TIM IT KAPPDE 2014

PEMERINTAH KOTA CIMAHI

APLIKASI INTERVENSI KEMISKINAN

**Dokumen Pengembangan Aplikasi**

# **DAFTAR ISI**

[**DAFTAR ISI** 1](#_Toc403211431)

[**PENDAHULUAN** 4](#_Toc403211432)

[**REQUIREMENT SPECIFICATION** 4](#_Toc403211433)

[**SYSTEM NEEDS** 5](#_Toc403211434)

[**INTERFACE SPESIFICATION** 5](#_Toc403211435)

[**PERANCANGAN** 6](#_Toc403211436)

[**DIAGRAM KONTEKS** 7](#_Toc403211437)

[**DATA FLOW DIAGRAM** 8](#_Toc403211438)

[Login 9](#_Toc403211439)

[Pemrosesan Pengguna 9](#_Toc403211440)

[Proses Komponen 10](#_Toc403211441)

[Proses Sub Komponen 11](#_Toc403211442)

[Proses Detail Sub Komponen 11](#_Toc403211443)

[Proses Kelurahan 12](#_Toc403211444)

[Proses Kecamatan 13](#_Toc403211445)

[Proses Kota 13](#_Toc403211446)

[Proses Provinsi 14](#_Toc403211447)

[Proses Kegiatan 15](#_Toc403211448)

[**DESAIN BASIS DATA** 16](#_Toc403211449)

[**IMPLEMENTASI** 17](#_Toc403211450)

[Implementasi User Interface 17](#_Toc403211451)

[**IMPLEMENTASI SISTEM** 32](#_Toc403211452)

[Kode Sumber (Source Code) 33](#_Toc403211453)

[**IMPLEMENTASI BASIS DATA** 37](#_Toc403211454)

# **DAFTAR GAMBAR**

[Gambar 1. Arsitektur Aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi 6](#_Toc403216313)

[Gambar 2. Application Stack 7](#_Toc403216314)

[Gambar 3. Diagram Konteks Aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi 7](#_Toc403216315)

[Gambar 4. Diagram Flow Diagram Level 1 8](#_Toc403216316)

[Gambar 5. Data Flow Diagram Level 2 (Login) 9](#_Toc403216317)

[Gambar 6. Diagram Flow Diagram Level 2 (Pemrosesan Pengguna) 10](#_Toc403216318)

[Gambar 7. Diagram Flow Diagram Level 2 (Pemrosesan Komponen) 10](#_Toc403216319)

[Gambar 8. Diagram Flow Diagram Level 2 (Pemrosesan Sub Komponen) 11](#_Toc403216320)

[Gambar 9. Diagram Flow Diagram Level 2 (Pemrosesan Detail Sub Komponen) 12](#_Toc403216321)

[Gambar 10. Diagram Flow Diagram Level 2 (Pemrosesan Kelurahan) 12](#_Toc403216322)

[Gambar 11. Diagram Flow Diagram Level 2 (Pemrosesan Kecamatan) 13](#_Toc403216323)

[Gambar 12. Diagram Flow Diagram Level 2 (Pemrosesan Kota) 13](#_Toc403216324)

[Gambar 13. Diagram Flow Diagram Level 2 (Pemrosesan Provinsi) 14](#_Toc403216325)

[Gambar 14. Diagram Flow Diagram Level 2 (Pemrosesan Kegiatan) 15](#_Toc403216326)

[Gambar 15. Desain Basis Data 16](#_Toc403216327)

[Gambar 16. Halaman Login 17](#_Toc403216328)

[Gambar 17. Halaman Utama Aplikasi 17](#_Toc403216329)

[Gambar 18. Tampilan Data Pengguna 18](#_Toc403216330)

[Gambar 19. Tampilan Form Tambah Pengguna 18](#_Toc403216331)

[Gambar 20. Tampilan Form Ubah Data Pengguna 19](#_Toc403216332)

[Gambar 21. Form Tampilan Data Komponen 19](#_Toc403216333)

[Gambar 22. Tampilan Form Tambah Komponen 19](#_Toc403216334)

[Gambar 23. Tampilan Form Ubah Data Komponen 20](#_Toc403216335)

[Gambar 24. Tampilan Data Sub Komponen 20](#_Toc403216336)

[Gambar 25. Tampilan Form Tambah Sub Komponen 20](#_Toc403216337)

[Gambar 26. Tampilan Form Ubah Sub Komponen 21](#_Toc403216338)

[Gambar 27. Tampilan Data Detail Sub Komponen 21](#_Toc403216339)

[Gambar 28. Tampilan Form Tambah Detail Sub Komponen 22](#_Toc403216340)

[Gambar 29. Tampilan Form Ubah Data Detail Sub Komponen 22](#_Toc403216341)

[Gambar 30. Tampilan Data Kelurahan 23](#_Toc403216342)

[Gambar 31. Tampilan Form Tambah Kelurahan 23](#_Toc403216343)

[Gambar 32. Tampilan Form Ubah Kelurahan 23](#_Toc403216344)

[Gambar 33. Tampilan Data Provinsi 24](#_Toc403216345)

[Gambar 34. Tampilan Form Tambah Provinsi 24](#_Toc403216346)

[Gambar 35. Tampilan Form Ubah Data Provinsi 24](#_Toc403216347)

[Gambar 36. Tampilan Data Kota 25](#_Toc403216348)

[Gambar 37. Tampilan Form Tambah Kota 25](#_Toc403216349)

[Gambar 38. Tampilan Form Ubah Data Kota 25](#_Toc403216350)

[Gambar 39. Tampilan Data Kegiatan Admin 26](#_Toc403216351)

[Gambar 40. Tampilan Form Tambah Kegiatan Admin 26](#_Toc403216352)

[Gambar 41. Tampilan Form Ubah Kegiatan Admin 27](#_Toc403216353)

[Gambar 42. Tampilan Data Kegiatan Pengguna 27](#_Toc403216354)

[Gambar 43. Tampilan Form Tambah Kegiatan Pengguna 28](#_Toc403216355)

[Gambar 44. Tampilan Form Ubah Kegiatan Pengguna 28](#_Toc403216356)

[Gambar 45. Tampilan Menu Penerima Berdasarkan Kegiatan yang Dipilih 29](#_Toc403216357)

[Gambar 46. Tampilan Form Tambah Penerima 29](#_Toc403216358)

[Gambar 47. Tampilan Popoup Penduduk PPLS untuk Tambah Penerima 30](#_Toc403216359)

[Gambar 48. Tampilan Form Ubah Data Penerima 30](#_Toc403216360)

[Gambar 49. Tampilan Rekap Kelurahan 31](#_Toc403216361)

[Gambar 50. Tampilan Hasil Rekap Kelurahan 31](#_Toc403216362)

[Gambar 51.Tampilan Rekap Kegiatan 31](#_Toc403216363)

[Gambar 52. Tampilan Hasil rekap Kegiatan 32](#_Toc403216364)

[Gambar 53. Tampilan Kontak 32](#_Toc403216365)

# **PENDAHULUAN**

Aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi adalah sebuah aplikasi berbasis web yang digunakan untuk memonitoring pemerataan intervensi kemiskinan dan pendataan masyarakat kurang mampu sehingga bantuan Pemerintah Kota Cimahi bisa terpantau dan tepat sasaran.

Mengacu kepada *requirement specification* yang ada, aplikasi Intervensi Kemiskinan 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 pemakai/pengakses aplikasi oleh pengelola aplikasi Intervensi Kemiskinan.

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

Fleksibilitas yang ditawarkan aplikasi Intervensi Kemiskinan 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 Intervensi Kemiskinan Kota Cimahi secara teknikal, baik dari sisi aplikasi maupun dari sisi sistem basis data yang digunakan.

Penjelasan aplikasi Intervensi Kemiskinan 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 Intervensi Kemiskinan Pemerintah Kota Cimahi dibangun berdasar kepada daftar kebutuhan yang telah ditetapkan oleh Badan Perencanaan dan Pembangunan Daerah (Bappeda) Kota Cimahi. Secara umum, spesifikasi kebutuhan aplikasi Intervensi Kemiskinan Kota Cimahi dapat dibagi menjadi tiga kebutuhan, yaitu

1. Kebutuhan laporan data kemiskinan Kota Cimahi yang up to date, akurat, dan informatif;
2. Kebutuhan laporan penyebaran bantuan Kota Cimahi yang tepat, cepat, dan akurat;
3. Kebutuhan interface yang menarik, interactive, dan mudah digunakan.

# **SYSTEM NEEDS**

Daftar kebutuhan aplikasi Intervensi Kemiskinan 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;
5. Pembatasan hak akses penggunaan aplikasi.

# **INTERFACE SPESIFICATION**

Spesifikasi antarmuka dalam aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi adalah sebagai berikut:

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

# **PERANCANGAN**

Pada bagian perancangan ini akan dijelaskan lebih terperinci mengenai rancangan aplikasi Intervensi Kemiskinan 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 Intervensi Kemiskinan Pemerintah Kota Cimahi dapat digambarkan kedalam sebuah bagan aplikasi sebagai berikut:



Gambar 1. Arsitektur Aplikasi Intervensi Kemiskinan 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 Intervensi Kemiskinan Pemerintah Kota Cimahi merupakan aplikasi yang berbasis web yang dapat diakses melalui semua platform yang ada. Baik pada desktop, laptop, smartphone, maupun tablet.

Aplikasi Intervensi Kemiskinan 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) MySQL yang merupakan basis data utama aplikasi Intervensi Kemiskinan 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 Intervensi Kemiskinan Pemerintah Kota Cimahi dapat digambarkan dalam sebuah diagram konteks sebagai berikut:



Gambar 3. Diagram Konteks Aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi

Dalam aplikasi ini, ada 2 jenis hak akses yaitu admin dan pengguna. Admin memiliki hak akses untuk mengelola semua menu yang terdapat pada aplikasi, termasuk mengelola pengguna yang diijinkan masuk pada aplikasi. Sedangkan pengguna hanya memiliki hak akses beberapa menu seperti program kegiatan, penerima bantuan, dan rekap laporan.

# **DATA FLOW DIAGRAM**

Aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi terbagi dalam 11 modul utama yaitu proses login, pemrosesan pengguna, proses komponen, proses komponen sub, proses komponen sub detail, proses kelurahan, proses kecamatan, proses kota, proses provinsi, proses program kegiatan, pemrosesan rekap laporan. Setiap proses membutuhkan dukungan data penyimpanan dalam pemrosesannya.



Gambar 4. Diagram Flow Diagram Level 1

## Login

Modul Login digunakan untuk melakukan kontrol terhadap pengguna aplikasi. Hanya pengguna yang telah terdaftar dalam aplikasi yang dapat mengakses aplikasi Intervensi Kemiskinan Pemerintah Kota Cimahi.



Gambar 5. Data Flow Diagram Level 2 (Login)

## Pemrosesan Pengguna

Modul Pemrosesan Pengguna adalah modul yang digunakan Admin untuk mengelola pengguna aplikasi.



Gambar 6. Diagram Flow Diagram Level 2 (Pemrosesan Pengguna)

## Proses Komponen

Komponen merupakan jenis bantuan utama terdiri dari komponen arsitektur, komponen ekonomi, dan komponen sosial. Gambar 7 merupakan detil dari pemrosesan komponen, dimana terdapat empat proses yang dilakukan oleh admin yang memproses data komponen.



Gambar 7. Diagram Flow Diagram Level 2 (Pemrosesan Komponen)

## Proses Sub Komponen

Sub Komponen ialah jenis bantuan yang merupakan penjabaran dari komponen. Gambar 8 merupakan detil dari pemrosesan sub komponen, dimana terdapat empat proses yang dilakukan oleh admin yang memproses data sub komponen.



Gambar 8. Diagram Flow Diagram Level 2 (Pemrosesan Sub Komponen)

## Proses Detail Sub Komponen

Detail Sub Komponen ialah jenis bantuan yang merupakan penjabaran dari sub komponen. Gambar 9 merupakan pemrosesan detail sub komponen, dimana terdapat empat proses yang dilakukan oleh admin yang memproses data detail sub komponen.



Gambar 9. Diagram Flow Diagram Level 2 (Pemrosesan Detail Sub Komponen)

## Proses Kelurahan

Gambar 10 merupakan pemrosesan kelurahan, dimana terdapat empat proses yang dilakukan oleh admin yang memproses data kelurahan.



Gambar 10. Diagram Flow Diagram Level 2 (Pemrosesan Kelurahan)

## Proses Kecamatan

Gambar 11 merupakan pemrosesan kecamatan, dimana terdapat empat proses yang dilakukan oleh admin yang memproses data kecamatan.



Gambar 11. Diagram Flow Diagram Level 2 (Pemrosesan Kecamatan)

## Proses Kota

Gambar 12 merupakan pemrosesan kota, dimana terdapat empat proses yang dilakukan oleh admin yang memproses data kota.



Gambar 12. Diagram Flow Diagram Level 2 (Pemrosesan Kota)

## Proses Provinsi

Gambar 13 merupakan pemrosesan provinsi, dimana terdapat empat proses yang dilakukan oleh admin yang memproses data provinsi.



Gambar 13. Diagram Flow Diagram Level 2 (Pemrosesan Provinsi)

## Proses Kegiatan

Pada Proses kegiatan ini, admin dan pengguna dapat mengelola data kegiatan. Kemudian admin dan pengguna dapat mengelola penerima bantuan dari kegiatan yang ditambahkan.



Gambar 14. Diagram Flow Diagram Level 2 (Pemrosesan Kegiatan)

# **DESAIN BASIS DATA**

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



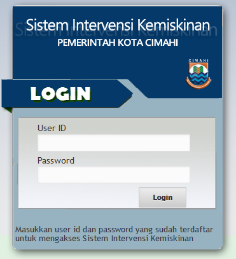
Gambar 15. Desain Basis Data

# **IMPLEMENTASI**

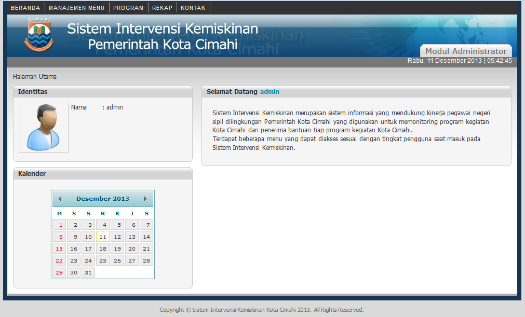
## Implementasi User Interface

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

1. Login



Gambar 16. Halaman Login

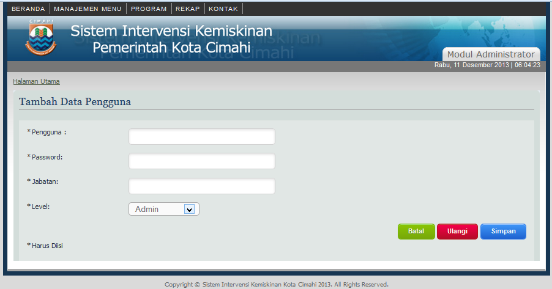
1. Halaman Utama

Gambar 17. Halaman Utama Aplikasi

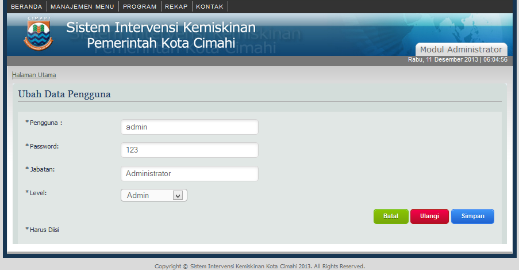
1. Halaman Manajemen Pengguna



Gambar 18. Tampilan Data Pengguna

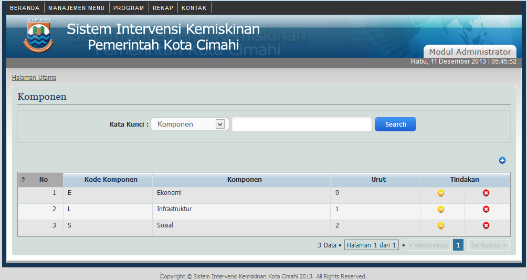


Gambar 19. Tampilan Form Tambah Pengguna

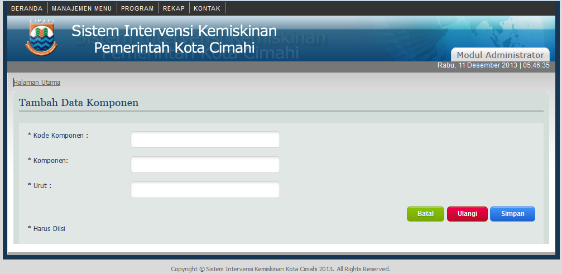


Gambar 20. Tampilan Form Ubah Data Pengguna

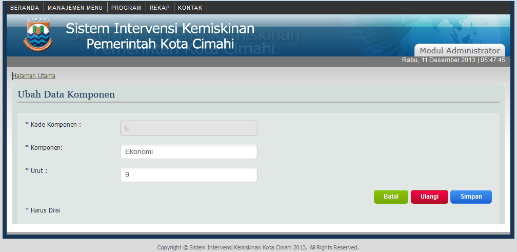
1. Halaman Manajemen Komponen



Gambar 21. Form Tampilan Data Komponen



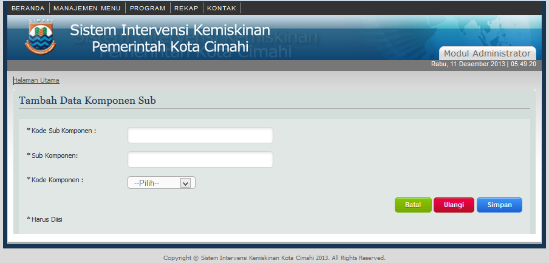
Gambar 22. Tampilan Form Tambah Komponen



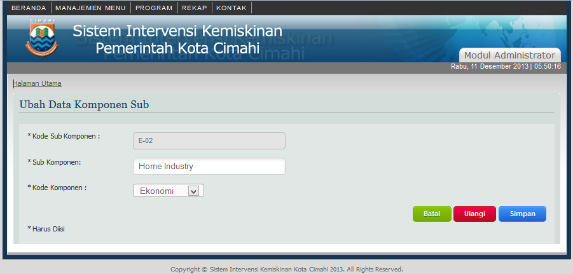
Gambar 23. Tampilan Form Ubah Data Komponen

1. Halaman Manajemen Sub Komponen

Gambar 24. Tampilan Data Sub Komponen



Gambar 25. Tampilan Form Tambah Sub Komponen

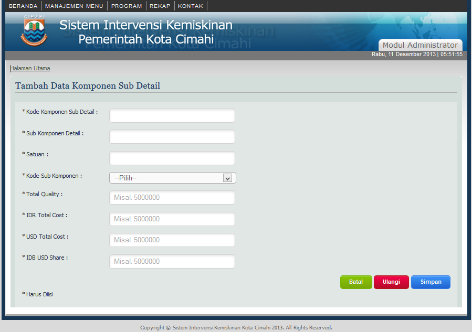


Gambar 26. Tampilan Form Ubah Sub Komponen

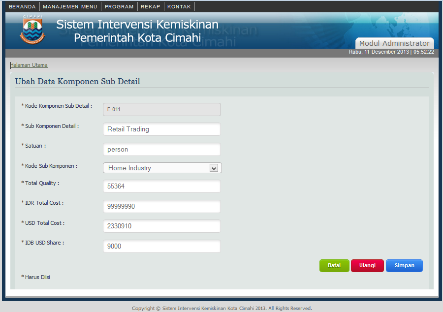
1. Halaman Manajemen Detail Sub Komponen



Gambar 27. Tampilan Data Detail Sub Komponen



Gambar 28. Tampilan Form Tambah Detail Sub Komponen

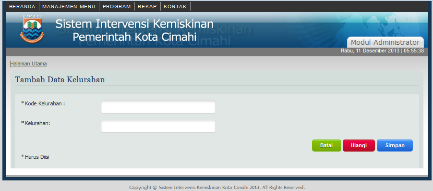


Gambar 29. Tampilan Form Ubah Data Detail Sub Komponen

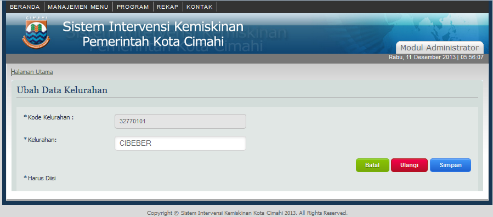
1. Halaman Manajemen Kelurahan



Gambar 30. Tampilan Data Kelurahan



Gambar 31. Tampilan Form Tambah Kelurahan



Gambar 32. Tampilan Form Ubah Kelurahan

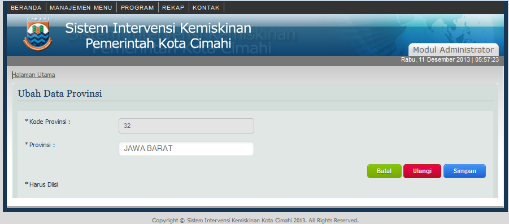
1. Halaman Manajemen Provinsi



Gambar 33. Tampilan Data Provinsi



Gambar 34. Tampilan Form Tambah Provinsi

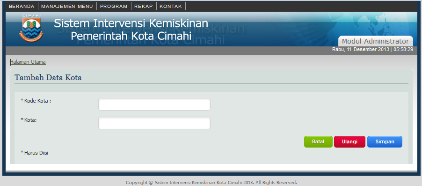


Gambar 35. Tampilan Form Ubah Data Provinsi

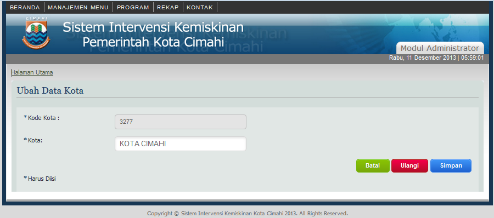
1. Halaman Manajemen Kota



Gambar 36. Tampilan Data Kota



Gambar 37. Tampilan Form Tambah Kota

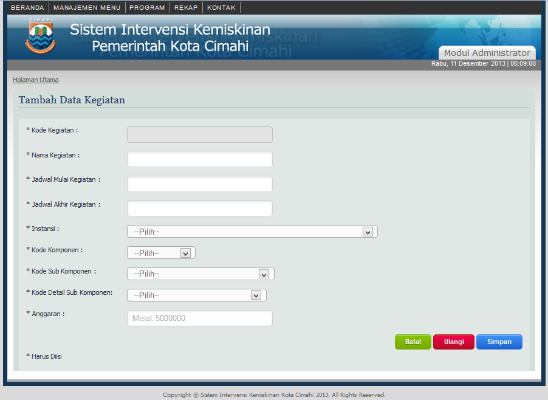


Gambar 38. Tampilan Form Ubah Data Kota

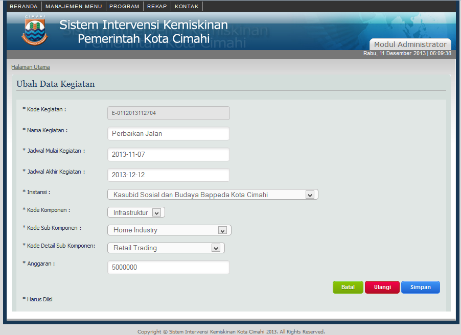
1. Halaman Manajemen Kegiatan Admin



Gambar 39. Tampilan Data Kegiatan Admin



Gambar 40. Tampilan Form Tambah Kegiatan Admin

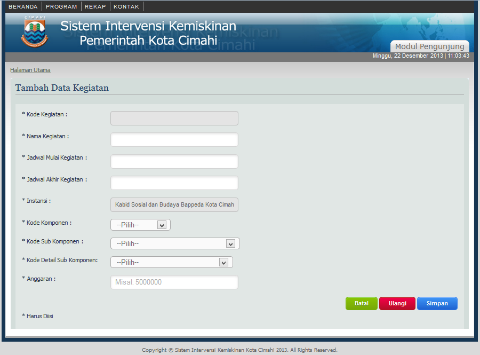


Gambar 41. Tampilan Form Ubah Kegiatan Admin

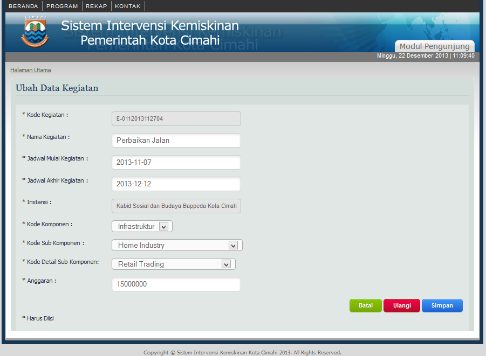
1. Halaman Manajemen Kegiatan Pengguna

****

Gambar 42. Tampilan Data Kegiatan Pengguna



Gambar 43. Tampilan Form Tambah Kegiatan Pengguna

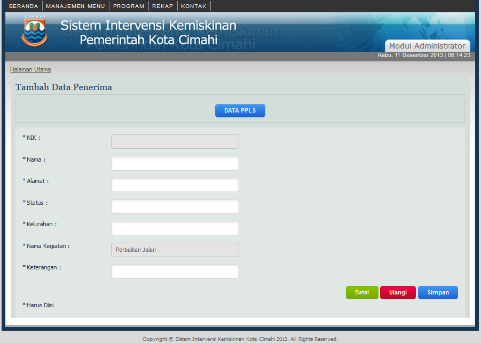


Gambar 44. Tampilan Form Ubah Kegiatan Pengguna

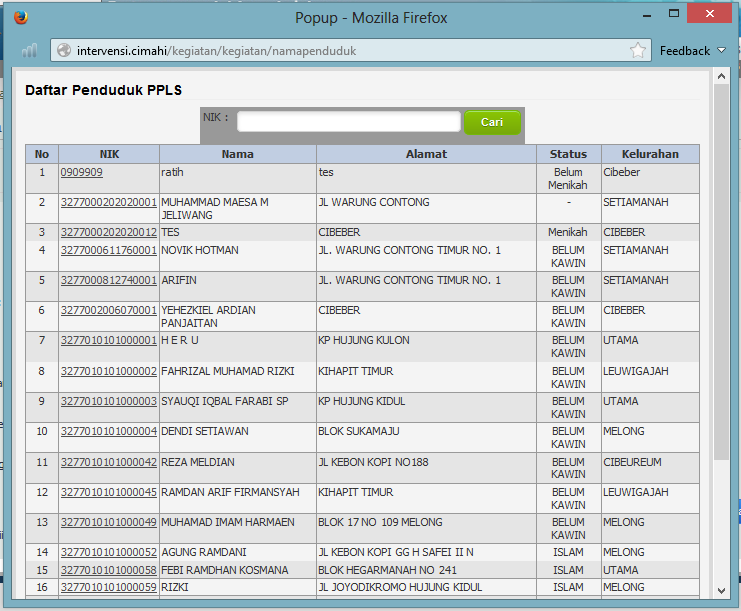
1. Halaman Manajemen Penerima Bantuan



Gambar 45. Tampilan Menu Penerima Berdasarkan Kegiatan yang Dipilih



Gambar 46. Tampilan Form Tambah Penerima



Gambar 47. Tampilan Popoup Penduduk PPLS untuk Tambah Penerima

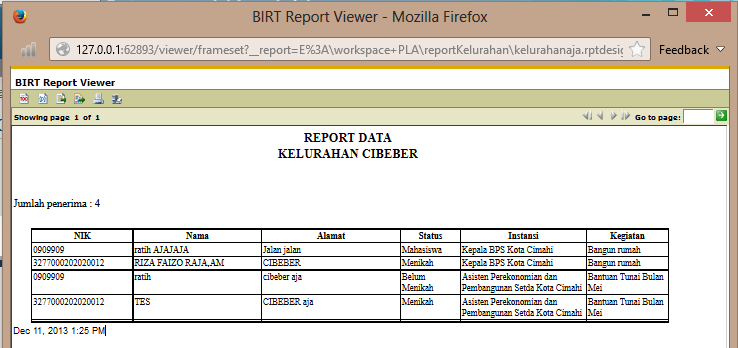


Gambar 48. Tampilan Form Ubah Data Penerima

1. Rekap Kelurahan

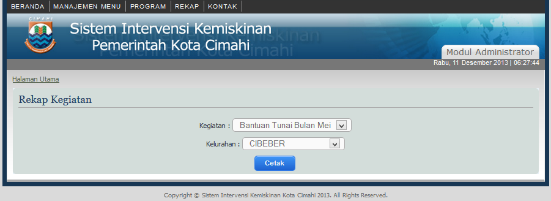


Gambar 49. Tampilan Rekap Kelurahan

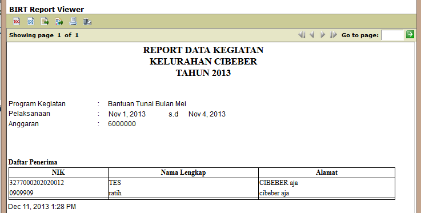


Gambar 50. Tampilan Hasil Rekap Kelurahan

1. Rekap Kegiatan

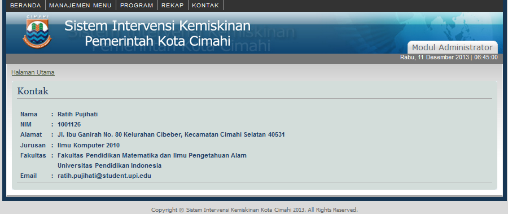


Gambar 51.Tampilan Rekap Kegiatan



Gambar 52. Tampilan Hasil rekap Kegiatan

1. Halaman Kontak



Gambar 53. Tampilan Kontak

# **IMPLEMENTASI SISTEM**

Aplikasi Intervensi Kemiskinan 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. Librari 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 MySQL 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 Intervensi Kemiskinan Pemerintah Kota Cimahi menggunakan MySQL sebagai mesin penyimpanan data. Berikut ini detail informasi yang berkaitan dengan struktur data MySQL yang digunakan oleh aplikasi Intervensi Kemiskinan 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