

## PERTEMUAN KE – 3

### Operasi *CRUD* (*Create, Read, Update, Delete*)

#### A. TUJUAN

1. Mampu mengimplementasikan operasi CRUD, dan menciptakan Model, realasi antar objek menggunakan kelas CActiveRecord.
2. Penggunaan CRUD genertor.
3. Mengenal Struktur folder file-file yang dihasilkan dari perintah gii generator.

#### 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

1. Menciptakan Database seperti pada skrip berikut:

```
CREATE DATABASE web_lanjut1;
```

2. Menciptakan tabel jenis barang,

```
CREATE TABLE jenis_barang (
  id INT(10) UNSIGNED NOT NULL AUTO INCREMENT,
  jenis_barang VARCHAR(40) DEFAULT NULL,
  PRIMARY KEY (id)
) ENGINE=INNODB;
```

3. Menambahkan rekaman jenis barang

```
INSERT INTO jenis_barang(id,jenis_barang)
VALUES (1,'Sembako'),(2,'Alat Rumah Tangga'),(3,'Kosmetik'),(4,'Alat
Tulis');
```

4. Membuat aplikasi baru

Jalankan yiic generator seperti pada perintah berikut:

```
C:\xampp\htdocs\framework>yiic webapp C:\xampp\htdocs\pertemuan3
Create a Web application under 'C:\xampp\htdocs\pertemuan3'? (yes|no)
[no]:yes
      mkdir C:/xampp/htdocs/pertemuan3
      mkdir C:/xampp/htdocs/pertemuan3/assets
      mkdir C:/xampp/htdocs/pertemuan3/css
dst.....
```

5. Pengaturan Koneksi

buka file `/protected/config/main.php`, ubah seperti pada skrip berikut :

```
21     'modules'=>array(  
22         // uncomment the following to enable the Gii tool  
23  
24         'gii'=>array(  
25             'class'=>'system.gii.GiiM  
26             'password'=>'123',  
27             // If removed, Gii defaults to localhost only.  
28             'ipFilters'=>array('127.0.0.1',':::1'),  
29         ),  
30     ),  
31 ),
```

mengganti password  
gii generator

6. Koneksi ke database, user, host dan password

```
53 // u  
54  
55 'db'=>array(  
56     'connectionString' => 'mysql:host=localhost;dbname=web_lanjut1',  
57     'emulatePrepare' => true,  
58     'username' => 'root',  
59     'password' => '',  
60     'charset' => 'utf8',  
61 ),  
62
```

nama host/server

nama database

nama user

mengganti password  
ke mysql

7. Menciptakan model dengan menggunakan generator gii
- <http://localhost/pertemuan3/index.php?r=gii>
- Klik Model Generator
  - Isikan Table Name : jenis\_barang
  - Secara otomatis Model Name : JenisBarang (nama kelas huruf depan kapital)
  - Klik Preview
  - Klik Generate, seperti pada berikut :

Database Connection \* nama koneksi ke database  
db

Table Prefix  
[empty]

Table Name \* nama tabel  
jenis\_barang

Model Class \* nama model/kelas  
JenisBarang

Base Class \*  
CActiveRecord

Model Path \*  
application.models

Build Relations  
☒

Use Column Comments as Attribute Labels  
☐

Code Template \* file hasil generator  
default (C:\xampp\htdocs\framework\yii\generator\models\templates\default)

Preview Generate

Code File	Generate
<a href="#">models\JenisBarang.php</a>	new <input checked="" type="checkbox"/>

#### 8. Menciptakan Controller dan view dengan Crud Generator

- pilih dan Klik Crud Generator
- Masukkan Model Class : JenisBarang
- Controller ID : jenisBarang
- Klik Tombol Preview
- Klik Tombol Generate, seperti pada gambar berikut:

Model Class \*

JenisBarang

nama model/kelas

Controller ID \*

jenisBarang

id controller

Base Controller Class \*

Controller

ekstensikelas

Code Template \*

default (C:\xampp\htdocs\framework\gii\generators\crud\templates\default)

Preview

Generate

Code File	file controller hasil generator	Generate <input checked="" type="checkbox"/>
<a href="#">controllers\JenisBarangController.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\_form.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\_search.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\_view.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\admin.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\create.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\index.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\update.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<a href="#">views\jenisBarang\view.php</a>	new <input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

9. Hasil kode program seperti pada skrip berikut :

model JenisBarang.php

file JenisBarang.php

```
<?php

/*****
 * This is the model class for table "jenis_barang".
 *
 * The followings are the available columns in table 'jenis_barang':
 * @property string $id
 * @property string $jenis_barang
 *****/
class JenisBarang extends CActiveRecord
{
    /**
     * @return string the associated database table name
     */
    public function tableName()
    {
        return 'jenis_barang';
    }

    /**
     * @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('jenis_barang', 'length', 'max'=>40),
        // The following rule is used by search().
        // @todo Please remove those attributes that should not be searched.
        array('id, jenis_barang', '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(
    );
}

/**
 * @return array customized attribute labels (name=>label)
 */
public function attributeLabels()
{
    return array(
        'id' => 'ID',
        'jenis_barang' => 'Jenis Barang',
    );
}

/*****
 * 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()
{
    $criteria=new CDbCriteria;

    $criteria->compare('id',$this->id,true);
    $criteria->compare('jenis_barang',$this->jenis_barang,true);

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

/**
 * 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 $className active record class name.
 * @return JenisBarang the static model class
 */
public static function model($className=__CLASS__)
{

```

```

        return parent::model($className);
    }
}

```

file JenisBarang.php

Controller JenisBarangController.php

file JenisBarangController.php

```

<?php

class JenisBarangController 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('admin'),
            ),
            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 JenisBarang;

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

    if(isset($_POST['JenisBarang']))
    {
        $model->attributes=$_POST['JenisBarang'];
        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['JenisBarang']))
    {
        $model->attributes=$_POST['JenisBarang'];
        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('JenisBarang');
    $this->render('index',array(
        'dataProvider'=>$dataProvider,
    ));
}

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

    $this->render('admin',array(
        'model'=>$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 $id the ID of the model to be loaded
 * @return JenisBarang the loaded model
 * @throws CHttpException
 */
public function loadModel($id)
{
    $model=JenisBarang::model()->findByPk($id);
    if($model===null)
        throw new CHttpException(404,
            'The requested page does not exist.');
```

```

    return $model;
}

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

```

akhir file JenisBarangController.php

View jenisBarang/\_index.php

file index.php

```

<?php
/* @var $this JenisBarangController */
/* @var $dataProvider CActiveDataProvider */

$this->breadcrumbs=array(

```



```

        'Jenis Barangs',
    );

    $this->menu=array(
        array('label'=>'Create JenisBarang', 'url'=>array('create')),
        array('label'=>'Manage JenisBarang', 'url'=>array('admin')),
    );
    ?>

<h1>Jenis Barangs</h1>

<?php $this->widget('zii.widgets.CListView', array(
    'dataProvider'=>$dataProvider,
    'itemView'=>'_view',
)); ?>

```

akhir file index.php

Bagian \_view.php seperti pada kode program berikut:

file \_view.php

```

<?php
/* @var $this JenisBarangController */
/* @var $data JenisBarang */
?>

<div class="view">

    <b><?php echo CHtml::encode($data->getAttributeLabel('id')); ?></b>
    <?php echo CHtml::link(CHtml::encode($data->id),
        array('view', 'id'=>$data->id));
    ?>
    <br />

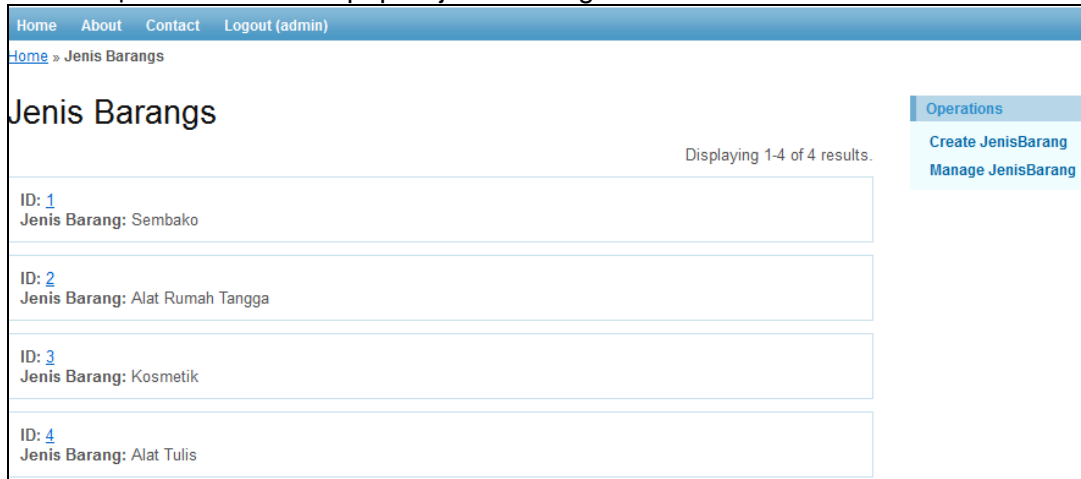
    <b>
    <?php echo CHtml::encode($data->getAttributeLabel('jenis_barang')); ?>:
    </b>
    <?php echo CHtml::encode($data->jenis_barang); ?>
    <br />

</div>

```

file \_view.php

http://localhost/pertemuan3/index.php?r=jenisBarang/index



file admin.php

```

<?php
/* @var $this JenisBarangController */
/* @var $model JenisBarang */

$this->breadcrumbs=array(
    'Jenis Barangs'=>array('index'),
    'Manage',
);

$this->menu=array(
    array('label'=>'List JenisBarang', 'url'=>array('index')),
    array('label'=>'Create JenisBarang', 'url'=>array('create')),
);

Yii::app()->clientScript->registerScript('search', "
$( '.search-button' ).click(function() {
    $( '.search-form' ).toggle();
    return false;
});
$( '.search-form form' ).submit(function() {
    $( '#jenis-barang-grid' ).yiiGridView('update', {
        data: $(this).serialize()
    });
    return false;
});
");
?>

<h1>Manage Jenis Barangs</h1>

<p>
You may optionally enter a comparison operator (<, <=,
>, >=, <>) at the beginning of each of your search values to specify how
the comparison should be done.
</p>

<?php echo CHtml::link('Advanced Search','#',array('class'=>'search-
button')); ?>
<div class="search-form" style="display:none">
<?php $this->renderPartial('_search',array(
    'model'=>$model,
)); ?>
</div><!-- search-form -->

<?php $this->widget('zii.widgets.grid.CGridView', array(
    'id'=>'jenis-barang-grid',
    'dataProvider'=>$model->search(),
    'filter'=>$model,
    'columns'=>array(
        'id',
        'jenis_barang',
        array(
            'class'=>'CButtonColumn',
        ),
    ),
)); ?>

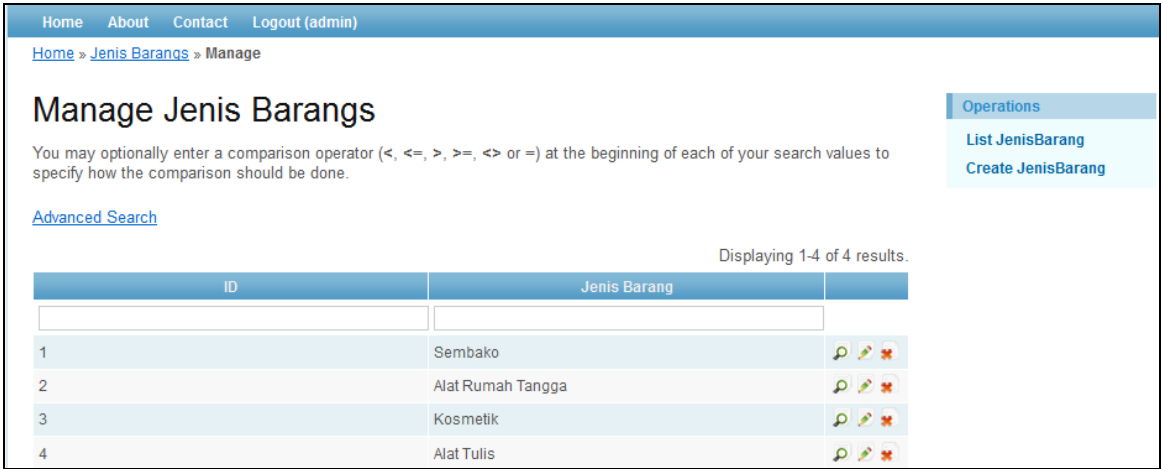
```

akhir file admin.php

Ujikan dengan menjalankan url:

<http://localhost/pertemuan3/index.php?r=jenisBarang/admin>

hasil keluaranya seperti pada gambar berikut:



Ubah dibagian views/barang/\_serach.php seperti pada kode program berikut:

file \_search.php

```
<?php
/* @var $this JenisBarangController */
/* @var $model JenisBarang */
/* @var $form CActiveForm */
?>

<div class="wide form">

<?php $form=$this->beginWidget('CActiveForm', array(
    'action'=>Yii::app()->createUrl($this->route),
    'method'=>'get',
)); ?>

    <div class="row">
        <?php echo $form->label($model,'id'); ?>
        <?php echo $form->textField($model,'id',
            array('size'=>10,'maxlength'=>10)); ?>
    </div>

    <div class="row">
        <?php echo $form->label($model,'jenis_barang'); ?>
        <?php echo $form->textField($model,'jenis_barang',
            array('size'=>40,'maxlength'=>40)); ?>
    </div>

    <div class="row buttons">
        <?php echo CHtml::submitButton('Search'); ?>
    </div>

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

</div><!-- search-form -->
```

akhir file \_search.php

Ujikan klik link Advance Search:  
hasil keluaranya seperti pada gambar berikut:

# Manage Jenis Barangs

You may optionally enter a comparison operator (<, <=, >, >=, <> or =) at the beginning of each of your search values to specify how the comparison should be done.

[Advanced Search](#)

ID

Jenis Barang

Search

Displaying 1-4 of 4 results.

ID	Jenis Barang	
1	Sembako	  
2	Alat Rumah Tangga	  
3	Kosmetik	  
4	Alat Tulis	  

file \_from.php

```
<?php
/* @var $this JenisBarangController */
/* @var $model JenisBarang */
/* @var $form CActiveForm */
?>

<div class="form">

<?php $form=$this->beginWidget('CActiveForm', array(
    'id'=>'jenis-barang-form',
    // Please note: When you enable ajax validation, make sure the corresponding
    // controller action is handling ajax validation correctly.
    // There is a call to performAjaxValidation() commented in generated controller code.
    // See class documentation of CActiveForm for details on this.
    'enableAjaxValidation'=>false,
)); ?>

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

<?php echo $form->errorSummary($model); ?>

<div class="row">
    <?php echo $form->labelEx($model,'jenis_barang'); ?>
    <?php echo $form->textField($model,'jenis_barang',
        array('size'=>40,'maxlength'=>40)); ?>
    <?php echo $form->error($model,'jenis_barang'); ?>
</div>

<div class="row buttons">
    <?php echo CHtml::submitButton($model->isNewRecord ?
        'Create' : 'Save'); ?>
</div>

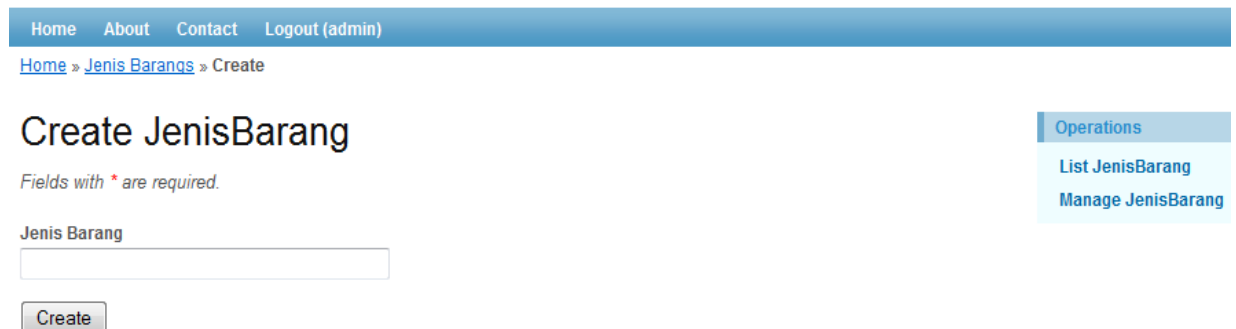
<?php $this->endWidget(); ?>
```

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

file \_from.php

Ujikan dengan menjalankan url:

<http://localhost/pertemuan3/index.php?r=jenisBarang/create>



file view.php

```
<?php
/*@var $this JenisBarangController */
/*@var $model JenisBarang */

$this->breadcrumbs=array(
    'Jenis Barangs'=>array('index'),
    $model->id,
);

$this->menu=array(
    array('label'=>'List Jenis Barang', 'url'=>array('index')),
    array('label'=>'Create Jenis Barang', 'url'=>array('create')),
    array('label'=>'Update Jenis Barang', 'url'=>array('update',
        'id'=>$model->id)),
    array('label'=>'Delete Jenis Barang', 'url'=>'#',
        'linkOptions'=>array('submit'=>array('delete',
            'id'=>$model->id),
            'confirm'=>'Are you sure you want to delete this item?')),
    array('label'=>'Manage Jenis Barang', 'url'=>array('admin')),
);
?>

<h1>View Jenis Barang #<?php echo $model->id; ?></h1>

<?php $this->widget('zii.widgets.CDetailView', array(
    'data'=>$model,
    'attributes'=>array(
        'id',
        'jenis_barang',
    ),
)); ?>
```

file view.php

Ujikan dengan menjalankan url:

<http://localhost/pertemuan3/index.php?r=jenisBarang/view&id=1>

hasil keluaranya seperti pada gambar berikut:

<a href="#">Home</a> <a href="#">About</a> <a href="#">Contact</a> <a href="#">Logout (admin)</a>	
<a href="#">Home</a> » <a href="#">Jenis Barangs</a> » 1	
View JenisBarang #1	
ID	1
Jenis Barang	Sembako
Operations	
<a href="#">List JenisBarang</a>	
<a href="#">Create JenisBarang</a>	
<a href="#">Update JenisBarang</a>	
<a href="#">Delete JenisBarang</a>	
<a href="#">Manage JenisBarang</a>	

D. LATIHAN

1. Menciptakan tabel barang

```
CREATE TABLE barang (  
  id INT(11) NOT NULL AUTO_INCREMENT,  
  jenis_barang_id INT(11) DEFAULT NULL,  
  nama_barang VARCHAR(40) DEFAULT NULL,  
  satuan VARCHAR(30) DEFAULT NULL,  
  harga INT(11) DEFAULT '0',  
  stok INT(11) DEFAULT '0',  
  PRIMARY KEY (id)  
) ENGINE=INNODB ;
```

2. Menambahkan rekaman barang

```
INSERT INTO barang(id,jenis_barang_id,nama_barang,satuan,harga,stok)  
VALUES (1,1,'Gula','Kg',5000,20),  
       (2,1,'Beras Cianjur','Kg',10000,100),  
       (3,1,'Minyak Goreng','Kg',20000,25),  
       (4,2,'Gelas','Dosin',30000,3),  
       (5,2,'Sendok Makan','Dosin',10000,10),  
       (6,3,'Sabun Mandi','Bungkus',5000,10),  
       (7,3,'Sampho ','Botol 800 mg',20000,10),  
       (8,4,'Buku Tulis','Pcs',5000,10),  
       (9,1,'Jagung','Kg',4000,100),  
       (10,1,'Beras C4','Kg',8000,100);
```

3. Buatlah model-view-controller dengan menggunakan gii Generator, ubah sehingga isian form seperti berikut:

Create Barang

Fields with \* are required.

Jenis Barang \*

Sembako

Jenis Barang tidak boleh kosong.

Nama Barang \*

Satuan

Harga

0

Stok

0

Create

4. Untuk membuat masukkan dengan menggunakan dropDownList ubah bagian \_form.php seperti pada kode program berikut:

```

<div class="row">
    <?php echo $form->labelEx($model,'jenis_barang_id'); ?>
    <?php echo $form->dropDownList($model,'jenis_barang_id',
        CHtml::listData(JenisBarang::model()->findAll(), 'id','jenis_barang'),
        array('prompt'=>'= Pilihan =','style'=>'width:200px;')); ?>
    <?php echo $form->error($model,'jenis_barang_id'); ?>
</div>

```

## E. TUGAS

1. Bagaimana cara kerja pencarian `_search` ?
2. Apa yang dimaksud perintah  

```

<?php echo CHtml::submitButton($model->isNewRecord ? 'Create' : 'Save');
?>

```
3. Apa yang dimaksud perintah `if($this->isNewRecord)` ?
4. Apa yang dimaksud perintah `$barang->save()` ;