Sistem monografi kelurahan Dokumen Pengembangan Aplikasi

Pemerintah Kota Cimahi

2014

# **DAFTAR ISI**

[**DAFTAR ISI** 1](#_Toc405455758)

[**DAFTAR GAMBAR** 2](#_Toc405455759)

[**PENDAHULUAN** 3](#_Toc405455760)

[**REQUIREMENT SPECIFICATION** 4](#_Toc405455761)

[**SYSTEM NEEDS** 4](#_Toc405455762)

[**INTERFACE SPESIFICATION** 4](#_Toc405455763)

[**PERANCANGAN** 5](#_Toc405455764)

[**DIAGRAM KONTEKS** 6](#_Toc405455765)

[**DATA FLOW DIAGRAM** 7](#_Toc405455766)

[Proses Monografi Kelurahan 16](#_Toc405455767)

[Proses Kejadian Kelurahan 16](#_Toc405455768)

[Proses Laporan 17](#_Toc405455769)

[**DESAIN BASIS DATA** 18](#_Toc405455770)

[**IMPLEMENTASI** 19](#_Toc405455771)

[Implementasi User Interface 19](#_Toc405455772)

[**IMPLEMENTASI SISTEM** 26](#_Toc405455773)

[Kode Sumber (Source Code) 27](#_Toc405455774)

[**IMPLEMENTASI BASIS DATA** 31](#_Toc405455775)

[Data Definition Language 31](#_Toc405455776)

# **DAFTAR GAMBAR**

[Gambar 1. Arsitektur Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi 5](#_Toc405455861)

[Gambar 2. Application Stack 6](#_Toc405455862)

[Gambar 3. Diagram Konteks Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi 6](#_Toc405455863)

[Gambar 4. Diagram Flow Diagram Level 1 7](#_Toc405455864)

[Gambar 5. Diagram Flow Level 2 (Proses Monografi Kelurahan) 16](#_Toc405455865)

[Gambar 6. Diagram Flow Level 2 (Proses Kejadian Kelurahan) 17](#_Toc405455866)

[Gambar 7. Diagram Flow Level 2 (Proses Laporan Monografi Kelurahan) 17](#_Toc405455867)

[Gambar 8. Desain Basis Data 18](#_Toc405455868)

[Gambar 9. Halaman Login 19](#_Toc405455869)

[Gambar 10. Halaman Utama Aplikasi 19](#_Toc405455870)

[Gambar 11. Tampilan Data Monografi Kelurahan 20](#_Toc405455871)

[Gambar 12. Tampilan Ubah Data Kelurahan 20](#_Toc405455872)

[Gambar 13. Detail Data Monografi Kelurahan Melong 21](#_Toc405455873)

[Gambar 14. Tambah Data Umum Kelurahan 21](#_Toc405455874)

[Gambar 15. Tambah Data Personil Kelurahan 22](#_Toc405455875)

[Gambar 16. Tambah Data Kewenangan Kelurahan 23](#_Toc405455876)

[Gambar 17. Tambah Data Keuangan Kelurahan 23](#_Toc405455877)

[Gambar 18. Tambah Data Kelembagaan Kelurahan 24](#_Toc405455878)

[Gambar 19. Tambah Data Ketentraman dan Ketertiban 24](#_Toc405455879)

[Gambar 20. Tampilan Data Kejadian Kelurahan 25](#_Toc405455880)

[Gambar 21. Tambah Data Kejadian Kelurahan 25](#_Toc405455881)

[Gambar 22. Ubah Data Kejadian Kelurahan 26](#_Toc405455882)

[Gambar 23. Tampilan Menu Laporan Monografi Kelurahan 26](#_Toc405455883)

# **PENDAHULUAN**

Aplikasi Monografi Kelurahan adalah aplikasi yang memantau rincian data dan statistik pemerintahan, sumber daya alam, sumber daya manusia, ekonomi, pendidikan, dan kondisi geografis dari suatu wilayah. Adanya data monografi, kita dapat melihat gambaran dari situasi dan kondisi kelurahan tertentu. Manfaat Monografi Kelurahan adalah untuk mempermudah para pihak yang memerlukan data dari suatu kelurahan.

Mengacu kepada *requirement specification* yang ada, aplikasi Monografi Kelurahan Pemerintah Kota Cimahi diharapkan dapat diakses dengan *web browser* baik dari komputer desktop/laptop maupun dari smartphone serta diharapkan pula dapat diakses secara bersama-sama oleh beberapa pengguna sekaligus yang telah terdaftar melalui jaringan internal di lingkungan kerja Pemerintah Kota Cimahi. Pengguna dapat didaftarkan sebagai pengguna/pengakses aplikasi oleh pengelola aplikasi Monografi Kelurahan.

Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi di desain sedemikian rupa sehingga memudahkan bagi pengguna aplikasi dalam melihat tampilan data dan laporan yang ada.

Fleksibilitas yang ditawarkan aplikasi Monografi Kelurahan Pemerintah Kota Cimahi bukan merupakan fleksibilitas yang tanpa batas. Ada beberapa batasan penggunaan aplikasi terutama terkait dengan performansi yang dapat ditangani oleh server aplikasi utama maupun server database sebagai server penyedia dan penyimpanan data yang diakses oleh aplikasi.

Dokumen ini disusun sebagai sebuah dokumen yang diharapkan dapat memberikan gambaran secara menyeluruh tentang aplikasi Monografi Kelurahan Kota Cimahi secara teknikal, baik dari sisi aplikasi maupun dari sisi sistem basis data yang digunakan.

Penjelasan aplikasi Monografi Kelurahan Pemerintah Kota Cimahi dimulai dari pemaparan daftar kebutuhan (*requirements specification*) sebagai dasar pembuatan aplikasi, kemudian dilanjutkan pada proses analisa kebutuhan, proses desain sistem aplikasi, hingga implementasi aplikasi baik implementasi dari sisi *user interface*, maupun implementasi basis datanya.

# **REQUIREMENT SPECIFICATION**

Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi dibangun berdasar kepada daftar kebutuhan yang telah ditetapkan oleh Bagian Pemerintahan Kota Cimahi. Secara umum, spesifikasi kebutuhan aplikasi Monografi Kelurahan Kota Cimahi dapat dibagi menjadi tiga kebutuhan, yaitu

1. Kebutuhan data kelurahan Kota Cimahi yang up to date, akurat, dan informatif;
2. Kebutuhan data kejadian pada kelurahan Kota Cimahi yang tepat dan akurat;
3. Kebutuhan laporan data kelurahan dan kejadian kelurahan yang mudah dan cepat;
4. Kebutuhan interface yang menarik, interactive, dan mudah digunakan.

# **SYSTEM NEEDS**

Daftar kebutuhan aplikasi Monografi Kelurahan Pemerintah Kota Cimahi yang terkait dengan kebutuhan sistem adalah sebagai berikut:

1. Sistem yang handal dan stabil;
2. Kemudahan dalam penggunaan;
3. Mudah dilakukan modifikasi apabila terjadi perubahan;
4. Kemudahan dalam menampilkan laporan dalam bentuk tabel dan dokumen;
5. Pembatasan hak akses penggunaan aplikasi.

# **INTERFACE SPESIFICATION**

Spesifikasi antarmuka dalam aplikasi Monografi Kelurahan Pemerintah Kota Cimahi adalah sebagai berikut:

1. Tampilan data yang informatif sesuai dengan format yang telah ditentukan;
2. Menampilkan laporan dalam bentuk tabel dan dokumen yang telah ditentukan.

# **PERANCANGAN**

Pada bagian perancangan ini akan dijelaskan lebih terperinci mengenai rancangan aplikasi Monografi Kelurahan Pemerintah Kota Cimahi dalam bentuk diagram-diagram sehingga dapat memberikan gambaran secara lebih terperinci tentang hal-hal teknis yang ada dalam aplikasi.

Mengacu pada daftar requirement spesification yang diajukan, maka secara umum desain sistem aplikasi Monografi Kelurahan Pemerintah Kota Cimahi dapat digambarkan kedalam sebuah bagan aplikasi sebagai berikut:



Gambar 1. Arsitektur Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi

Desain gambar 1 diatas disusun sebagai sebuah solusi menyeluruh bagi kebutuhan sebuah aplikasi yang fleksibel dan memungkinkan bagi pengguna dan pengelola aplikasi untuk dapat lebih berfokus pada pelayanan dan mengurangi kerumitan atau kesulitan. Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi merupakan aplikasi yang berbasis web yang dapat diakses melalui semua platform yang ada. Baik pada desktop, laptop, smartphone, maupun tablet.

Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi dibangun dengan menggunakan bahasa pemrograman di sisi server menggunakan PHP dengan *framework* PHP yaitu *Zend Framework* dan dari sisi client menggunakan HTML, CSS dan Javascript. Untuk penyimpanan data, aplikasi ini menggunakan mesin basis data (DBMS) SQL Server yang merupakan basis data utama aplikasi Monografi Kelurahan Pemerintah Kota Cimahi.



Gambar 2. Application Stack

Aplikasi dapat diakses menggunakan web browser oleh pengguna yang telah terdaftar atau memiliki hak akses sebagai pengguna aplikasi.

# **DIAGRAM KONTEKS**

Secara garis besar, aplikasi Monografi Kelurahan Pemerintah Kota Cimahi dapat digambarkan dalam sebuah diagram konteks sebagai berikut:



Gambar 3. Diagram Konteks Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi

Dalam aplikasi ini, ada 2 jenis hak akses yaitu admin dan pengguna. Admin memiliki hak akses untuk mengelola data monografi seluruh kelurahan Kota Cimahi. Sedangkan pengguna adalah operator kelurahan yang hanya memiliki hak akses untuk mengelola data monografi kelurahannya masing-masing.

# **DATA FLOW DIAGRAM**

Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi terbagi dalam 4 modul utama yaitu proses login, pemrosesan monografi kelurahan, proses kejadian kelurahan, dan proses laporan. Setiap proses membutuhkan dukungan data penyimpanan dalam pemrosesannya.



Gambar 4. Diagram Flow Diagram Level 1

## Proses Monografi Kelurahan

Monografi Kelurahan merupakan menu untuk mengelola data kelurahan. Gambar 7 merupakan detil dari proses Monografi Kelurahan, dimana terdapat empat proses yang dilakukan oleh admin dan pengguna sebagai berikut:



Gambar 5. Diagram Flow Level 2 (Proses Monografi Kelurahan)

## Proses Kejadian Kelurahan

Kejadian Kelurahan merupakan menu untuk mengelola kejadian-kejadian yang terdapat pada kelurahan Kota Cimahi. Gambar 7 merupakan detil dari proses Kejadian Kelurahan, dimana terdapat empat proses yang dilakukan oleh admin dan pengguna sebagai berikut:



Gambar 6. Diagram Flow Level 2 (Proses Kejadian Kelurahan)

## Proses Laporan

Menu laporan merupakan menu rekapan data monografi yang dapat dicetak dalam bentuk dokumen. Gambar 7 adalah detil dari proses laporan monografi sebagai berikut:



Gambar 7. Diagram Flow Level 2 (Proses Laporan Monografi Kelurahan)

# **DESAIN BASIS DATA**

Secara garis besar diagram basis data yang digunakan dalam aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi adalah sebagai berikut :



Gambar 8. Desain Basis Data

# **IMPLEMENTASI**

## Implementasi User Interface

Mengacu pada *requirements spesification* yang terkait dengan *user interface* aplikasi Monografi Kelurahan Pemerintah Kota Cimahi, berikut ini tampilan dari *user interface* aplikasi Monografi Kelurahan Pemerintah Kota Cimahi. *User interface* aplikasi ini dibangun menggunakan *Cascading Style Sheet* (CSS) dan javascript.

1. Login



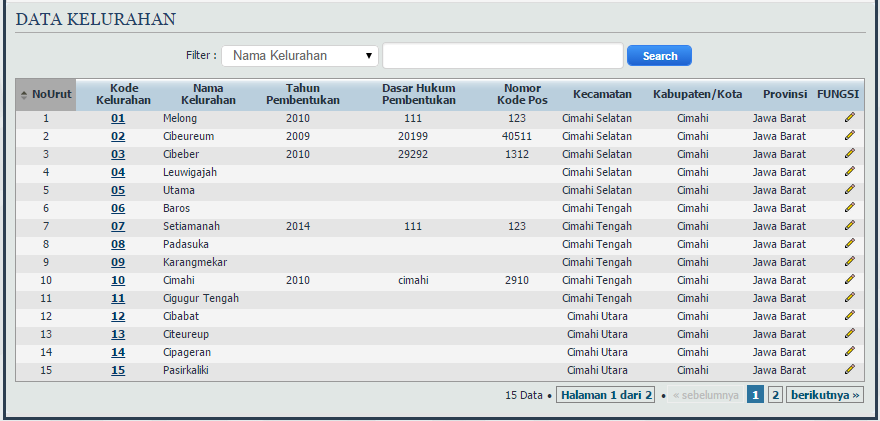
Gambar 9. Halaman Login

1. Halaman Utama Modul Monografi

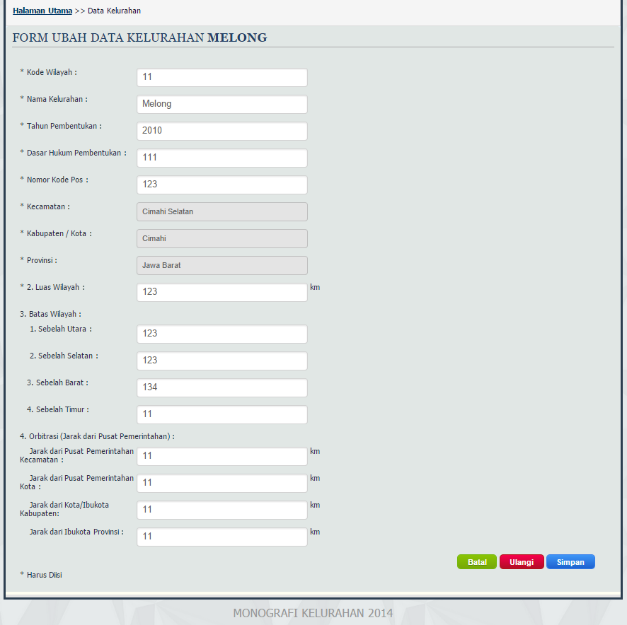


Gambar 10. Halaman Utama Aplikasi

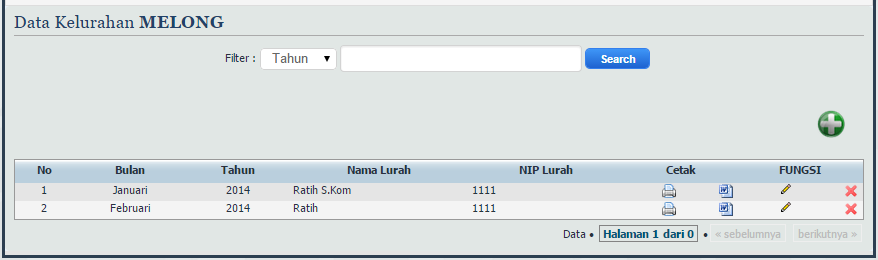
1. Menu Monografi Kelurahan



Gambar 11. Tampilan Data Monografi Kelurahan



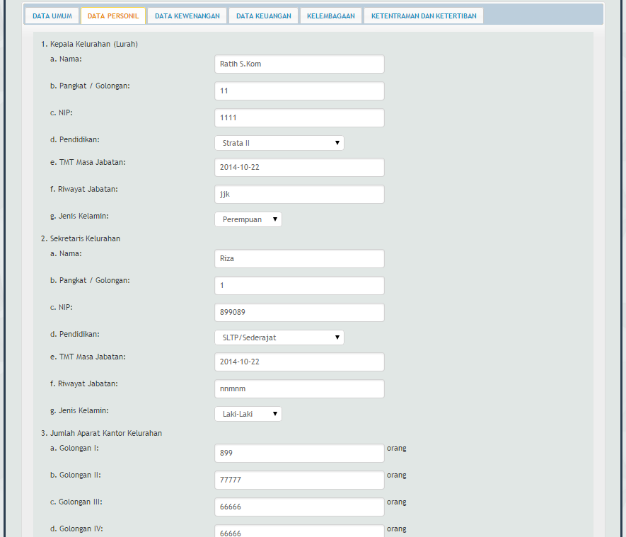
Gambar 12. Tampilan Ubah Data Kelurahan



Gambar 13. Detail Data Monografi Kelurahan Melong



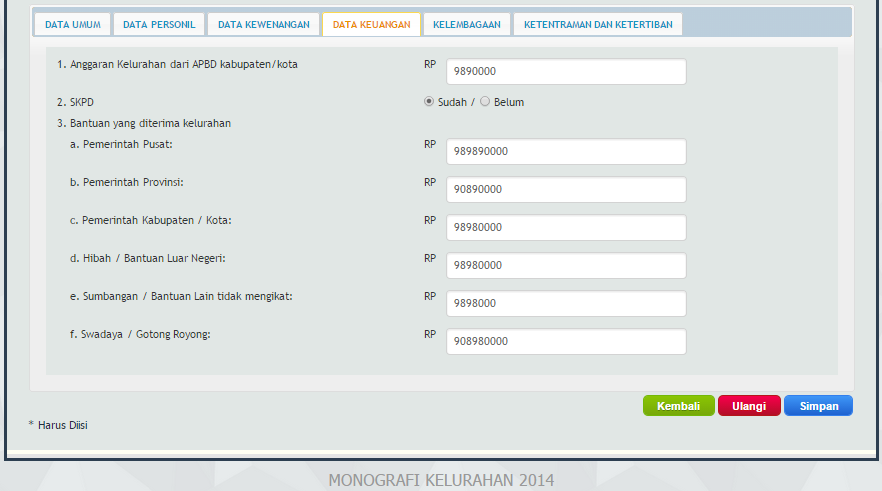
Gambar 14. Tambah Data Umum Kelurahan



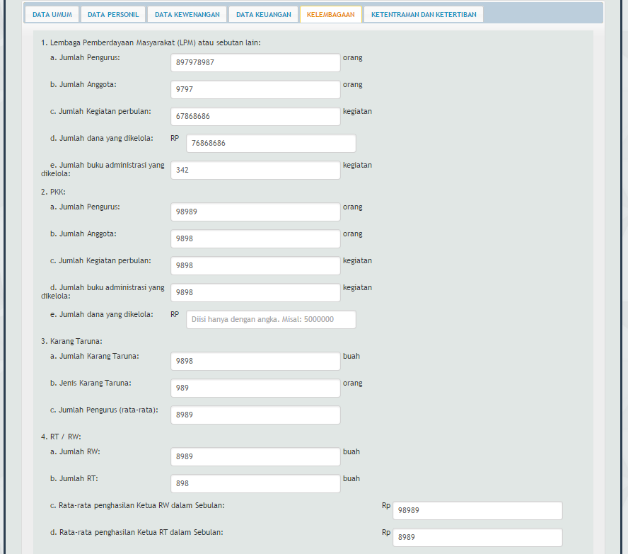
Gambar 15. Tambah Data Personil Kelurahan



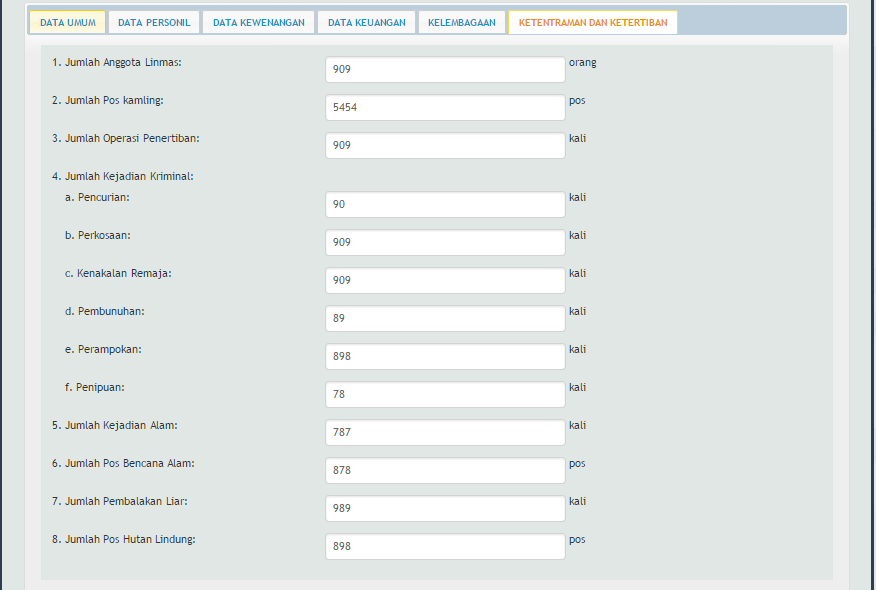
Gambar 16. Tambah Data Kewenangan Kelurahan



Gambar 17. Tambah Data Keuangan Kelurahan



Gambar 18. Tambah Data Kelembagaan Kelurahan

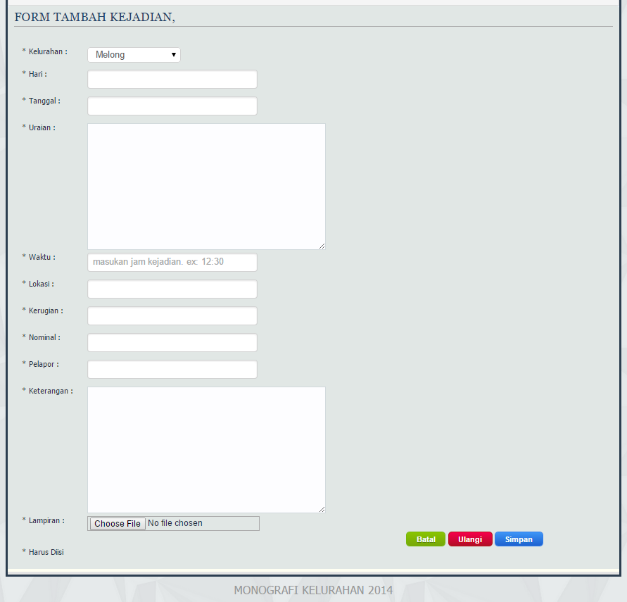


Gambar 19. Tambah Data Ketentraman dan Ketertiban

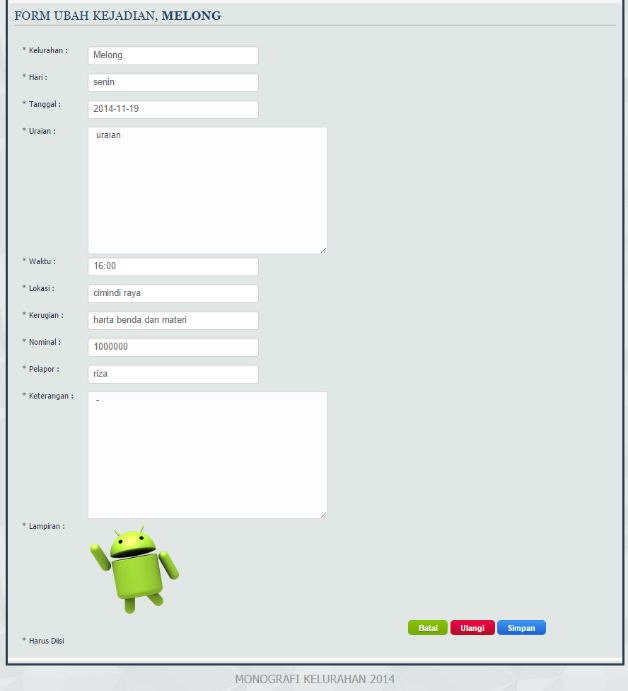
1. Menu Kejadian Kelurahan



Gambar 20. Tampilan Data Kejadian Kelurahan

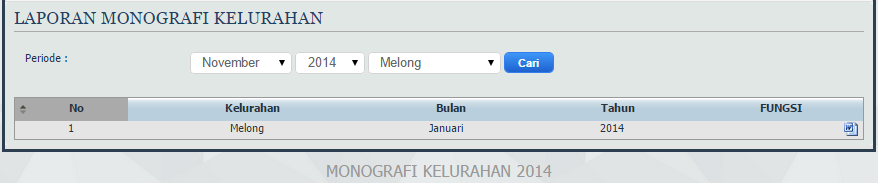


Gambar 21. Tambah Data Kejadian Kelurahan



Gambar 22. Ubah Data Kejadian Kelurahan

1. Menu Laporan Monografi Kelurahan



Gambar 23. Tampilan Menu Laporan Monografi Kelurahan

# **IMPLEMENTASI SISTEM**

Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi ini dibangun menggunakan bahasa pemrograman berbasis web, yaitu bahasa pemrograman php sebagai *server side* script dan html, javascript yang script yang berjalan di sisi *client* pengguna aplikasi. Bahasa pemrograman php yang digunakan menggunakan metode MVC (*model*, *view*, *controller*) dengan memanfaatkan fungsi-fungsi MVC yang ada pada php *Zend Framework*.

Untuk dapat berkomunikasi dengan database digunakan *ADOdb Library for PHP* *V5.11* yang merupakan *library* khusus untuk mengakses database dengan menggunakan bahasa pemrograman PHP. Library ini sangat mudah digunakan dan handal, disamping itu juga dapat dengan mudah melakukan modifikasi jika nantinya diperlukan adanya pemindahan DBMS dari yang sekarang digunakan yaitu SQL Server ke DBMS yang lain. Karena pada *library* ini mendukung koneksi ke beberapa jenis DBMS.

## Kode Sumber (Source Code)

HomeController.php

<?php

class Home\_IndexController extends Zend\_Controller\_Action {

public function init(){

/\* Initialize action controller here \*/

}

public function indexAction(){

// action body

}

public function homeAction(){

$pengguna = $\_POST['pengguna'];

$password = $\_POST['password'];

if ($pengguna && $password) {

$hasiluser = $this->sso\_serv-

>getDataUser1($pengguna,$password);

if($hasiluser){

$ssouserpengguna = new

Zend\_Session\_Namespace('ssouserpengguna');

$ssouserpassword= new

Zend\_Session\_Namespace('ssouserpassword');

$ssouserKodeInstansi = new

Zend\_Session\_Namespace('ssouserKodeInstansi');

$ssouserlevel = new

Zend\_Session\_Namespace('ssouserlevel');

$pengguna = $hasiluser->pengguna;

$password = $hasiluser->password;

$KodeInstansi = $hasiluser->KodeInstansi;

$level = $hasiluser->level;

$instansi = $hasiluser->Instansi;

$ssouserpengguna->pengguna = $hasiluser-

>pengguna;

$ssouserpassword->password = $hasiluser-

>password;

$ssouserKodeInstansi->KodeInstansi =

$hasiluser->KodeInstansi;

$ssouserlevel->level = $hasiluser->level;

$this->view->pengguna = $ssouserpengguna-

>pengguna;

$this->view->password = $ssouserpassword-

>password;

$this->view->KodeInstansi =

$ssouserKodeInstansi->KodeInstansi;

$this->view->level = $ssouserlevel->level;

$this->view->Instansi = $instansi;

if (!$pengguna){

$pengguna =$this->pengguna;

}

$this->view;

$this->view->menuKomponen = $this->menu\_serv-

>getKomponen();

$this->view->menuKomponenSub = $this-

>menu\_serv->getKomponenSub();

$this->view->menuKomponenSubDetail = $this-

>menu\_serv->getKomponenSubDetail();

$this->view->kelurahan = $this->menu\_serv-

>getKelurahan();

$this->view->kecamatan = $this->menu\_serv-

>getKecamatan();

}else{

$this->view->pengguna=$\_POST['pengguna'];

$this->view->password=$\_POST['password'];

$this->view->pesanlogin ="salah";

$this->view->pesanKesalahan = 'Nama Pengguna atau Kata Sandi Salah atau menu aplikasi ini bukan autorisasi anda';

$this->indexAction();

$this->render('index');

$this->view->par=$\_POST['par'];

}

} else {

$this->view->pesanlogin ="kosong";

$this->view->pesan = "User dan Password Kosong";

$this->indexAction();

$this->render('index');

$this->view->par=$\_POST['par'];

}

}

public function logoutAction(){

$globalSession = new

Zend\_Session\_Namespace('intervesi\_kemiskinan');

$sid = $globalSession->session\_id;

Zend\_Session::destroy(true);

$this->\_redirect('../');

}

public function sessioncheckAction()

{

// action body

}

public function updatelogAction()

{

// action body

}

KegiatanController.php

<?php

class Kegiatan\_KegiatanController extends Zend\_Controller\_Action{

public function init(){

/\* Initialize action controller here \*/

}

public function kegiatanmenuAction(){

$currentPage = $\_REQUEST['currentPage'];

if((!$currentPage) || ($currentPage == 'undefined')){

$currentPage = 1;

}

$kategoriCari = $\_REQUEST['kategoriCari'];

$katakunciCari = trim($\_POST['carii']);

$sortBy = 'KodeKegiatan';

$sort = 'asc';

$KodeInstansi = $this->KodeInstansi;

$dataMasukan = array("kategoriCari" => $kategoriCari,

"katakunciCari" => $katakunciCari,

"sortBy" => $sortBy,

"sort" => $sort);

$numToDisplay = 10;

$this->view->cari = $katakunciCari;

$this->view->KodeInstansi = $this->KodeInstansi;

$this->view->numToDisplay = $numToDisplay;

$this->view->currentPage = $currentPage;

$this->view->totKegiatan = $this->kegiatan\_serv-

>getcarikegiatan($KodeInstansi, $dataMasukan,0,0,0);

$this->view->kegiatanMenu = $this->kegiatan\_serv-

>getcarikegiatan($KodeInstansi,

$dataMasukan,$currentPage, $numToDisplay,$this->view-

>totKegiatan);

}

//Manajemen Kegiatan

public function kegiatanolahdataAction(){

$this->view->jenisForm = $this->\_getParam('jenisForm');

$this->view->KodeKegiatan= $\_REQUEST['KodeKegiatan'];

$this->view->KodeInstansi = $this->KodeInstansi;

$this->view->kegiatanMenu = $this->kegiatan\_serv-

>getkegiatanedit($this->view->KodeKegiatan, $this->view-

>KodeInstansi);

$this->view->Instansi = $this->kegiatan\_serv-

>getInstansi($this->view->KodeInstansi);

$this->view->instansiList = $this->kegiatan\_serv-

>getInstansiListAll();

$this->view->komponenList = $this->kegiatan\_serv-

>getKomponenListAll();

$this->view->subKomponenList = $this->kegiatan\_serv-

>getSubKomponenListAll();

$this->view->subKomponenDetailList = $this-

>kegiatan\_serv->getSubKomponenDetailListAll();

}

//Tambah Penerima berdasarkan kegiatan yang dipilih

public function penerimamenuAction(){

$KodeKegiatan = $this->\_getParam("KodeKegiatan");

$currentPage = $\_REQUEST['currentPage'];

if((!$currentPage) || ($currentPage == 'undefined')){

$currentPage = 1;

}

$KodeKegiatan = $\_REQUEST['KodeKegiatan'];

$kategoriCari = $\_REQUEST['kategoriCari'];;

$katakunciCari = $\_POST['carii'];

$sortBy = 'KodeKegiatan';

$sort = 'asc';

$dataMasukan = array("kategoriCari" => $kategoriCari,

"katakunciCari" => $katakunciCari,

"sortBy" => $sortBy,

"sort" => $sort);

$numToDisplay = 10;

$this->view->KodeKegiatan = $KodeKegiatan;

$this->view->kategoriCari = $kategoriCari;

$this->view->cari = $katakunciCari;

$this->view->numToDisplay = $numToDisplay;

$this->view->currentPage = $currentPage;

$this->view->totKegiatan = $this->kegiatan\_serv-

>getpenerima($KodeKegiatan,$dataMasukan,0,0,0);

$this->view->penerimaMenu = $this->kegiatan\_serv-

>getpenerima($KodeKegiatan, $dataMasukan,$currentPage,

$numToDisplay,$this->view->totKegiatan);

$this->view->penerimaMenu2 = $this->kegiatan\_serv-

>getpenerima2($KodeKegiatan,$currentPage,

$numToDisplay,$this->view->totKegiatan);

}

//Manajemen Penerima Bantuan Berdasarkan Kegiatan yang Dipilih

public function penerimaolahdataAction(){

$this->view->jenisForm = $\_REQUEST['jenisForm'];

$KodeKegiatan= $\_REQUEST['KodeKegiatan'];

$NamaKegiatan = $this->\_getParam("NamaKegiatan");

$NIK = $\_REQUEST["NIK"];

$this->view->kelurahanList = $this->kegiatan\_serv-

>getKelurahanListAll();

$this->view->KodeKegiatan = $KodeKegiatan;

$this->view->NamaKegiatan = $NamaKegiatan;

$this->view->penerimaOlahMenu = $this-

>kegiatan\_serv->getpenerimaedit($NIK, $KodeKegiatan);

}

# **IMPLEMENTASI BASIS DATA**

Aplikasi Monografi Kelurahan Pemerintah Kota Cimahi menggunakan SQL Server sebagai mesin penyimpanan data. Berikut ini detail informasi yang berkaitan dengan struktur data SQL Server yang digunakan oleh aplikasi Monografi Kelurahan Pemerintah Kota Cimahi.

## Data Definition Language

<?Php

public function getsimpankegiatan(array $dataMasukan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->beginTransaction();

$paramInput = array("KodeKegiatan" =>

$dataMasukan['KodeKegiatan'],

"NamaKegiatan" => $dataMasukan['NamaKegiatan'],

"JadwalAwal" => $dataMasukan['JadwalAwal'],

"JadwalAkhir" => $dataMasukan['JadwalAkhir'],

"KodeInstansi" => $dataMasukan['KodeInstansi'],

"KodeKomponen" => $dataMasukan['KodeKomponen'],

"KodeSubKomponen" =>

$dataMasukan['KodeSubKomponen'],

"KodeDetailSubKomponen" =>

$dataMasukan['KodeDetailSubKomponen'],

"Anggaran" => $dataMasukan['Anggaran']);

$db->insert('kegiatan',$paramInput);

$db->commit();

return 'sukses';

} catch (Exception $e) {

$db->rollBack();

echo $e->getMessage().'<br>';

return 'gagal';

}

}

function getkegiatanedit($KodeKegiatan, $KodeInstansi){

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->setFetchMode(Zend\_Db::FETCH\_OBJ);

$where = " and KodeKegiatan = '$KodeKegiatan' ";

if($KodeInstansi==0){

$sqlProses = "select k.KodeKegiatan,

k.NamaKegiatan, k.JadwalAwal,k.KodeInstansi,

k.JadwalAkhir, k.KodeKomponen,

k.KodeSubKomponen, k.KodeDetailSubKomponen,

k.Anggaran, i.Instansi from kegiatan k,

m\_instansi i where k.KodeInstansi=i.KodeInstansi";

}else{

$sqlProses = "select k.KodeKegiatan,

k.NamaKegiatan, k.JadwalAwal,k.KodeInstansi,

k.JadwalAkhir, k.KodeKomponen,

k.KodeSubKomponen, k.KodeDetailSubKomponen,

k.Anggaran, i.Instansi from kegiatan k,

m\_instansi i where k.KodeInstansi=i.KodeInstansi

and k.KodeInstansi='$KodeInstansi'";

}

$sqlData = $sqlProses.$where;

$result = $db->fetchRow($sqlData);

$hasilAkhir = array("KodeKegiatan"

=>(string)$result->KodeKegiatan,

"NamaKegiatan" =>(string)$result->NamaKegiatan,

"JadwalAwal" =>(string)$result->JadwalAwal,

"JadwalAkhir" =>(string)$result->JadwalAkhir,

"KodeInstansi" =>(string)$result->KodeInstansi,

"Instansi" =>(string)$result->Instansi,

"KodeKomponen" =>(string)$result->KodeKomponen,

"KodeSubKomponen"=>(string)$result->KodeSubKomponen,

"KodeDetailSubKomponen" =>(string)$result-

>KodeDetailSubKomponen,

"Anggaran" =>(string)$result->Anggaran );

return $hasilAkhir;

} catch (Exception $e) {

echo $e->getMessage().'<br>';

return 'Data tidak ada <br>';

}

}

public function getsimpankegiatanedit(array $dataMasukan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->beginTransaction();

$paramInput = array("KodeKegiatan" =>

$dataMasukan['KodeKegiatan'],

"NamaKegiatan" => $dataMasukan['NamaKegiatan'],

"JadwalAwal" => $dataMasukan['JadwalAwal'],

"JadwalAkhir" => $dataMasukan['JadwalAkhir'],

"KodeInstansi" => $dataMasukan['KodeInstansi'],

"KodeKomponen" => $dataMasukan['KodeKomponen'],

"KodeSubKomponen" => $dataMasukan['KodeSubKomponen'],

"KodeDetailSubKomponen" =>

$dataMasukan['KodeDetailSubKomponen'],

"Anggaran" => $dataMasukan['Anggaran']);

$where[] = " KodeKegiatan =

'".$dataMasukan['KodeKegiatan']."'";

$db->update('kegiatan',$paramInput, $where);

$db->commit();

return 'sukses';

} catch (Exception $e) {

$db->rollBack();

$errmsgArr = explode(":",$e->getMessage());

$errMsg = $errmsgArr[0];

if($errMsg == "SQLSTATE[23000]")

{

return "gagal.Data Sudah Ada.";

}

else

{

return "sukses";

}

}

}

public function gethapuskegiatan($KodeKegiatan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->beginTransaction();

$where[] = " KodeKegiatan = '".$KodeKegiatan."'";

$db->delete('kegiatan', $where);

$db->commit();

return 'sukses';

} catch (Exception $e) {

$db->rollBack();

$errmsgArr = explode(":",$e->getMessage());

$errMsg = $errmsgArr[0];

if($errMsg == "SQLSTATE[23000]")

{

return "gagal.Data Sudah Ada.";

}

else

{

return "sukses";

}

}

}

public function getcarikegiatan($KodeInstansi, array $dataMasukan, $pageNumber, $itemPerPage,$total){

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

$kategoriCari = $dataMasukan['kategoriCari'];

$katakunciCari = $dataMasukan['katakunciCari'];

$sortBy = $dataMasukan['sortBy'];

$sort = $dataMasukan['sort'];

try {

$db->setFetchMode(Zend\_Db::FETCH\_OBJ);

$xLimit=$itemPerPage;

$xOffset=($pageNumber-1)\*$itemPerPage;

$whereOpt = " and ($kategoriCari like '%$katakunciCari%')";

if($katakunciCari != "") { $where = $whereOpt;}

$order = " order by k.JadwalAkhir desc";

if($KodeInstansi == 0){

$sqlProses = "select k.KodeKegiatan,

k.NamaKegiatan, k.JadwalAwal,k.KodeInstansi,

k.JadwalAkhir, k.KodeKomponen,

k.KodeSubKomponen, k.KodeDetailSubKomponen,

k.Anggaran, i.Instansi from kegiatan k,

m\_instansi i where k.KodeInstansi=i.KodeInstansi".$where;

}else{

$sqlProses = "select k.KodeKegiatan,

k.NamaKegiatan, k.JadwalAwal,k.KodeInstansi,

k.JadwalAkhir, k.KodeKomponen,

k.KodeSubKomponen, k.KodeDetailSubKomponen,

k.Anggaran, i.Instansi from kegiatan k,

m\_instansi i where

k.KodeInstansi=i.KodeInstansi

and k.KodeInstansi=$KodeInstansi".$where;

}

$sqlProses1 = $sqlProses.$order;

if(($pageNumber==0) && ($itemPerPage==0)){

$sqlTotal = "select count(\*) from ($sqlProses) a";

$hasilAkhir = $db->fetchOne($sqlTotal);

}else{

$sqlData = $sqlProses.$order." limit $xLimit

offset $xOffset";

$result = $db->fetchAll($sqlData);

}

$jmlResult = count($result);

for ($j = 0; $j < $jmlResult; $j++) {

$hasilAkhir[$j] = array("KodeKegiatan"

=>(string)$result[$j]->KodeKegiatan,

"NamaKegiatan" =>(string)$result[$j]->NamaKegiatan,

"JadwalAwal" =>(string)$result[$j]->JadwalAwal,

"JadwalAkhir" =>(string)$result[$j]->JadwalAkhir,

"KodeInstansi" =>(string)$result[$j]->KodeInstansi,

"Instansi" =>(string)$result[$j]->Instansi,

"KodeKomponen" =>(string)$result[$j]->KodeKomponen,

"KodeSubKomponen" =>(string)$result[$j]->KodeSubKomponen,

"KodeDetailSubKomponen" =>(string)$result[$j]->KodeDetailSubKomponen,

"Anggaran" =>(string)$result[$j]->Anggaran);

}

return $hasilAkhir;

} catch (Exception $e) {

echo $e->getMessage().'<br>';

return 'gagal <br>';

}

}

>?

<?php

public function getNamaPenduduk2(array $dataMasukan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

$NIK = ($dataMasukan['NIK']);

try {

$db->setFetchMode(Zend\_Db::FETCH\_OBJ);

if($NIK == ''){

$sql = "SELECT \* FROM data\_penduduk ";

$result = $db->fetchAll($sql);

// $kataKunci = '--';

}else{

$sql = "SELECT \* FROM data\_penduduk where NIK like '$NIK%' ";

$result = $db->fetchAll($sql);

}

$jmlResult = count($result);

for ($j = 0; $j < $jmlResult; $j++) {

$data[$j] = array("NIK"

=>(string)$result[$j]->NIK,

"NamaLengkap" =>(string)$result[$j]->NamaLengkap,

"Alamat" =>(string)$result[$j]->Alamat,

"RT" =>(string)$result[$j]->RT,

"RW" =>(string)$result[$j]->RW,

"Dusun" =>(string)$result[$j]->Dusun,

"KodePos" =>(string)$result[$j]->KodePos,

"JK" =>(string)$result[$j]->JK,

"TempatLahir" =>(string)$result[$j]->TempatLahir,

"TglLahir" =>(string)$result[$j]->TglLahir,

"NoAkta" =>(string)$result[$j]->NoAkta,

"GolDarah" =>(string)$result[$j]->GolDarah,

"Agama" =>(string)$result[$j]->Agama,

"Pekerjaan" =>(string)$result[$j]->Pekerjaan,

"NamaIbu"=>(string)$result[$j]->NamaIbu,

"NamaAyah"=>(string)$result[$j]->NamaAyah,

"Status" =>(string)$result[$j]->Status,

"Kelurahan" =>(string)$result[$j]->Kelurahan

);

}

return $data;

} catch (Exception $e) {

echo $e->getMessage().'<br>';

return 'Data tidak ada <br>';

}

}

public function getpenerima($KodeKegiatan, array $dataMasukan,

$pageNumber, $itemPerPage, $total){

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

$kategoriCari = $dataMasukan['kategoriCari'];

$katakunciCari = $dataMasukan['katakunciCari'];

$sortBy = $dataMasukan['sortBy'];

$sort = $dataMasukan['sort'];

try {

$db->setFetchMode(Zend\_Db::FETCH\_OBJ);

$xLimit=$itemPerPage;

$xOffset=($pageNumber-1)\*$itemPerPage;

$whereOpt = " AND ($kategoriCari like

%$katakunciCari%')";

if($katakunciCari != "") { $where = $whereOpt;}

$group = " GROUP BY p.KodeKegiatan ";

$order = " order by p.NIK ";

$sqlProses = "SELECT p.NIK, p.NamaLengkap, p.Alamat,

p.Status , p.Kelurahan, p.Ket, p.KodeKegiatan

FROM pendaftaran p

WHERE p.KodeKegiatan = '$KodeKegiatan' ".$where;

$sqlProses1 = $sqlProses.$group.$order;

if(($pageNumber==0) && ($itemPerPage==0)){

$sqlTotal = "select count(\*) from ($sqlProses)

a";

$hasilAkhir = $db->fetchOne($sqlTotal);

}else{

$sqlData = $sqlProses.$order." limit $xLimit

offset $xOffset";

$result = $db->fetchAll($sqlData);

}

$jmlResult = count($result);

for ($j = 0; $j < $jmlResult; $j++) {

$hasilAkhir[$j] = array("NIK"

=>(string)$result[$j]->NIK,

"NamaLengkap" =>(string)$result[$j]

->NamaLengkap,

"Alamat" =>(string)$result[$j]->Alamat,

"Status" =>(string)$result[$j]->Status,

"Kelurahan" =>(string)$result[$j]-

>Kelurahan,

"KodeKegiatan" =>(string)$result[$j]-

>KodeKegiatan,

"Ket" =>(string)$result[$j]->Ket

);

}

return $hasilAkhir;

} catch (Exception $e) {

echo $e->getMessage().'<br>';

return 'gagal <br>';

}

}

public function getsimpanpenerima(array $dataMasukan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->beginTransaction();

$paramInput = array("IdPendaftaran" =>

$dataMasukan['IdPendaftaran'],

"NIK" => $dataMasukan['NIK'],

"NamaLengkap" => $dataMasukan['NamaLengkap'],

"Alamat" => $dataMasukan['Alamat'],

"Status" => $dataMasukan['Status'],

"Kelurahan" => $dataMasukan['Kelurahan'],

"KodeKegiatan" =>$dataMasukan['KodeKegiatan'],

"Ket" => $dataMasukan['Ket']);

$db->insert('pendaftaran',$paramInput);

$db->commit();

return 'sukses';

} catch (Exception $e) {

$db->rollBack();

$errmsgArr = explode(":",$e->getMessage());

$errMsg = $errmsgArr[0];

if($errMsg == "SQLSTATE[23000]")

{

return "gagal.Data Sudah Ada.";

}

else

{

return "sukses";

}

}

}

public function getCekDuplikasi($NIK, $KodeKegiatan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->setFetchMode(Zend\_Db::FETCH\_OBJ);

$sql = "SELECT \* FROM pendaftaran WHERE NIK ='$NIK'

AND KodeKegiatan='$KodeKegiatan'";

$result = $db->fetchAll($sql);

$jmlResult = count($result);

for ($j = 0; $j < $jmlResult; $j++) {

$data[$j] = array("IdPendaftaran"

=>(string)$result->IdPendaftaran,

"NIK" =>(string)$result->NIK,

"NamaLengkap" =>(string)$result->NamaLengkap,

"Alamat" =>(string)$result->Alamat,

"Status" =>(string)$result->Status,

"Kelurahan" =>(string)$result->Kelurahan,

"KodeKegiatan" =>(int)$result->KodeKegiatan,

"Ket" =>(string)$result->Ket);

}

return $data;

} catch (Exception $e) {

echo $e->getMessage().'<br>';

return 'Data tidak ada <br>';

}

}

public function getpenerimaedit($NIK, $KodeKegiatan){

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->setFetchMode(Zend\_Db::FETCH\_OBJ);

$sqlProses = " SELECT p.\*, k.NamaKegiatan

FROM pendaftaran p, kegiatan k WHERE

p.KodeKegiatan=k.KodeKegiatan and NIK = '$NIK' and

p.KodeKegiatan='$KodeKegiatan'";

$sqlData = $sqlProses;

$result = $db->fetchRow($sqlData);

$hasilAkhir = array("IdPendaftaran"

=>(string)$result->IdPendaftaran,

"NIK" =>(string)$result->NIK,

"NamaLengkap" =>(string)$result->NamaLengkap,

"Alamat" =>(string)$result->Alamat,

"Status" =>(string)$result->Status,

"Kelurahan" =>(string)$result->Kelurahan,

"KodeKegiatan" =>(int)$result->KodeKegiatan,

"NamaKegiatan" =>(string)$result->NamaKegiatan,

"Ket" =>(string)$result->Ket);

return $hasilAkhir;

} catch (Exception $e) {

echo $e->getMessage().'<br>';

return 'Data tidak ada <br>';

}

}

public function getsimpanpenerimaedit(array $dataMasukan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->beginTransaction();

$paramInput = array("IdPendaftaran" =>

$dataMasukan['IdPendaftaran'],

"NIK" => $dataMasukan['NIK'],

"NamaLengkap" => $dataMasukan['NamaLengkap'],

"Alamat" => $dataMasukan['Alamat'],

"Status" => $dataMasukan['Status'],

"Kelurahan" => $dataMasukan['Kelurahan'],

"KodeKegiatan" => $dataMasukan['KodeKegiatan'],

"Ket" => $dataMasukan['Ket']);

$where[] = " NIK = '".$dataMasukan['NIK']."' and

KodeKegiatan ='".$dataMasukan['KodeKegiatan']."' ";

$db->update('pendaftaran',$paramInput, $where);

$db->commit();

return 'sukses';

} catch (Exception $e) {

$db->rollBack();

$errmsgArr = explode(":",$e->getMessage());

$errMsg = $errmsgArr[0];

if($errMsg == "SQLSTATE[23000]")

{

return "gagal.Data Sudah Ada.";

}

else

{

return "sukses";

}

}

}

public function gethapuspenerima($NIK, $KodeKegiatan) {

$registry = Zend\_Registry::getInstance();

$db = $registry->get('db');

try {

$db->beginTransaction();

$where[] = " NIK = '".$NIK."' and

KodeKegiatan='".$KodeKegiatan."' ";

$db->delete('pendaftaran', $where);

$db->commit();

return 'sukses';

} catch (Exception $e) {

$db->rollBack();

$errmsgArr = explode(":",$e->getMessage());

$errMsg = $errmsgArr[0];

if($errMsg == "SQLSTATE[23000]")

{

return "gagal.Data Sudah Ada.";

}

else

{

return "sukses";

}

}

}

?>php