dari Database Untuk Fungsi Cetak

Gambar 4.17 merupakan fungsi untuk mengambil data dari database untuk fungsi cetak.

### Fungsi Menambah Jenis Dokumen

Fungsi berikut digunakan untuk menambahkan data jenis dokumen. Data jenis dokumen merupakan data yang digunakan untuk mengupload dokumen kelengkapan berkas pindah datang. Data jenis dokumen dapat ditambah, diedit dan dihapus sesuai kebutuhan dari kelengkapan berkas.

(Controllers JenisDok.php)

/\*\*

\* Fungsi ini digunakan untuk menampilkan form tambah jenis dokumen

\*/

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 4.18 Fungsi Untuk Menampilkan Form Tambah Jenis Dokumen

Gambar 4.18 merupakan fungsi untuk menampilkan form tambah jenis. Fungsi tersebut akan menampilkan form untuk menambahkan jenis dokumen yang digunakan untuk mengupload dokumen.

function addNewJenisDok()

{

if($this->isForAll() == TRUE)

{

$this->loadThis();

}

else

{

$this->load->library('form\_validation');

$this->form\_validation->set\_rules('kodeDok','Kode','trim|required|max\_length[10]|xss\_clean');

$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('kodeDok');

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

$jenisdokInfo = array('kodeDok'=>$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 4.19 Fungsi Untuk Validasi Data Input Dan Inisialisasi Data Untuk Dikirim ke Database

(Models jenisdok\_model.php)

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 4.20 Fungsi Untuk Menambahkan Jenis Dokumen di Database

Gambar 4.20 merupakan Fungsi untuk validasi data input dan inisialisasi data untuk dikirim ke database

### Fungsi Mengedit Jenis Dokumen

Fungsi berikut digunakan untuk mengedit jenis dokumen. Fungsi ini digunakan pada halaman menu Jenis Dokumen.

(Controllers JenisDok.php)

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 Jenisd Dokumen';

$this->loadViews("jenisdok/editOldJdk", $this->global, $data, NULL);

}

}

#### Gambar 4.21 Fungsi Untuk Menampilkan Form Edit dan Data Jenis Dokumen Lama

Gambar 4.21 merupakan fungsi untuk menampilkan form edit jenis dokumen dan menampilkan informasi jenis dokumen lama.

(Controllers JenisDok.php)

function editJenisDok()

{

if($this->isForAll() == TRUE)

{

$this->loadThis();

}

else

{

$this->load->library('form\_validation');

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

$this->form\_validation->set\_rules('kodeDok','Kode','trim|required|max\_length[10]|xss\_clean');

$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('kodeDok');

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

$jenisdokInfo = array('kodeDok'=>$kode, 'nama'=>$nama, 'diubahOleh'=>$this->vendorId, '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 4.22 Fungsi Untuk Validasi Data dan Inisialisasi Data

Gambar 4.22merupakan Fungsi untuk validasi data jenis dokumen yang terlah diinput dan inisialisasi data untuk dikirim ke database. Jika sudah lengkap maka akan dapat dikirim ke database untuk disimpan, jika gagal akan menampilkan pesan gagal simpan.

(Models jenisdok\_model)

function editJenisDok($jenisdokInfo, $jenisdokId)

{

$this->db->where('jenisdokId', $jenisdokId);

$this->db->update('tbl\_jenisdok', $jenisdokInfo);

return TRUE;

}

#### Gambar 4.23 Fungsi Untuk Menyimpan Data Perubahan Jenis Dokumen

Gambar 4.23 merupakan fungsi untuk menyimpan data perubahan jenis dokumen ke dalam database. Setelah selesai berhasil divalidasi dan diinisialisasi data akan dikirim ke database untuk disimpan.

### Fungsi Menghapus Jenis Dokumen

Fungsi berikut merupakan fungsi yang digunakan untuk menhapus jenis dokumen dimana fungsi ini digunakan pada halaman menu jenis dokumen. Status jenis dokumen yang aktif adalah berstatus nol (status 0) dan untuk jenis dokumen yang sudah dihapus akan berstatus satu (status 1).

(Controllers JenisDok.php)

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 4.24 Fungsi Untuk Menghapus Jenis Dokumen

Gambar 4.24 merupakan fungsi untuk menghapus jenis dokumen pada halaman menu Jenis Dokumen.

(Models jenisdok\_model.php)

function deleteJenisDok($jenisdokId, $jenisdokInfo)

{

$this->db->where('jenisdokId', $jenisdokId);

$this->db->update('tbl\_jenisdok', $jenisdokInfo);

return $this->db->affected\_rows();

}

#### Gambar 4.25 Fungsi Untuk Menghapus Jenis Dokumen di Database

Gambar 4.25 merupakan fungsi yang digunakan untuk menghapus jenis dokumen pada database

### Fungsi Menampilkan Biodata Adminduk

Fungsi berikut merupakan fungsi yang digunakan untuk menampilkan biodata adminduk. Fungsi ini digunakan pada halaman menu Biodata Adminduk. Biodata adminduk merupakan data penduduk Kabupaten Bantul yang berisi informasi tentang data kependudukan.

(Controllers BioAdm.php)

function DaftarBioAdm()

{

if($this->isForAll() == TRUE)

{

$this->loadThis();

}

else

{

$this->load->model('bioadm\_model');

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

$data['searchText'] = $searchText;

$this->load->library('pagination');

$count = $this->bioadm\_model->DaftarBioAdmCount($searchText);

$returns = $this->paginationCompress ( "DaftarBioAdm/", $count, 5 );

$data['bioadmRecords'] = $this->bioadm\_model->DaftarBioAdm($searchText, $returns["page"], $returns["segment"]);

$this->global['pageTitle'] = 'Adminduk : Daftar Biodata Adminduk';

$this->loadViews("bioadm/viewList", $this->global, $data, NULL);

}

}

#### Gambar 4.26 Fungsi Untuk Menampilkan Daftar Biodata Adminduk

Gambar 4.26 merupakan fungsi untuk menampilkan daftar biodata adminduk. Fungsi ini akan menampilkan halaman Biodata Adminduk

(Models bioadm\_model.php)

function DaftarBioAdm($searchText = '', $page, $segment)

{

$this->db->select('\*');

$this->db->from('tbl\_bioadm');

if(!empty($searchText)) {

$likeCriteria = "(NIK LIKE '%".$searchText."%'

OR NAMA\_LGKP LIKE '%".$searchText."%'

OR JENIS\_KLMIN LIKE '%".$searchText."%'

OR TMPT\_LHR LIKE '%".$searchText."%'

OR NO\_AKTA\_LHR LIKE '%".$searchText."%'

OR AGAMA LIKE '%".$searchText."%'

OR STAT\_KWN LIKE '%".$searchText."%'

OR NO\_AKTA\_KWN LIKE '%".$searchText."%'

OR NO\_AKTA\_CRAI LIKE '%".$searchText."%'

OR STAT\_HBKEL LIKE '%".$searchText."%'

OR JENIS\_PKRJN LIKE '%".$searchText."%'

OR NAMA\_LGKP\_IBU LIKE '%".$searchText."%'

OR NAMA\_LGKP\_AYAH LIKE '%".$searchText."%'

OR NO\_KK LIKE '%".$searchText."%'

)";

$this->db->where($likeCriteria);

}

$this->db->where('terhapus', 0);

$this->db->limit($page, $segment);

$query = $this->db->get();

$result = $query->result();

return $result;

}

#### Gambar 4.27 Fungsi Mengambil Data Dari Database

Gambar 4.27 merupakan fungsi mengambil data dari database untuk ditampilkan. Fungsi ini mengambil data dari database untuk ditampilkan pada halaman Biodata Adminduk.

### Fungsi Menampilkan Detail Biodata Adminduk

Fungsi berikut merupakan fungsi yang digunakan untuk menampilkan detail biodata adminduk pada halaman menu Biodata Adminduk. Data detail biodata adminduk akan ditampilkan pada halaman Biodata Adminduk, data tidak dapat diubah, ditambah maupun dihapus.

(Controllers BioAdm.php)

function detailBam($bioadmId = NULL)

{

if($this->isForAll() == TRUE)

{

$this->loadThis();

}

else

{

if($bioadmId == null)

{

redirect('DaftarBioAdm');

}

$data['bioadmInfo'] = $this->bioadm\_model->getBioAdmInfo($bioadmId);

$this->global['pageTitle'] = 'Adminduk : Detail Biodata Adminduk';

$this->loadViews("bioadm/detailBam", $this->global, $data, NULL);

}

}

#### Gambar 4.28 Fungsi Untuk Informasi Detail Biodata

Gambar 4.28 merupakan fungsi yang digunakan untuk informasi detail biodata adminduk.

(Models bioadm\_model)

function getBioAdmInfo($bioadmId)

{

$this->db->select('\*');

$this->db->from('tbl\_bioadm');

$this->db->where('bioadmId', $bioadmId);

$query = $this->db->get();

return $query->result();

}

#### Gambar 4.29 Fungsi Mengambil Data Dari Database

Gambar 4.29 merupakan fungsi yang digunakan untuk mengambil data dari database dan selanjutnya data tersebut ditampilkan pada halaman menu Biodata Adminduk.

### Fungsi Upload Dokumen Pindah Datang

Fungsi berikut merupakan fungsi yang digunakan untuk mengupload dokumen pindah datang. Fungsi ini digunakan pada halaman menu Mutasi Datang dan Mutasi Keluar.

(Controlles Upload.php)

public function set\_upload\_options($nik,$nmfile,$type){

$config = array();

$nmfile = str\_replace(' ', '\_', $nmfile);

if(!is\_dir('./dokumen')) mkdir('./dokumen');

if(!is\_dir('./dokumen/mutasi')) mkdir('./dokumen/mutasi');

if(!is\_dir('./dokumen/mutasi/'.$type)) mkdir('./dokumen/mutasi/'.$type);

if(!is\_dir('./dokumen/mutasi/'.$type.'/'.$nik)) mkdir('./dokumen/mutasi/'.$type.'/'.$nik);

$config['upload\_path'] = './dokumen/mutasi/'.$type.'/'.$nik;

$config['allowed\_types'] = 'pdf';

$config['max\_size'] = 0;

$config['max\_width'] = 0;

$config['max\_height'] = 0;

$config['file\_name'] = $nmfile;

$arrfile = array();

if(is\_dir('./dokumen/mutasi/'.$type.'/'.$nik)){

if($handle = opendir('./dokumen/mutasi/'.$type.'/'.$nik))

{

while(($file = readdir($handle)) !== false)

{

if($file != '.' && $file != '..')

{

$arrfile[] = $file;

}

}

closedir($handle);

}

}

foreach($arrfile as $k=>$v){

if(preg\_match("/".$nmfile."/i", $v)){

if(file\_exists('./dokumen/mutasi/'.$type.'/'.$nik.'/'.$v)) unlink('./dokumen/mutasi/'.$type.'/'.$nik.'/'.$v);

}

}

return $config;

}

public function aksi\_upload(){

date\_default\_timezone\_set("Asia/Jakarta");

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

$type = $this->uri->segment('2');

$this->load->model('upload\_model');

$this->load->model('jenisdok\_model');

$this->load->model('config\_model');

$this->global['pageTitle'] = 'Adminduk : Upload Dokumen';

$this->load->model('bioadm\_model');

$databioadm = $this->bioadm\_model->getBioAdmInfoByNIK($nik);

$dataconfig = $this->config\_model->getConfigInfo();

$datajenis = $this->jenisdok\_model->ListJenisDok();

$valx = array();

foreach($dataconfig as $k=>$v){

if($type=='datang'){

if($v->conf\_key=='mutasidatang'){

$valx[] = $v->conf\_val;

}

}

if($type=='keluar'){

if($v->conf\_key=='mutasikeluar'){

$valx[] = $v->conf\_val;

}

}

}

$newvalx = json\_decode($valx[0],true);

$this->load->library('upload');

$files = $\_FILES;

$error= array();

if(is\_array($files) && count($files)>0){

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

$errx = array();

foreach($newvalx as $k=>$v){

if($files['berkas']['name'][$k]!=''){

if($v==1){

$errx[$k] = 0;

}

else{

$errx[$k] = 0;

}

}

else{

if($v==1){

$errx[$k] = 1;

}

else{

$errx[$k] = 0;

}

}

}

if(in\_array('1',$errx)){

?>

<script>

alert('Maaf, mohon perhatikan wajib isi pada setiap field');

</script>

<?php

}

else{

foreach($\_FILES['berkas']['name'] as $k=>$v)

{

$\_FILES['berkas']['name']= $files['berkas']['name'][$k];

$\_FILES['berkas']['type']= $files['berkas']['type'][$k];

$\_FILES['berkas']['tmp\_name']= $files['berkas']['tmp\_name'][$k];

$\_FILES['berkas']['error']= $files['berkas']['error'][$k];

$\_FILES['berkas']['size']= $files['berkas']['size'][$k];

// echo $\_FILES['berkas']['name'].'<br>';

if($\_FILES['berkas']['name']!=''){

//$this->upload->deletedatafile($databioadm[0]->NIK,$\_FILES['berkas']['name']);

$dr\_[$k]['filename'] = $nmfile[$k];

$dr\_[$k]['filesize'] = $\_FILES['berkas']['size'];

$dr\_[$k]['jenisdokId'] = $k;

$this->upload->initialize($this->set\_upload\_options($databioadm[0]->NIK,$nmfile[$k],$type));

$tes = $this->upload->do\_upload('berkas');

if(!$tes){

//echo $\_FILES['berkas']['name'].'no upload <br>';

$error[] = $this->upload->display\_errors();

}

}

}

$lct = './dokumen/mutasi/'.$type.'/'.$nik;

if(!file\_exists($lct.'/waktu.txt')){

$filex = fopen($lct."/waktu.txt","w");

fwrite($filex,strtotime(date('d-m-Y H:i:s')));

fclose($filex);

}

$this->upload\_model->FileDBproses($databioadm[0]->NIK,$dr\_,$type,$this->vendorId);

?>

<script>

alert('Berhasil Upload Data !');

</script>

<?php

}

}

$data['config'] = $dataconfig;

$data['jenis'] = $datajenis;

$data['error'] = $error;

$data['bioadm'] = $databioadm;

$data['type'] = $type;

$this->loadViews("upload/v\_upload", $this->global, $data, NULL);

}

#### Gambar 4.30 Fungsi Menampilkan Form dan Mengupload Dokumen

Gambar 4.30 merupakan fungsi untuk menampilkan form upload dan untuk mengupload dokumen ke server.

(Models upload\_model.php)

function FileDBproses($nik,$data,$type,$user)

{

date\_default\_timezone\_set("Asia/Jakarta");

foreach($data as $k=>$v){

$this->db->where('nik', $nik);

$this->db->where('namafile', $v['filename']);

$this->db->delete('tbl\_digitalisasi');

$dt = array(

'nik' => $nik,

'namafile' => $v['filename'],

'filesize' => $v['filesize'],

'jenismutasi' => $type,

'jenisdokId' => $v['jenisdokId'],

'userId' => $user,

'lastupload' => date('Y-m-d H:i:s')

);

$this->db->insert('tbl\_digitalisasi', $dt);

}

return TRUE;

}

#### Gambar 4.31 Fungsi Meyimpan Informasi File Upload ke Database

Gambar 4.31 merupakan fungsi untuk meyimpan informasi file upload ke database. Fungsi ini digunakan pada halaman menu Mutasi Datang dan Mutasi Keluar.

### Koneksi Basis Data

Fungsi berikut merupakan fungsi yang digunakan untuk mengkoneksikan dengan database yang akan digunakan.

$active\_group = 'default';

$active\_record = TRUE;

$db['default']['hostname'] = 'localhost';

$db['default']['username'] = 'root';

$db['default']['password'] = '';

$db['default']['database'] = 'cipipit';

$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 4.32 Koneksi Database

## Pembahasan Sistem

Sistem bantu digitalisasi arsip ini memiliki 2 (dua) bagian sistem, untuk sistem yang pertama adalah admin dan yang kedua adalah petugas. Untuk hak akses admin dapat melakukan seluruh rangkaian proses. Sedangkan untuk hak akses petugas terbatas.

### Halaman Login



#### Gambar 4.34 Tampilan Halaman Login

Gambar 4.34 merupakan tampilan halaman login, baik yang digunakan oleh admin maupun petugas tampilan halaman loginnya sama.

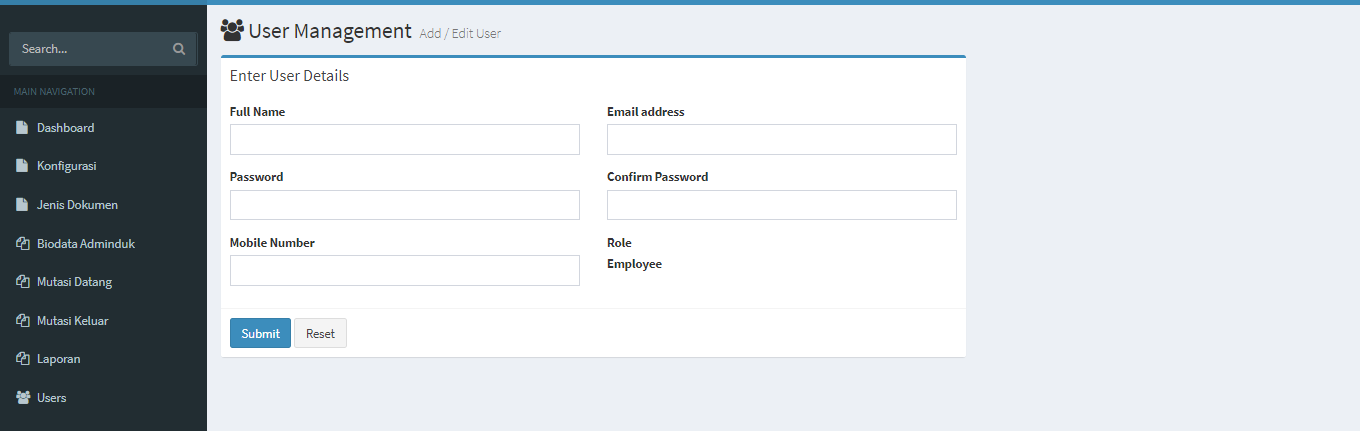
### Halaman Utama Admin



#### Gambar 4.35 Tampilan Halaman Utama Admin

Gambar 4.35 merupakan tampilan halaman utama admin dengan login admin dan memiliki hak akses penuh atas sistem. Admin memiliki beberapa menu yang dapat diakses, antara lain menu jenis dokumen, biodata adminduk, mutase datang, muasi keluar, dan users.

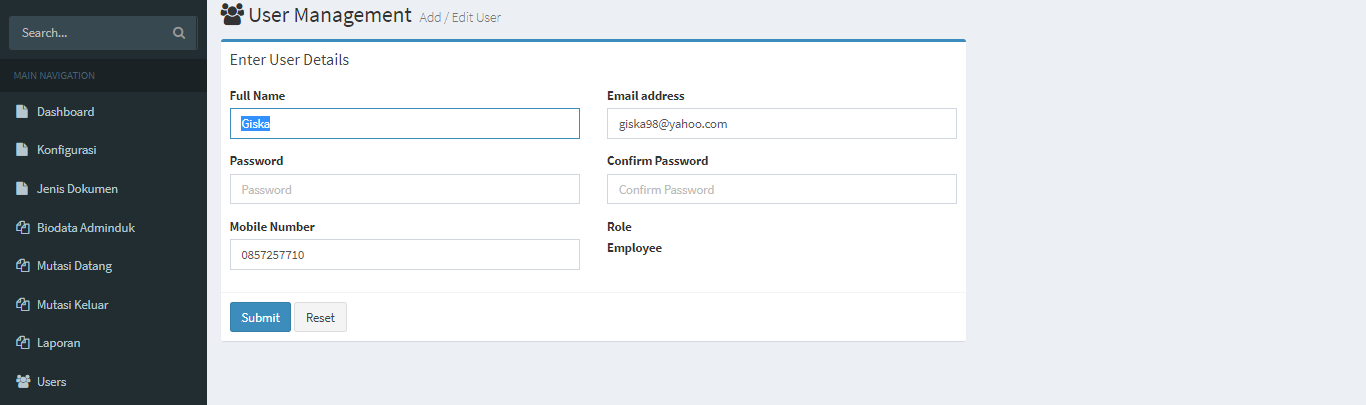
### Halaman Tambah User



#### Gambar 4.36 Tampilan Halaman Tambah User

Gambar 4.36 merupakan tampilan halaman tambah user pada hak akses admin. Untuk menambahkan data user hanya dapat dilakukan oleh admin dengan login admin, kemudian mengisikan data user pada form tambah user. Untuk role tidak dapat diubah karena pada sistem bantu ini admin hanya 1 (satu).

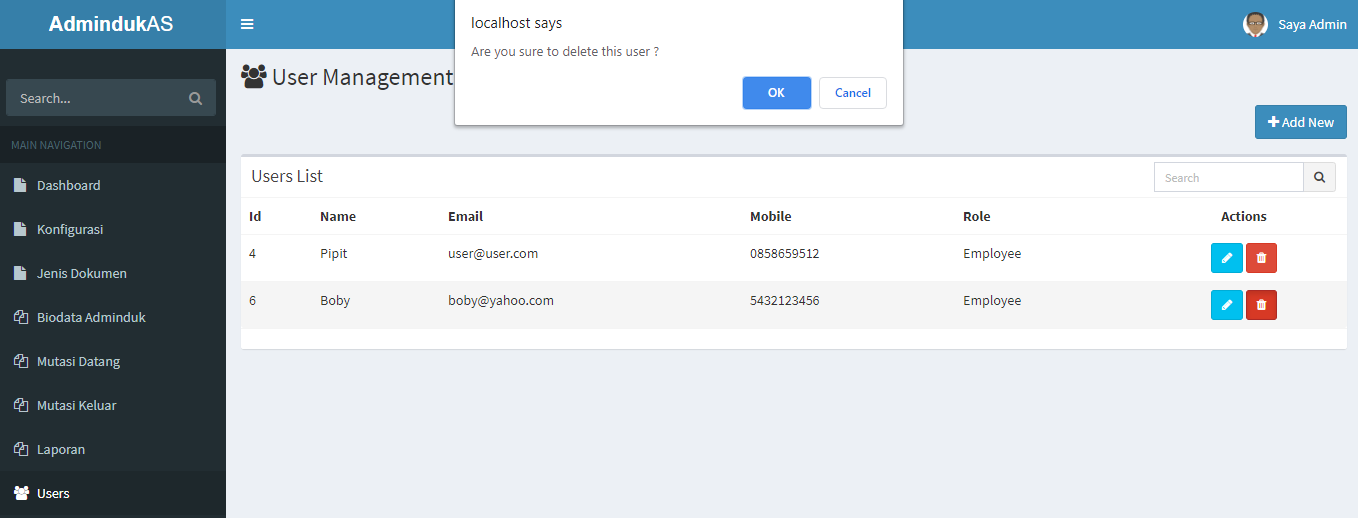
### Halaman Edit User



#### Gambar 4.37 Tampilan Halaman Edit User

Gambar 4.37 merupakan tampilan halaman edit user, dimana user yang sudah ada diedit datanya sesuai dengan data yang diinginkan. Setelah selesai melakukan edit user klik tombol submit untuk menyimpan.

### Halaman Hapus User



#### Gambar 4.38 Tampilan Halaman Hapus User

Gambar 4.38 merupakan tampilan halaman hapus user. Untuk menghapus user harus dengan hak akses sebagai admin. Sebelum menghapus user terlebih dahulu masuk pada halaman user kemudian memilih user yang akan dihapus dan klik tombol sampah atau hapus, maka akan tampil pesan untuk menghapus user kemudian klik ok.

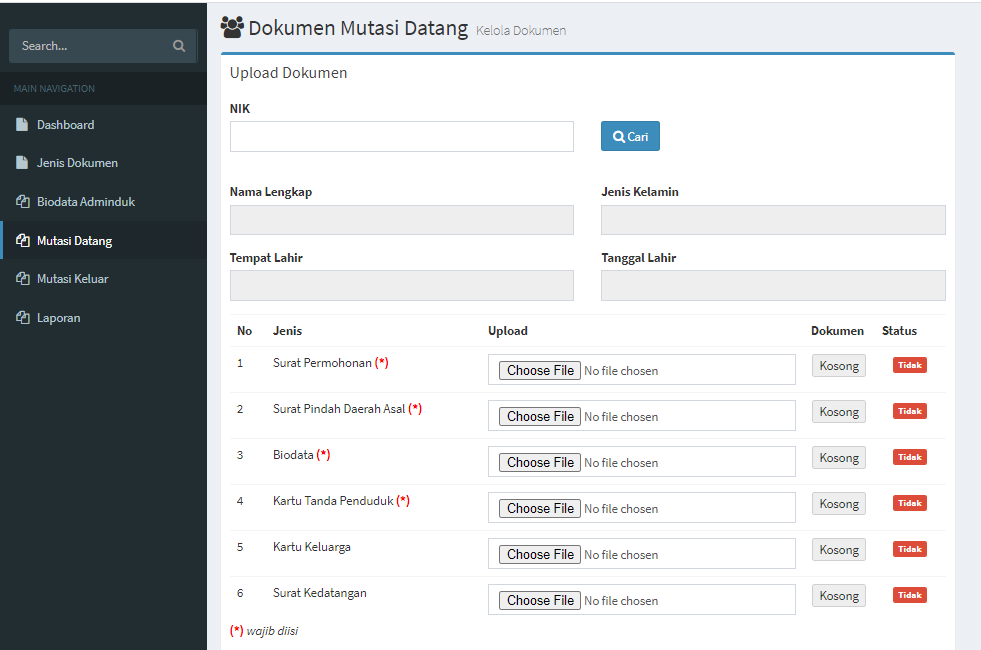
### Halaman Utama User



#### Gambar 4.39 Tampilan Halaman Utama User (Petugas)

Gambar 4.39 merupakan tampilan halaman utama user (petugas). Hak akses user (petugas) terdapat 5 (lima) menu yang dapat diakses, yaitu menu jenis dokumen, biodata admonduk, mutase datang, mutasi keluar dan laporan.

### Halaman Upload Mutasi Datang

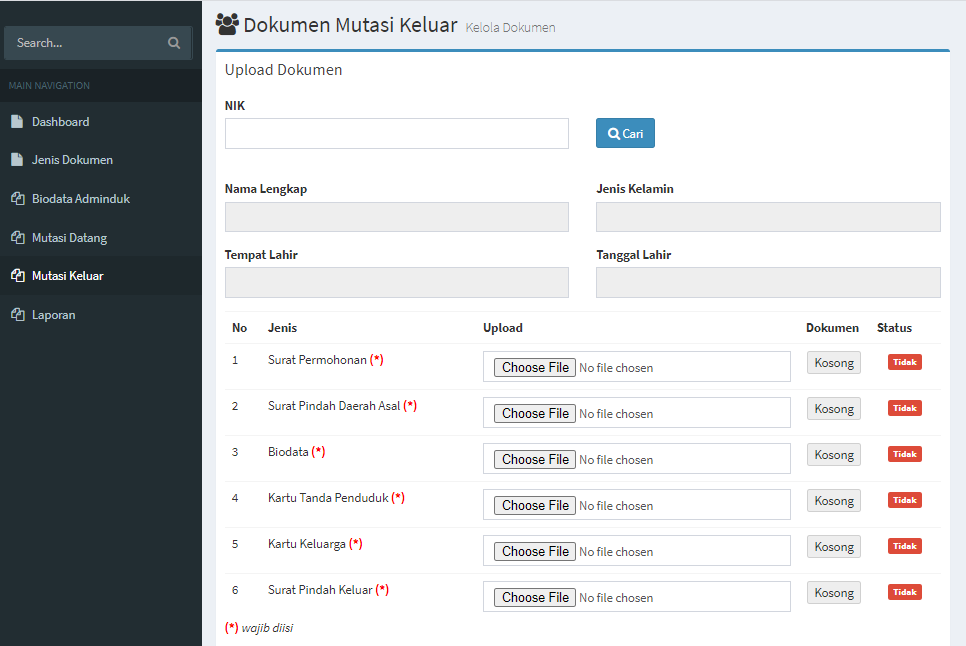


#### Gambar 4.39 Tampilan Halaman Upload Mutasi Dtang

Gambar 4.39 merupakan tampilan halaman upload mutasi datang. Pada menu ini dapat diakses pada login admin maupun user (petugas). Menu ini digunakan untuk mengupload berkas atau dokumen mutasi datang. Pada tampilan menu ini petugas menginputkan NIK terlebih dahulu kemudian klik tombol cari untuk menampilkan datanya. Petugas memastikan data tersebut benar kemudian langkah berikutnya adalah mengupload dokumen pada menu upload yang sudah disediakan.

Upload dokumen sudah dibedakan berdasarkan jenis dokumen sehingga dokumen tidak akan tertukar dan dipastikan lengkap dokumen tersebut. Setelah upload berkas selesai, data tersebut akan ditandai sudah dilakukan digitalisasi arsip untuk mutasi datang.

### Halaman Upload Mutasi Keluar

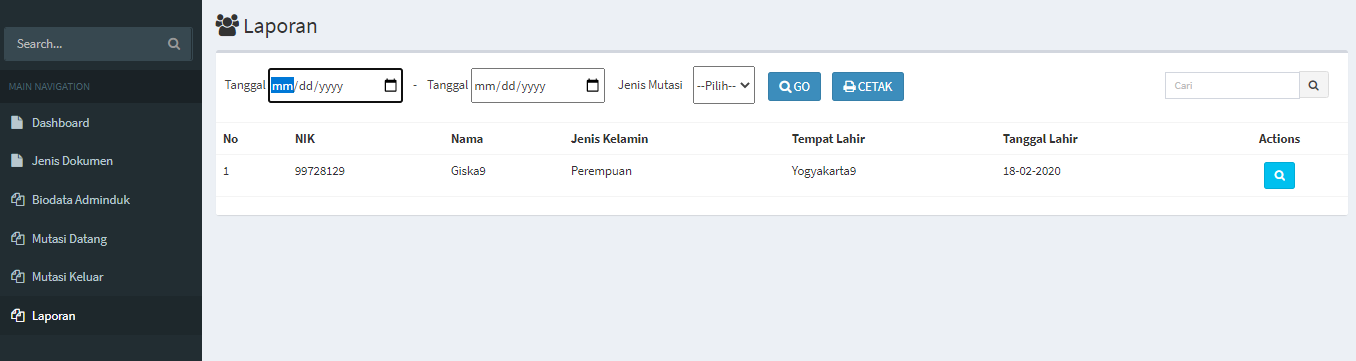


#### Gambar 4.40 Tampilan Halaman Upload Mutasi Keluar

Gambar 4.40 merupakan tampilan halaman upload mutasi keluar. Pada menu ini dapat diakses pada login admin maupun user (petugas). Menu ini digunakan untuk mengupload berkas atau dokumen mutasi keluar. Pada tampilan menu ini petugas menginputkan NIK terlebih dahulu kemudian klik tombol cari untuk menampilkan datanya. Petugas memastikan data tersebut benar kemudian langkah berikutnya adalah mengupload dokumen pada menu upload yang sudah disediakan.

Upload dokumen sudah dibedakan berdasarkan jenis dokumen sehingga dokumen tidak akan tertukar dan dipastikan lengkap dokumen tersebut. Setelah upload berkas selesai, data tersebut akan ditandai sudah dilakukan digitalisasi arsip untuk mutasi keluar.

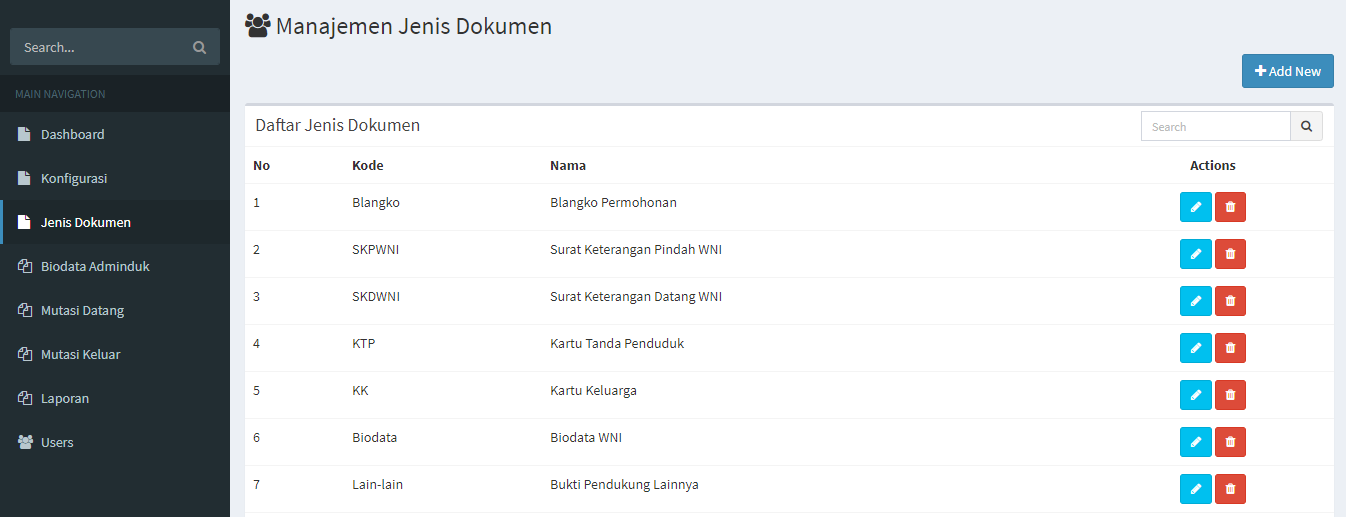
### Halaman Menu Laporan



#### Gambar 4.41 Tampilan Halaman Menu Laporan

Gambar 4.41 merupakan tampilan halaman menu laporan. Halaman menu laporan dapat diakses dengan login admin maupun user (petugas). Halaman ini akan menampilkan data penduduk yang sudah dilakukan digitalisasi arsipnya. Data yang disajikan oleh menu laporan dengan rentan 1 (satu) bulan. Tampilan laporan adalah berupa file pdf.

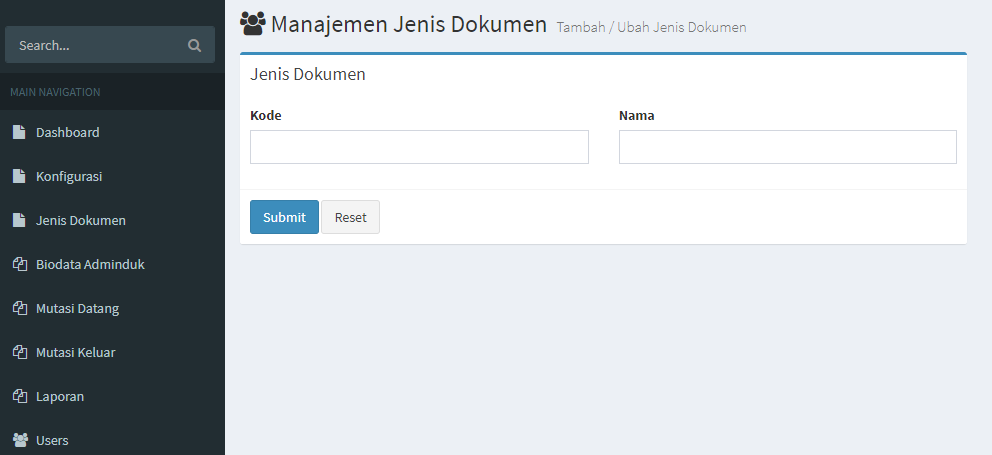
### Halaman Jenis Dokumen



#### Gambar 4.42 Tampilan Halaman Jenis Dokumen

Gambar 4.42 merupakan tampilan halaman jenis dokumen. Menu ini dapat diakses dengan login admin maupun login user (petugas). Kedua hak akses tersebut dapat melakukan aksi tambah, ubah dan hapus pada data jenis dokumen,

### Halaman Tambah Jenis Dokumen

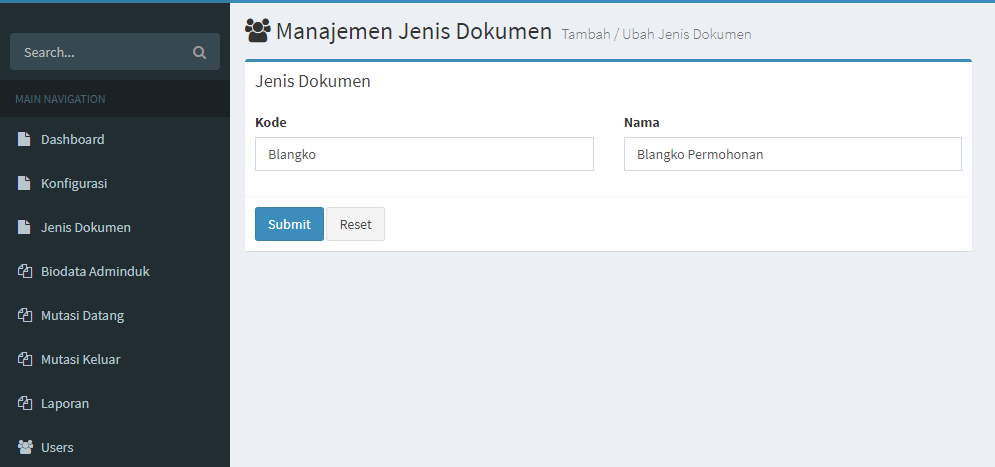


#### Gambar 4.43 Tampilan Halaman Tambah Jenis Dokumen

Gambar 4.43 merupakan tampilan halaman tambah jenis dokumen. Apabila ada tambahan jenis dokumen dalam proses upload dokumen pindah datang, admin maupun user (petugas) dapat melakukan tambah jenis dokumen. Jenis dokumen dapat ditambahkan sesuai dengan kebutuhan dari berkas permohonan pindah datang.

Untuk menambahkan jenis dokumen isikan kode dan nama dari jenis dokumen tersebut. Selanjutnya klik submit untuk menyimpan data jenis dokumen.

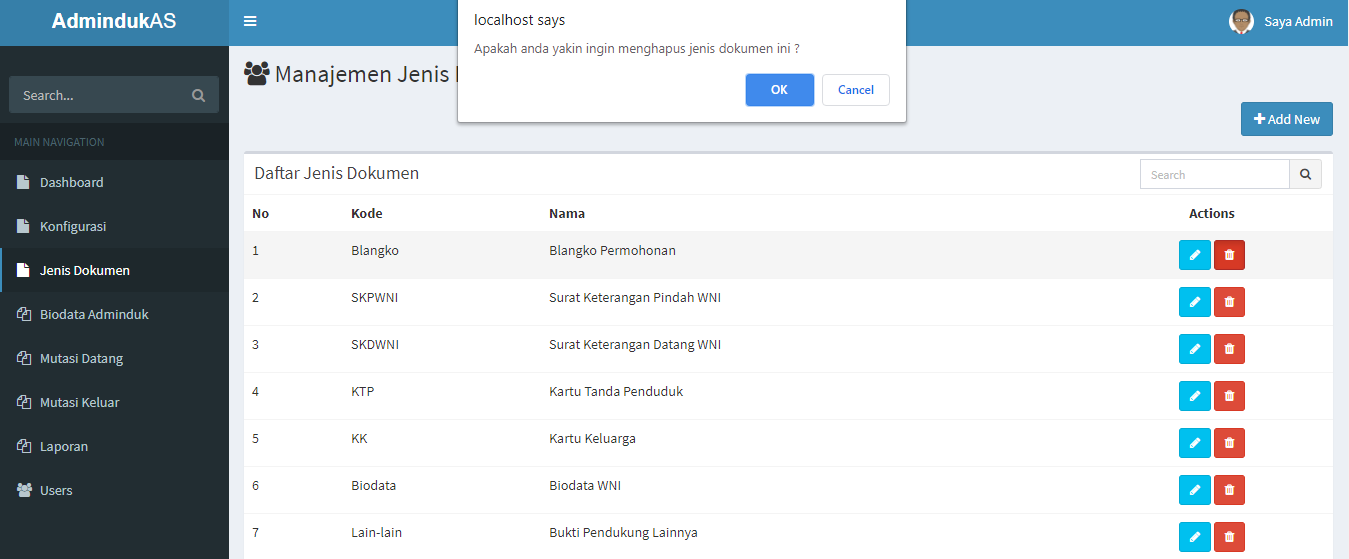
### Halaman Edit Jenis Dokumen



#### Gambar 4.44 Tampilan Halaman Edit Jenis Dokumen

Gambar 4.44 merupakan tampilan halaman edit jenid dokumen. Jika jenis dokumen dirasa kurang sesuai dengan dokumen pindah datang, maka jenis dokumen dapat diupdate sesuai dengan yang diinginkan. Admin maupun user (petugas) hanya tinggal mengubah kode dan nama sesuai data yang baru.

### Halaman Hapus Jenis Dokumen



#### Gambar 4.45 Tampilan Halaman Hapus Jenis Dokumen

Gambar 4.45 merupakan tampilan halaman hapus jenis dokumen. Apabila disasa jenis dokumen tersebut sudah tidak digunakan ataupun sudah tidak relevan dengan berkas pindah datang, maka jenis dokumen dapat dihapus oleh admin maupuan user (petugas). Sebelum menghapus data pastikan terlebih dahulu data jenis dokumen sudah tidak digunakan, kemudian klik tombol hapus. Maka, akan tampil pesan hapus data klik tombol ok dan data jenis dokumen sudah terhapus.