LAMPIRAN

Source code controller jadwal acara

```
class JadwalAcara extends CActiveRecord
{
        public $kategori_s;
        public $lokasi_s;
        /**
         * @return string the associated dat   abase
table name
         */
        public function tableName()
        {
                return 'jadwal_acara';
        }
        public function getKategori() {
                $data = array ();
                foreach      (Kategori::model()-
>findAll() as $value) {
                        $data[]    =    $value-
>nama;
                }
                return $data;
        }

        public function getLokasi() {
                $data = array ();
                foreach       (Lokasi::model()-
>findAll() as $value) {
                        $data[]    =    $value-
>nama;
                }
                return $data;
        }
        /**
         * @return array validation rules for model
attributes.
         */
        public function rules()
        {
                // NOTE: you should only define
rules for those attributes that
                // will receive user inputs.
                return array(
                        array('kategori, nama,
deskripsi, fasilitas, lokasi, waktu, pendaftaran,
gambar', 'required'),
                        array('kategori,
lokasi', 'numerical', 'integerOnly'=>true),
```

Source code controller Kategori

```
class KategoriController extends Controller
{
        /**
         * @var string the default layout for the
views. Defaults to '//layouts/column2', meaning
         *   using   two-column   layout.   See
'protected/views/layouts/column2.php'.
         */
        public $layout='//layouts/column2';

        /**
         * @return array action filters
         */
        public function filters()
        {
                return array(
                        'accessControl',      //
perform access control for CRUD operations
                        'postOnly + delete', //
we only allow deletion via POST request
                );
        }

        /**
         * Specifies the access control rules.
         *  This   method   is   used   by   the
'accessControl' filter.
         * @return array access control rules
         */
        public function accessRules()
        {
                return array(
                        array('allow',  // allow
all users to perform 'index' and 'view' actions

        'actions'=>array('index','view'),

        'users'=>array('*'),
                        ),
                        array('allow', // allow
authenticated user to perform 'create' and 'update'
actions

        'actions'=>array('create','update'),

        'users'=>array('@'),
                        ),
                        array('allow', // allow
admin user to perform 'admin' and 'delete' actions

        'actions'=>array('admin','delete'),

        'users'=>array('@'),
                        ),
                              array('nama', 'length',
'max'=>150),
                              array('pendaftaran',
'length', 'max'=>30),
                              array('gambar',
'length', 'max'=>50),
                              array('gambar', 'file',
'types'=>'jpg, gif, png, jpeg', 'allowEmpty'=>false),
                              // The following rule is
used by search().
                              // @todo Please
remove those attributes that should not be searched.
                              array('id', kategori,
nama, deskripsi, fasilitas, lokasi, waktu, pendaftaran,
gambar, kategori_s, lokasi_s', 'safe', 'on'=>'search'),
                    );
          }

          /**
           * @return array relational rules.
           */
          public function relations()
          {
                    // NOTE: you may need to adjust
the relation name and the related
                    // class name for the relations
automatically generated below.
                    return array(
                              'kategori0' =>
array(self::BELONGS_TO, 'Kategori', 'kategori'),
                              'lokasi0' =>
array(self::BELONGS_TO, 'Lokasi', 'lokasi'),
                              'pesertas' =>
array(self::HAS_MANY, 'Peserta', 'id_acara'),
                    );
          }

          /**
           * @return array customized attribute
labels (name=>label)
           */
          public function attributeLabels()
          {
                    return array(
                              'id' => 'ID',
                              'kategori' =>
'Kategori',
                              'kategori_s' =>
'Kategori',
                              'nama' => 'Nama',
                              'deskripsi' =>
'Deskripsi',
                              'fasilitas' => 'Fasilitas',
                              'lokasi' => 'Lokasi',
```

```
                              array('deny', // deny
all users

          'users'=>array('*'),
                              ),
                    );
          }

          /**
           * Displays a particular model.
           * @param integer $id the ID of the model
to be displayed
           */
          public function actionView($id)
          {
                    $this->render('view',array(
                              'model'=>$this-
>loadModel($id),
                    ));
          }

          /**
           * Creates a new model.
           * If creation is successful, the browser
will be redirected to the 'view' page.
           */
          public function actionCreate()
          {
                    $model=new Kategori;

                    // Uncomment the following
line if AJAX validation is needed
                    // $this-
>performAjaxValidation($model);

                    if(isset($_POST['Kategori']))
                    {
                              $model-
>attributes=$_POST['Kategori'];
                              if($model->save())
                                        $this-
>redirect(array('view','id'=>$model->id));
                    }

                    $this->render('create',array(
                              'model'=>$model,
                    ));
          }

          /**
           * Updates a particular model.
           * If update is successful, the browser will
be redirected to the 'view' page.
           * @param integer $id the ID of the model
to be updated
           */
          public function actionUpdate($id)
          {
                    $model=$this-
>loadModel($id);
```

```
			'lokasi_s' => 'Lokasi',
			'waktu' => 'Waktu',
			'pendaftaran' => 'Pendaftaran',
			'gambar' => 'Gambar',
		);
	}

	/**
	 * Retrieves a list of models based on the current search/filter conditions.
	 *
	 * Typical usecase:
	 * - Initialize the model fields with values from filter form.
	 * - Execute this method to get CActiveDataProvider instance which will filter
	 * models according to data in model fields.
	 * - Pass data provider to CGridView, CListView or any similar widget.
	 *
	 * @return CActiveDataProvider the data provider that can return the models
	 * based on the search/filter conditions.
	 */
	public function search()
	{
		// @todo Please modify the following code to remove attributes that should not be searched.

		$criteria=new CDbCriteria;

		$criteria->compare('id',$this->id);
		$criteria->compare('kategori',$this->kategori);
		$criteria->compare('nama',$this->nama,true);
		$criteria->compare('deskripsi',$this->deskripsi,true);
		$criteria->compare('fasilitas',$this->fasilitas,true);
		$criteria->compare('lokasi',$this->lokasi);
		$criteria->compare('waktu',$this->waktu,true);
		$criteria->compare('pendaftaran',$this->pendaftaran,true);
		$criteria->compare('gambar',$this->gambar,true);
```

```
		// Uncomment the following line if AJAX validation is needed
		// $this->performAjaxValidation($model);

		if(isset($_POST['Kategori']))
		{
			$model->attributes=$_POST['Kategori'];
			if($model->save())
				$this->redirect(array('view','id'=>$model->id));
		}

		$this->render('update',array(
			'model'=>$model,
		));
	}

	/**
	 * Deletes a particular model.
	 * If deletion is successful, the browser will be redirected to the 'admin' page.
	 * @param integer $id the ID of the model to be deleted
	 */
	public function actionDelete($id)
	{
		$this->loadModel($id)->delete();

		// if AJAX request (triggered by deletion via admin grid view), we should not redirect the browser
		if(!isset($_GET['ajax']))
			$this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl'] : array('admin'));
	}

	/**
	 * Lists all models.
	 */
	public function actionIndex()
	{
		$dataProvider=new CActiveDataProvider('Kategori');
		$this->render('index',array(
	'dataProvider'=>$dataProvider,
		));
	}

	/**
	 * Manages all models.
	 */
	public function actionAdmin()
	{
		$model=new Kategori('search');
		$model->unsetAttributes();  // clear any default values
		return new
CActiveDataProvider($this, array(
			'criteria'=>$criteria,
		));
	}

	public function search2()
	{
		// @todo Please modify the
following code to remove attributes that should not
be searched.

		$criteria=new CDbCriteria;
		$criteria->with = array(
'kategori0' );
		$criteria->with = array( 'lokasi0'
);

		$criteria->compare('id',$this-
>id);
		$criteria-
>compare('kategori',$this->kategori);
		$criteria-
>compare('nama',$this->nama,true);
		$criteria-
>compare('deskripsi',$this->deskripsi,true);
		$criteria-
>compare('fasilitas',$this->fasilitas,true);
		$criteria-
>compare('lokasi',$this->lokasi);
		$criteria-
>compare('waktu',$this->waktu,true);
		$criteria-
>compare('pendaftaran',$this->pendaftaran,true);
		$criteria-
>compare('gambar',$this->gambar,true);
		$criteria->compare(
'kategori0.nama', $this->kategori_s, true );
		$criteria->compare(
'lokasi0.nama', $this->lokasi_s, true );

		return new
CActiveDataProvider($this, array(
			'criteria'=>$criteria,
			'sort'=>array(

	'attributes'=>array(

	'kategori_s'=>array(

	'asc'=>'kategori0.nama',

	'desc'=>'kategori0.nama DESC',
								),
```

```
		if(isset($_GET['Kategori']))
			$model-
>attributes=$_GET['Kategori'];

		$this->render('admin',array(
			'model'=>$model,
		));
	}

	/**
	 * Returns the data model based on the
primary key given in the GET variable.
	 * If the data model is not found, an HTTP
exception will be raised.
	 * @param integer $id the ID of the model
to be loaded
	 * @return Kategori the loaded model
	 * @throws CHttpException
	 */
	public function loadModel($id)
	{
		$model=Kategori::model()-
>findByPk($id);
		if($model===null)
			throw new
CHttpException(404,'The requested page does not
exist.');
		return $model;
	}

	/**
	 * Performs the AJAX validation.
	 * @param Kategori $model the model to
be validated
	 */
	protected function
performAjaxValidation($model)
	{
		if(isset($_POST['ajax']) &&
$_POST['ajax']==='kategori-form')
		{
			echo
CActiveForm::validate($model);
			Yii::app()->end();
		}
	}
}
```

Source code controller lokasi

```
Class LokasiController extends Controller
{
	/**
	 * @var string the default layout for the
views. Defaults to '//layouts/column2', meaning
	 * using two-column layout. See
'protected/views/layouts/column2.php'.
	 */
	public $layout='//layouts/column2';

	/**
```

```
			'lokasi_s'=>array(

			'asc'=>'lokasi0.nama',

			'desc'=>'lokasi0.nama DESC',
							),

			'*',
						),
					),
				));
			}

			/**
			 * Returns the static model of the specified AR class.
			 * Please note that you should have this exact method in all your CActiveRecord descendants!
			 * @param string $className active record class name.
			 * @return JadwalAcara the static model class
			 */
			public static function model($className=__CLASS__)
			{
				return parent::model($className);
			}
}
```

Source code controller peserta

```
class PesertaController extends Controller
{
	/**
	 * @var string the default layout for the views. Defaults to '//layouts/column2', meaning
	 * using two-column layout. See 'protected/views/layouts/column2.php'.
	 */
	public $layout='//layouts/column2';

	/**
	 * @return array action filters
	 */
	public function filters()
	{
		return array(
			'accessControl', // perform access control for CRUD operations
			'postOnly + delete', // we only allow deletion via POST request
		);
	}

	/**
	 * Specifies the access control rules.
	 * This method is used by the 'accessControl' filter.
	 * @return array access control rules
	 */
	public function accessRules()
	{
		return array(
			array('allow',  // allow all users to perform 'index' and 'view' actions

	'actions'=>array('index','view'),

	'users'=>array('*'),
			),
			array('allow', // allow authenticated user to perform 'create' and 'update' actions

	'actions'=>array('create','update'),

	'users'=>array('@'),
			),
			array('allow', // allow admin user to perform 'admin' and 'delete' actions

	'actions'=>array('admin','delete'),

	'users'=>array('@'),
			),
			array('deny',  // deny all users

	'users'=>array('*'),
			),
		);
	}

	/**
	 * Displays a particular model.
	 * @param integer $id the ID of the model to be displayed
	 */
	public function actionView($id)
	{
		$this->render('view',array(
			'model'=>$this->loadModel($id),
		));
		);
	}

	/**
	 * Specifies the access control rules.
	 * This method is used by the 'accessControl' filter.
	 * @return array access control rules
	 */
	public function accessRules()
	{
		return array(
			array('allow',  // allow all users to perform 'index' and 'view' actions
	'actions'=>array('index','view'),
	'users'=>array('*'),
			),
			array('allow', // allow authenticated user to perform 'create' and 'update' actions
	'actions'=>array('create','update'),
	'users'=>array('@'),
			),
			array('allow', // allow admin user to perform 'admin' and 'delete' actions
	'actions'=>array('admin','delete'),
	'users'=>array('@'),
			),
			array('deny',  // deny all users
	'users'=>array('*'),
			),
		);
	}

	/**
	 * Displays a particular model.
	 * @param integer $id the ID of the model to be displayed
	 */
	public function actionView($id)
	{
		$this->render('view',array(
			'model'=>$this->loadModel($id),
		));
	}

	/**
	 * Creates a new model.
	 * If creation is successful, the browser will be redirected to the 'view' page.
	 */
	public function actionCreate()
	{
		$model=new Lokasi;

		// Uncomment the following line if AJAX validation is needed
		// $this->performAjaxValidation($model);

		if(isset($_POST['Lokasi']))
		{
			$model->attributes=$_POST['Lokasi'];
			if($model->save())
				$this->redirect(array('view','id'=>$model->id));
		}

		$this->render('create',array(
			'model'=>$model,
		));
	}

	/**
	 * Updates a particular model.
	 * If update is successful, the browser will be redirected to the 'view' page.
	 * @param integer $id the ID of the model to be updated
	 */
	public function actionUpdate($id)
	{
		$model=$this->loadModel($id);

		// Uncomment the following line if AJAX validation is needed
		// $this->performAjaxValidation($model);

		if(isset($_POST['Lokasi']))
		{
			$model->attributes=$_POST['Lokasi'];
			if($model->save())
				$this->redirect(array('view','id'=>$model->id));
		}

		$this->render('update',array(
			'model'=>$model,
		));
	}

	/**
	/**
	 * Creates a new model.
	 * If creation is successful, the browser will
be redirected to the 'view' page.
	 */
	public function actionCreate()
	{
		$model=new Peserta;

		// Uncomment the following line
if AJAX validation is needed
		//                    $this-
>performAjaxValidation($model);

		if(isset($_POST['Peserta']))
		{
			$model-
>attributes=$_POST['Peserta'];
			if($model->save())
				$this-
>redirect(array('view','id'=>$model->id));
		}

		$this->render('create',array(
			'model'=>$model,
		));
	}

	/**
	 * Updates a particular model.
	 * If update is successful, the browser will
be redirected to the 'view' page.
	 * @param integer $id the ID of the model
to be updated
	 */
	public function actionUpdate($id)
	{
		$model=$this->loadModel($id);

		// Uncomment the following line
if AJAX validation is needed
		//                    $this-
>performAjaxValidation($model);

		if(isset($_POST['Peserta']))
		{
			$model-
>attributes=$_POST['Peserta'];
			if($model->save())
				$this-
>redirect(array('view','id'=>$model->id));
		}
```

```
	 * Deletes a particular model.
	 * If deletion is successful, the browser
will be redirected to the 'admin' page.
	 * @param integer $id the ID of the model
to be deleted
	 */
	public function actionDelete($id)
	{
		$this->loadModel($id)-
>delete();

		// if AJAX request (triggered by
deletion via admin grid view), we should not
redirect the browser
		if(!isset($_GET['ajax']))
			$this-
>redirect(isset($_POST['returnUrl'])            ?
$_POST['returnUrl'] : array('admin'));
	}

	/**
	 * Lists all models.
	 */
	public function actionIndex()
	{
		$dataProvider=new
CActiveDataProvider('Lokasi');
		$this->render('index',array(

	'dataProvider'=>$dataProvider,
		));
	}

	/**
	 * Manages all models.
	 */
	public function actionAdmin()
	{
		$model=new Lokasi('search');
		$model->unsetAttributes();    //
clear any default values
		if(isset($_GET['Lokasi']))
			$model-
>attributes=$_GET['Lokasi'];

		$this->render('admin',array(
			'model'=>$model,
		));
	}

	/**
	 * Returns the data model based on the
primary key given in the GET variable.
	 * If the data model is not found, an HTTP
exception will be raised.
	 * @param integer $id the ID of the model
to be loaded
	 * @return Lokasi the loaded model
	 * @throws CHttpException
	 */
	public function loadModel($id)
	{
		$this->render('update',array(
			'model'=>$model,
		));
	}

	/**
	 * Deletes a particular model.
	 * If deletion is successful, the browser will
be redirected to the 'admin' page.
	 * @param integer $id the ID of the model
to be deleted
	 */
	public function actionDelete($id)
	{
		$this->loadModel($id)-
>delete();

		// if AJAX request (triggered by
deletion via admin grid view), we should not redirect
the browser
		if(!isset($_GET['ajax']))
			$this-
>redirect(isset($_POST['returnUrl'])	?
$_POST['returnUrl'] : array('admin'));
	}

	/**
	 * Lists all models.
	 */
	public function actionIndex()
	{
		$dataProvider=new
CActiveDataProvider('Peserta');
		$this->render('index',array(

	'dataProvider'=>$dataProvider,
		));
	}

	/**
	 * Manages all models.
	 */
	public function actionAdmin()
	{
		$model=new Peserta('search');
		$model->unsetAttributes();	//
clear any default values
		if(isset($_GET['Peserta']))
			$model-
>attributes=$_GET['Peserta'];

		$this->render('admin',array(
			'model'=>$model,
		));
```

```
		$model=Lokasi::model()-
>findByPk($id);
		if($model===null)
			throw		new
CHttpException(404,'The requested page does not
exist.');
		return $model;
	}

	/**
	 * Performs the AJAX validation.
	 * @param Lokasi $model the model to be
validated
	 */
	protected		function
performAjaxValidation($model)
	{
		if(isset($_POST['ajax'])	&&
$_POST['ajax']==='lokasi-form')
		{
			echo
CActiveForm::validate($model);
			Yii::app()->end();
		}
	}
}
```

Source code controller petugas

```
<?php

class PetugasController extends Controller
{
	/**
	 * @var string the default layout for the
views. Defaults to '//layouts/column2', meaning
	 * using two-column layout. See
'protected/views/layouts/column2.php'.
	 */
	public $layout='//layouts/column2';

	/**
	 * @return array action filters
	 */
	public function filters()
	{
		return array(
			'accessControl',	//
perform access control for CRUD operations
			'postOnly + delete', //
we only allow deletion via POST request
		);
	}

	/**
	 * Specifies the access control rules.
	 * This method is used by the
'accessControl' filter.
	 * @return array access control rules
	 */
	public function accessRules()
	{
	}

	/**
	 * Returns the data model based on the primary key given in the GET variable.
	 * If the data model is not found, an HTTP exception will be raised.
	 * @param integer $id the ID of the model to be loaded
	 * @return Peserta the loaded model
	 * @throws CHttpException
	 */
	public function loadModel($id)
	{
		$model=Peserta::model()->findByPk($id);
		if($model===null)
			throw new CHttpException(404,'The requested page does not exist.');
		return $model;
	}

	/**
	 * Performs the AJAX validation.
	 * @param Peserta $model the model to be validated
	 */
	protected function performAjaxValidation($model)
	{
		if(isset($_POST['ajax']) && $_POST['ajax']==='peserta-form')
		{
			echo CActiveForm::validate($model);
			Yii::app()->end();
		}
	}
}
```

Source code controller site

```
<?php

class SiteController extends Controller
{
	/**
	 * Declares class-based actions.
	 */
	public function actions()
	{
		return array(
```

```
		return array(
			array('allow',  // allow all users to perform 'index' and 'view' actions
				'actions'=>array('index','view'),
				'users'=>array('*'),
			),
			array('allow', // allow authenticated user to perform 'create' and 'update' actions
				'actions'=>array('create','update'),
				'users'=>array('*'),
			),
			array('allow', // allow admin user to perform 'admin' and 'delete' actions
				'actions'=>array('admin','delete'),
				'users'=>array('*'),
			),
			array('deny',  // deny all users
				'users'=>array('*'),
			),
		);
	}

	/**
	 * Displays a particular model.
	 * @param integer $id the ID of the model to be displayed
	 */
	public function actionView($id)
	{
		$this->render('view',array(
			'model'=>$this->loadModel($id),
		));
	}

	/**
	 * Creates a new model.
	 * If creation is successful, the browser will be redirected to the 'view' page.
	 */
	public function actionCreate()
	{
		$model=new Petugas;

		// Uncomment the following line if AJAX validation is needed
		// $this->performAjaxValidation($model);

		if(isset($_POST['Petugas']))
		{
```

```
                        // captcha action
renders the CAPTCHA image displayed on the
contact page
                        'captcha'=>array(

        'class'=>'CCaptchaAction',

        'backColor'=>0xFFFFFF,
                        ),
                        // page action renders
"static" pages stored under
'protected/views/site/pages'
                        // They can be
accessed via:
index.php?r=site/page&view=FileName
                        'page'=>array(

        'class'=>'CViewAction',
                        ),
                );
        }

        /**
         * This is the default 'index' action that is
invoked
         * when an action is not explicitly
requested by users.
         */
        public function actionIndex()
        {
                // renders the view file
'protected/views/site/index.php'
                // using the default layout
'protected/views/layouts/main.php'
                $this->render('index');
        }

        /**
         * This is the action to handle external
exceptions.
         */
        public function actionError()
        {
                if($error=Yii::app()-
>errorHandler->error)
                {
                        if(Yii::app()-
>request->isAjaxRequest)
                                echo
$error['message'];
                        else
                                $this-
>render('error', $error);
                }
```

```
                        $model-
>attributes=$_POST['Petugas'];
                        if($model->save())
                                $this-
>redirect(array('view','id'=>$model->id));
                }

                $this->render('create',array(
                        'model'=>$model,
                ));
        }

        /**
         * Updates a particular model.
         * If update is successful, the browser will
be redirected to the 'view' page.
         * @param integer $id the ID of the model
to be updated
         */
        public function actionUpdate($id)
        {
                $model=$this-
>loadModel($id);

                // Uncomment the following
line if AJAX validation is needed
                //                    $this-
>performAjaxValidation($model);

                if(isset($_POST['Petugas']))
                {
                        $model-
>attributes=$_POST['Petugas'];
                        if($model->save())
                                $this-
>redirect(array('view','id'=>$model->id));
                }

                $this->render('update',array(
                        'model'=>$model,
                ));
        }

        /**
         * Deletes a particular model.
         * If deletion is successful, the browser
will be redirected to the 'admin' page.
         * @param integer $id the ID of the model
to be deleted
         */
        public function actionDelete($id)
        {
                $this->loadModel($id)-
>delete();

                // if AJAX request (triggered by
deletion via admin grid view), we should not
redirect the browser
                if(!isset($_GET['ajax']))
                        $this-
>redirect(isset($_POST['returnUrl'])            ?
$_POST['returnUrl'] : array('admin'));
```

```
	}

	/**
	 * Displays the contact page
	 */
	public function actionContact()
	{
		$model=new ContactForm;

	if(isset($_POST['ContactForm']))
		{
			$model-
>attributes=$_POST['ContactForm'];
			if($model-
>validate())
			{

	$name='=?UTF-
8?B?'.base64_encode($model->name).'?=';

	$subject='=?UTF-
8?B?'.base64_encode($model->subject).'?=';

	$headers="From:   $name   <{$model-
>email}>\r\n".

	"Reply-To: {$model->email}\r\n".

	"MIME-Version: 1.0\r\n".

	"Content-Type: text/plain; charset=UTF-
8";


	mail(Yii::app()-
>params['adminEmail'],$subject,$model-
>body,$headers);
				Yii::app()-
>user->setFlash('contact','Thank you for contacting
us. We will respond to you as soon as possible.');
				$this-
>refresh();
			}
		}
		$this-
>render('contact',array('model'=>$model));
	}

	/**
	 * Displays the login page
	 */
	public function actionLogin()
	{
		$model=new LoginForm;
```

```
	}

	/**
	 * Lists all models.
	 */
	public function actionIndex()
	{
		$dataProvider=new
CActiveDataProvider('Petugas');
		$this->render('index',array(

	'dataProvider'=>$dataProvider,
		));
	}

	/**
	 * Manages all models.
	 */
	public function actionAdmin()
	{
		$model=new Petugas('search');
		$model->unsetAttributes();   //
clear any default values
		if(isset($_GET['Petugas']))
			$model-
>attributes=$_GET['Petugas'];

		$this->render('admin',array(
			'model'=>$model,
		));
	}

	/**
	 * Returns the data model based on the
primary key given in the GET variable.
	 * If the data model is not found, an HTTP
exception will be raised.
	 * @param integer $id the ID of the model
to be loaded
	 * @return Petugas the loaded model
	 * @throws CHttpException
	 */
	public function loadModel($id)
	{
		$model=Petugas::model()-
>findByPk($id);
		if($model===null)
			throw            new
CHttpException(404,'The requested page does not
exist.');
		return $model;
	}

	/**
	 * Performs the AJAX validation.
	 * @param Petugas $model the model to
be validated
	 */
	protected                function
performAjaxValidation($model)
	{
```

```
                    // if it is ajax validation request
                    if(isset($_POST['ajax'])        &&
$_POST['ajax']==='login-form')
                    {
                              echo
CActiveForm::validate($model);
                              Yii::app()->end();
                    }

                    // collect user input data
                    if(isset($_POST['LoginForm']))
                    {
                              $model-
>attributes=$_POST['LoginForm'];
                              // validate user input
and redirect to the previous page if valid
                              if($model->validate()
&& $model->login())
                                        $this-
>redirect(Yii::app()->user->returnUrl);
                    }
                    // display the login form
                    $this-
>render('login',array('model'=>$model));
          }

          /**
           * Logs out the current user and redirect to
homepage.
           */
          public function actionLogout()
          {
                    Yii::app()->user->logout();
                    $this->redirect(Yii::app()-
>homeUrl);
          }
}
```

Source code controller user

```
<?php

class UserController extends Controller
{
          /**
           * @var string the default layout for the
views. Defaults to '//layouts/column2', meaning
           *   using   two-column   layout.   See
'protected/views/layouts/column2.php'.
           */
          public $layout='//layouts/column2';

          /**
```

```
                    if(isset($_POST['ajax'])      &&
$_POST['ajax']==='petugas-form')
                    {
                              echo
CActiveForm::validate($model);
                              Yii::app()->end();
                    }
          }
}


'actions'=>array('admin','delete'),

          'users'=>array('@'),
                              ),
                              array('deny',   // deny
all users

          'users'=>array('*'),
                              ),
                    );
          }

          /**
           * Displays a particular model.
           * @param integer $id the ID of the model
to be displayed
           */
          public function actionView($id)
          {
                    $this->render('view',array(
                              'model'=>$this-
>loadModel($id),
                    ));
          }

          /**
           * Creates a new model.
           * If creation is successful, the browser
will be redirected to the 'view' page.
           */
          public function actionCreate()
          {
                    $model=new User;

                    // Uncomment the following
line if AJAX validation is needed
                    //                    $this-
>performAjaxValidation($model);

                    if(isset($_POST['User']))
                    {
                              $model-
>attributes=$_POST['User'];
                              if($model->save())
```

```
	 * @return array action filters
	 */
	public function filters()
	{
		return array(
			'accessControl', // perform access control for CRUD operations
			'postOnly + delete', // we only allow deletion via POST request
		);
	}

	/**
	 * Specifies the access control rules.
	 * This method is used by the 'accessControl' filter.
	 * @return array access control rules
	 */
	public function accessRules()
	{
		return array(
			array('allow',  // allow all users to perform 'index' and 'view' actions
	'actions'=>array('index','view'),
	'users'=>array('*'),
			),
			array('allow', // allow authenticated user to perform 'create' and 'update' actions
	'actions'=>array('create','update'),
	'users'=>array('@'),
			),
			array('allow', // allow admin user to perform 'admin' and 'delete' actions

		$dataProvider=new CActiveDataProvider('User');
		$this->render('index',array(
	'dataProvider'=>$dataProvider,
		));
	}

	/**
	 * Manages all models.
	 */
	public function actionAdmin()
	{
		$model=new User('search');
				$this->redirect(array('view','id'=>$model->nim));
		}

		$this->render('create',array(
			'model'=>$model,
		));
	}

	/**
	 * Updates a particular model.
	 * If update is successful, the browser will be redirected to the 'view' page.
	 * @param integer $id the ID of the model to be updated
	 */
	public function actionUpdate($id)
	{
		$model=$this->loadModel($id);

		// Uncomment the following line if AJAX validation is needed
		// $this->performAjaxValidation($model);

		if(isset($_POST['User']))
		{
			$model->attributes=$_POST['User'];
			if($model->save())
				$this->redirect(array('view','id'=>$model->nim));
		}

		$this->render('update',array(
			'model'=>$model,
		));
	}

	/**
	 * Deletes a particular model.
	 * If deletion is successful, the browser will be redirected to the 'admin' page.
	 * @param integer $id the ID of the model to be deleted
	 */
	public function actionDelete($id)
	{
		$this->loadModel($id)->delete();
		$model->unsetAttributes();	// clear any default values
		if(isset($_GET['User']))
			$model->attributes=$_GET['User'];

		$this->render('admin',array(
			'model'=>$model,
		));
	}

	/**
	 * Returns the data model based on the primary key given in the GET variable.
	 * If the data model is not found, an HTTP exception will be raised.
	 * @param integer $id the ID of the model to be loaded
	 * @return User the loaded model
	 * @throws CHttpException
	 */
	public function loadModel($id)
	{
		$model=User::model()->findByPk($id);
		if($model===null)
			throw new CHttpException(404,'The requested page does not exist.');
		return $model;
	}

	/**
	 * Performs the AJAX validation.
	 * @param User $model the model to be validated
	 */
	protected function performAjaxValidation($model)
	{
		if(isset($_POST['ajax']) && $_POST['ajax']==='user-form')
		{
			echo CActiveForm::validate($model);
			Yii::app()->end();
		}
	}
}
```

```
		// if AJAX request (triggered by deletion via admin grid view), we should not redirect the browser
		if(!isset($_GET['ajax']))
			$this->redirect(isset($_POST['returnUrl']) ? $_POST['returnUrl'] : array('admin'));
	}

	/**
	 * Lists all models.
	 */
	public function actionIndex()
	{
```