

PERTEMUAN KE – 5

Transaksi Studi Kasus Sistem Penjualan

A. TUJUAN

1. Mampu mengimplementasikan Studi kasus transaksi penjualan, dan menciptakan Model, realasi antar objek menggunakan kelas CActiveRecord.
2. Mampu membuat aplikasi Interaktif menggunakan ajax dan JQuery
3. Mamhami cara penambahan metode dalam model

B. TEORI SINGKAT

Active Record (AR) adalah teknik populer Pemetaan Relasional Objek / *Object-Relational Mapping (ORM)*. Setiap kelas AR mewakili tabel database (atau view) yang atributnya diwakili oleh properti kelas AR, dan turunan AR mewakili rekaman pada tabel tersebut. Operasi umum CRUD diimplementasikan sebagai metode AR. Hasilnya, kita dapat mengakses data dengan cara lebih berorientasi-objek. Sebagai contoh, kita dapat menggunakan kode berikut untuk menyisipkan rekaman baru ke dalam tabel. Beberapa metode terdapat dalam CActiveRecord, untuk menambah dan mengubah rekaman `save()`, dan menghapus `delete()`, membaca `find`, `findAll()` dsb, yang akan dibahas sendiri dipertemuan berikutnya.

C. PRATIK

Transaksi Penjualan, dengan tabel barang, jenis_barang menggunakan pertemuan sebelumnya

1. Menciptakan tabel penjualan

```
CREATE TABLE penjualan (  
    id INT(11) NOT NULL AUTO_INCREMENT,  
    no_nota INT,  
    tanggal DATE DEFAULT NULL,  
    barang_id INT(11) DEFAULT NULL,  
    qty INT(11) DEFAULT NULL,  
    harga_saat_ini INT(11) DEFAULT NULL,  
    PRIMARY KEY (id),  
    KEY FK_penjualan (barang_id),  
    CONSTRAINT FK_penjualan FOREIGN KEY (barang_id)  
        REFERENCES barang (id) ON DELETE NO ACTION ON UPDATE CASCADE  
) ENGINE=INNODB;
```

2. Buatlah model dan CRUD Penjualan dengan gii generator
3. tambahkan di model Penjualan.php

Penjualan.php

```
<?php  
/**  
 * This is the model class for table "penjualan".
```

```

*
* The followings are the available columns in table 'penjualan':
* @property integer $id
* @property string $tanggal
* @property integer $barang_id
* @property integer $qty
* @property integer $harga_saat_ini
*
* The followings are the available model relations:
* @property Barang $barang
*/
class Penjualan extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'penjualan';
    }

    /**
     * @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('barang_id, qty, harga_saat_ini',
                'numerical', 'integerOnly'=>true),
            array('tanggal, no_nota', 'safe'),
            // The following rule is used by search().
            // @todo Please remove those attributes that
            // should not be searched.
            array('id, tanggal, barang_id, qty, harga_saat_ini',
                '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(
            'barang' => array(self::BELONGS_TO, 'Barang', 'barang_id'),
        );
    }

    /**
     * @return array customized attribute labels (name=>label)
     */
    public function attributeLabels()
    {
        return array(

```

```

        'id' => 'ID',
        'tanggal' => 'Tanggal',
        'barang_id' => 'KD',
        'qty' => 'Qty',
        'harga_saas_inis' => 'Harga Saat Ini',
    );
}

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);
    $criteria->compare('tanggal',$this->tanggal,true);
    $criteria->compare('barang_id',$this->barang_id);
    $criteria->compare('qty',$this->qty);
    $criteria->compare('harga_saas_inis',$this->harga_saas_inis);

    return new CActiveDataProvider($this, array(
        'criteria'=>$criteria,
    ));
}

/*
 * @return Penjualan the static model class
 */
public static function model($className=__CLASS__)
{
    return parent::model($className);
}
}
?>

```

akhir Penjualan.php

4. Buat tabel nota, isikan 1 rekaman, id=1 dan no_nota=1

```

CREATE TABLE `nota` (
  `id` smallint(6) NOT NULL,
  `no_nota` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB

```

5. Buatlah model Nota dengan gii generator
6. Buka model Nota.php tambahkan fungsi `getNo()` membaca nomor nota secara otomatis, seperti pada kode program berikut:

```

public static function getNo()
{
    self::model()->updateCounters(array('no_nota'=> + 1));
}

```

```
$no= Nota::model()->findByPk(1)->no_notas;  
  
return $no;  
}
```

7. Transaksi Penjualan

Buatlah model dari tabel penjualan dan barang
Selanjutnya gunakan CRUD generator

8. Buat model LoginKasir.php, seperti kode program berikut:

```
<?php  
class LoginKasir extends CFormModel  
{  
    public $username;  
    public $no_notas;  
    public $tanggal;  
  
    public function rules()  
    {  
        return array(  
            array('username, noNotas', 'required'),  
        );  
    }  
    public function attributeLabels()  
    {  
        return array(  
            'username'=>'Nama User',  
            'no_notas'=>'No Nota',  
            'tanggal'=>'Tanggal',  
        );  
    }  
}
```

9. Membuat Masukan Transaksi penjualan

Buka file c:\protected\views\penjualan\formKasir.php, dan tambahkan dan ubah seperti pada skrip berikut:

formKasir.php

```
<?php  
/* @var $this PenjualanController */  
/* @var $model Penjualan */  
/* @var $form CActiveForm */  
?>  
  
<div class="wide form">
```

```

<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'kasir-form',
    'enableAjaxValidation'=>true,
    'enableClientValidation'=>true,
)); ?>
<p class="note">Fields with <span class="required">*</span> are required.</p>
    <?php $modelKasir->tanggal=date('Y-m-d'); ?>

    <div class="row">
        <?php echo $form->labelEx($modelKasir,'username'); ?>
        <?php echo $form->textField($modelKasir,'username',
            array('value'=>Yii::app()->user->name)); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($modelKasir,'tanggal'); ?>
        <?php echo $form->textField($modelKasir,'tanggal'); ?>
    </div>
    <div class="row">
        <?php echo $form->labelEx($modelKasir,'no_nota'); ?>
        <?php echo $form->textField($modelKasir,'no_nota',
            array('value'=>$this->nota)); ?>
    </div>

    <div class="row">
        <?php echo CHtml::button('Lanjutkan>>',
            array('submit'=>array('penjualan/create',
                'nota'=>$this->nota))); ?>
    </div>

<?php $this->endWidget(); ?>

</div><!-- form -->

```

akhir formKasir.php



10. Buatlah dan tambahkan view `_formTransaksi.php` seperti pada kode program berikut:

`_formTransaksi.php`

```
<?php
/* @var $this PenjualanController */
/* @var $model Penjualan */
/* @var $form CActiveForm */
?>
<?php
    if(isset($_GET['nota']))
    {
        $this->nota = $_GET['nota'];
        $modelBaru->nota=$this->nota;
    }
    else
        $this->nota = $modelBaru->nota;
?>

<div class="wide form">

<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'penjualan-form',
    'enableAjaxValidation'=>true,
    'enableClientValidation'=>true,
    'clientOptions'=>array('validateOnSubmit'=>true,
                           'validateOnType'=>true,
                           'validateOnChange'=>true),
)); ?>

    <p class="note">Fields with <span class="required">*</span> are
    required.</p>

    <?php echo $form->errorSummary($modelBaru); ?>
    <?php $modelBaru->tanggal=date('Y-m-d');?>

    <div class="row">
        <?php echo $form->labelEx($modelBaru,'tanggal'); ?>
        <?php echo $form->textField($modelBaru,'tanggal'); ?>
        <?php echo $form->error($modelBaru,'tanggal'); ?>
    </div>
    <div class="row">
        <?php echo $form->labelEx($modelBaru,'no_nota'); ?>
        <?php echo $form->textField($modelBaru,'no_nota',
            array('value'=>$this->nota)); ?>
    </div>

    <div class="row">
        <?php echo $form->labelEx($modelBaru,'barang_id'); ?>
        <?php echo $form->textField($modelBaru,'barang_id',
```

```

        array('size'=>3, 'maxlength'=>4,
        'onChange'=>"{selectBarang();}")); ?>
<?php echo $form->error($modelBaru, 'barang_id'); ?>
<?php echo $form->textField($modelBaru, 'nama_barang',
        array('size'=>30, 'maxlength'=>30,)); ?>
<?php echo $form->error($modelBaru, 'nama_barang'); ?>
<?php echo $form->textField($modelBaru, 'qty',
        array('size'=>3, 'maxlength'=>4,
        'style'=>'text-align:right')); ?>
<?php echo $form->error($modelBaru, 'qty'); ?>
<?php echo $form->textField($modelBaru, 'harga_saat_ini',
        array('size'=>10, 'maxlength'=>10,
        'style'=>'text-align:right')); ?>
<?php echo $form->error($modelBaru, 'harga_saat_ini'); ?>
<?php echo $form->textField($modelBaru, 'total',
        array('size'=>10, 'maxlength'=>10,
        'style'=>'text-align:right')); ?>
<?php echo $form->error($modelBaru, 'total'); ?>
<?php echo CHtml::submitButton($modelBaru->isNewRecord ?
        'Simpan' : 'Ubah'); ?>
</div>

<?php $this->endWidget(); ?>

</div><!-- form -->

```

akhir_formTransaksi.php

11. Tambahkan bagian view create.php

create.php

```

<?php
/* @var $this PenjualanController */
/* @var $model Penjualan */

$this->breadcrumbs=array(
    'Penjualans'=>array('index'),
    'Create',
);
?>
<h1>Create Penjualan</h1>
<?php $this->renderPartial('_formTransaksi', array('modelBaru'=>$modelBaru)); ?>

```

akhir create.php

12. Tambahkan bagian PenjualanController.php

PenjualanController.php

```
class PenjualanController extends Controller
{
    public $nota=0;

    .....

    public function accessRules()
    {
        .....
        array('allow',
            'actions'=>array('create','update','formKasir'),
            'users'=>array('@'),
            .....
        )
        .....
    }

    // membaca nomor nota
    // menampilkan nomor nota otomatis
    /*****/

    public function actionFormKasir()
    {
        $modelKasir=new LoginKasir;
        $model=new Penjualan;
        $this->nota = Nota::getNo();
        $this->render('formKasir',array('modelKasir'=>$modelKasir));
    }

    // proses create
    // menambah rekaman transaksi penjualan
    /*****/
}
```



```

public function actionCreate()
{
    $model=new Penjualan('search');
    $model->unsetAttributes(); // clear any default values

    if(isset($_GET['Penjualan']))
    {
        $model->attributes=$_GET['Penjualan'];
    }
    $this->render('create',array(
        'model'=>$model,
        'modelBaru'=>$this->newCreate(),
    ));
}

public function newCreate()
{
    $model=new Penjualan('create');

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

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

        if($model->save())
            $this->refresh();
    }

    return $model;
}
}

```

akhir PenjualanController.php

13. Uji hasilnya dengan memanggil url

<http://localhost/pertemuan5/index.php?r=penjualan/formKasir>

14. Setelah anda selesai praktikum, lakukan backup program dan databasenya untuk dilanjutkan pada pertemuan berikutnya.

D. LATIHAN

Ada di pertemuan Lanjutannya (dilanjutkan pertemuan 6).

E. TUGAS

Jika tidak selesai bisa dilanjutkan di rumah