

Pengembangan Aplikasi Berbasis Yii

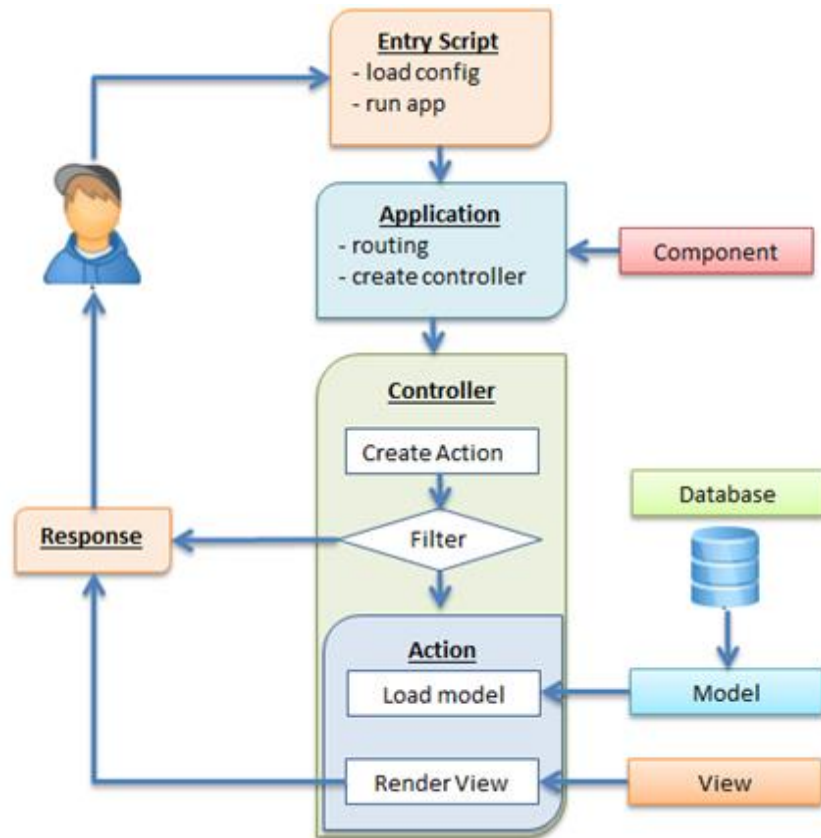
Chapter I

Agenda

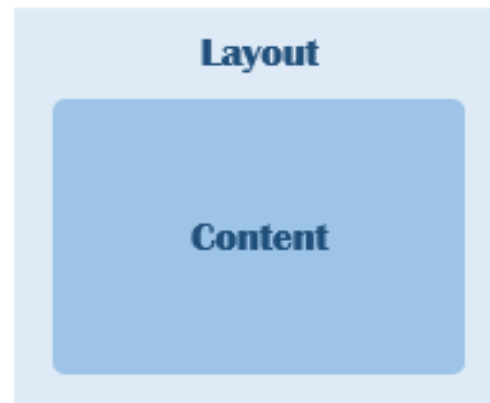
- ❖ Memahami Arsitektur Yii
- ❖ Implementasi MVC di Yii
- ❖ Mengenal ActiveRecord & Implementasi CRUD

Memahami Arsitektur Yii

Alur Cara Bekerja Aplikasi Yii



- Entryscript: `app\web\index.php`
- Default controller: `SiteController`
- Default method: `actionIndex`
- Default layout: `@app/views/layouts/main.php`
- Default view: `@app/views/site/index.php`

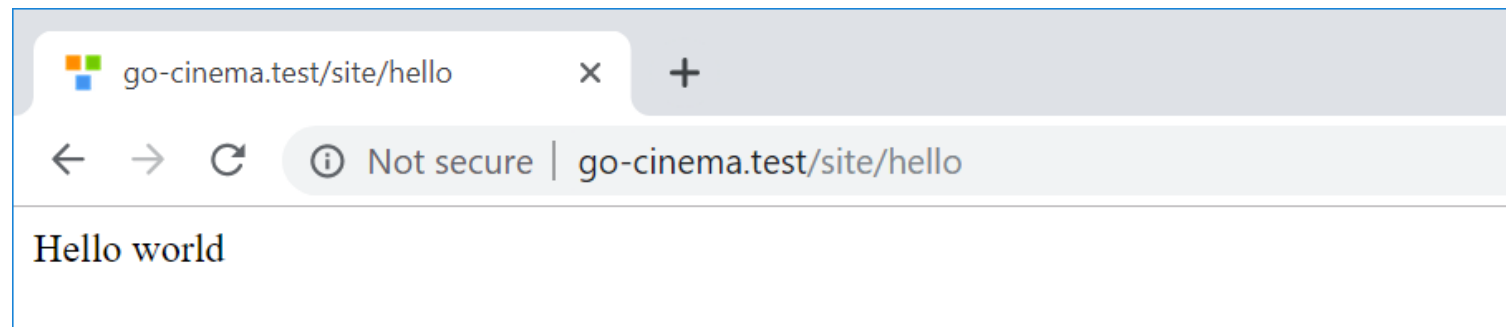


Menampilkan Hello World

1. Buat method actionHello pada file @/controllers/SiteController.php
2. Kode berikut

```
public function actionHello()  
{  
    return "Hello world";  
}
```

3. Pada browser akses <http://go-cinema.test/site/hello>



Aturan Routing Yii

❑ Controller

Format: **Nama**Controller

URL: /**nama**

Contoh: **Site**Controller => http://go-cinema.test/**site**

❑ Action pada Controller

Format: action**Nama** (**Site**Controller)

URL: /site/**nama**

Contoh: **Site**Controller dan action**Hello** => http://go-cinema.test/**site**/**hello**

❑ Penulisan URL

Format penulisan URL menggunakan lowercase

Jika dua kata gunakan format camelCase dan URLnya gunakan kebab-case

Contoh: actionHelloWorld => /hello-world

Hello World + View

Pada bagian sebelumnya kita hanya menggunakan controller saja, pada bagian ini kita akan mencoba menggunakan View.

1. Buat file hello.php pada folder @/views/site/, yang isinya

```
<h1>Hello world!</h1>
```

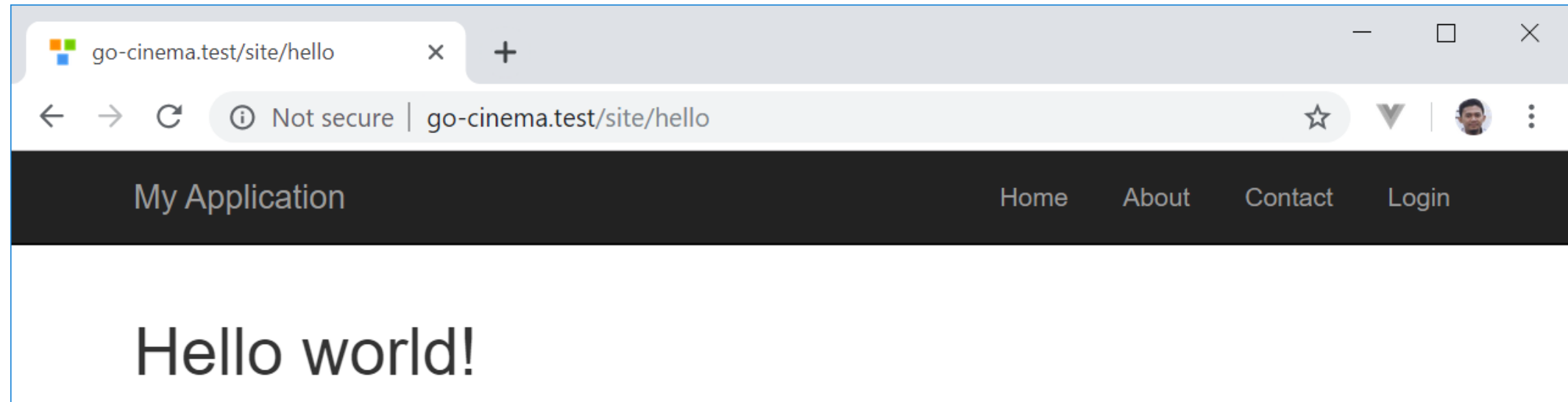
2. Kode berikut

```
public function actionHello()  
{  
    //return "Hello world";  
    return $this->render('hello');  
}
```

3. Pada browser refresh <http://go-cinema.test/site/hello>

Hello World + View

Perintah `$this->render('hello');` pada `actionHello()` maksudnya adalah merender file `hello.php` yang berada pada folder `@/views/site/` jika `SiteController`,



Implementasi MVC di Yii

MVC Movie

Membuat Model Movie

Buat file Movie.php pada @/models/

Sebagai contoh ada dua data yang kita gunakan yaitu title atau judul film dan description atau deskripsi film

```
<?php
namespace app\models;
use Yii;
use yii\base\Model;

class Movie extends Model
{
    public $title;
    public $description;

    public function rules()
    {
        return [
            [['title', 'description'], 'safe'],
        ];
    }
}
```

Membuat Controller Movie

Buat file MovieController pada folder @/controllers

<http://go-cinema.test/movie>

```
<?php
namespace app\controllers;
use Yii;
use yii\web\Controller;

class MovieController extends Controller
{

}
```

Membuat actionForm Pada MovieController

Buat method actionForm pada MovieController yang isinya inisiasi class Movie dan kemudian merendernya atau mengirimkan objectnya ke file form.php (view)

<http://go-cinema.test/movie/form>

```
use app\models\Movie;

class MovieController extends Controller
{
    public function actionForm()
    {
        $model = new Movie();
        return $this->render('form', [
            'model' => $model,
        ]);
    }
}
```

Membuat View Form

Buat file form.php pada direktori @/movie/,

Komponen form menggunakan widget ActiveForm bawaan Yii dan helper Html

```
<?php
use yii\helpers\Html;
use yii\bootstrap\ActiveForm;
?>
<?php $form = ActiveForm::begin(); ?>
<?= $form->field($model, 'title')->textInput() ?>
<?= $form->field($model, 'description')->textarea() ?>
<div class="form-group">
    <?= Html::submitButton('Submit', [
        'class' => 'btn btn-primary'
    ]) ?>
</div>
<?php ActiveForm::end(); ?>
```

Penampakan Sementara

←

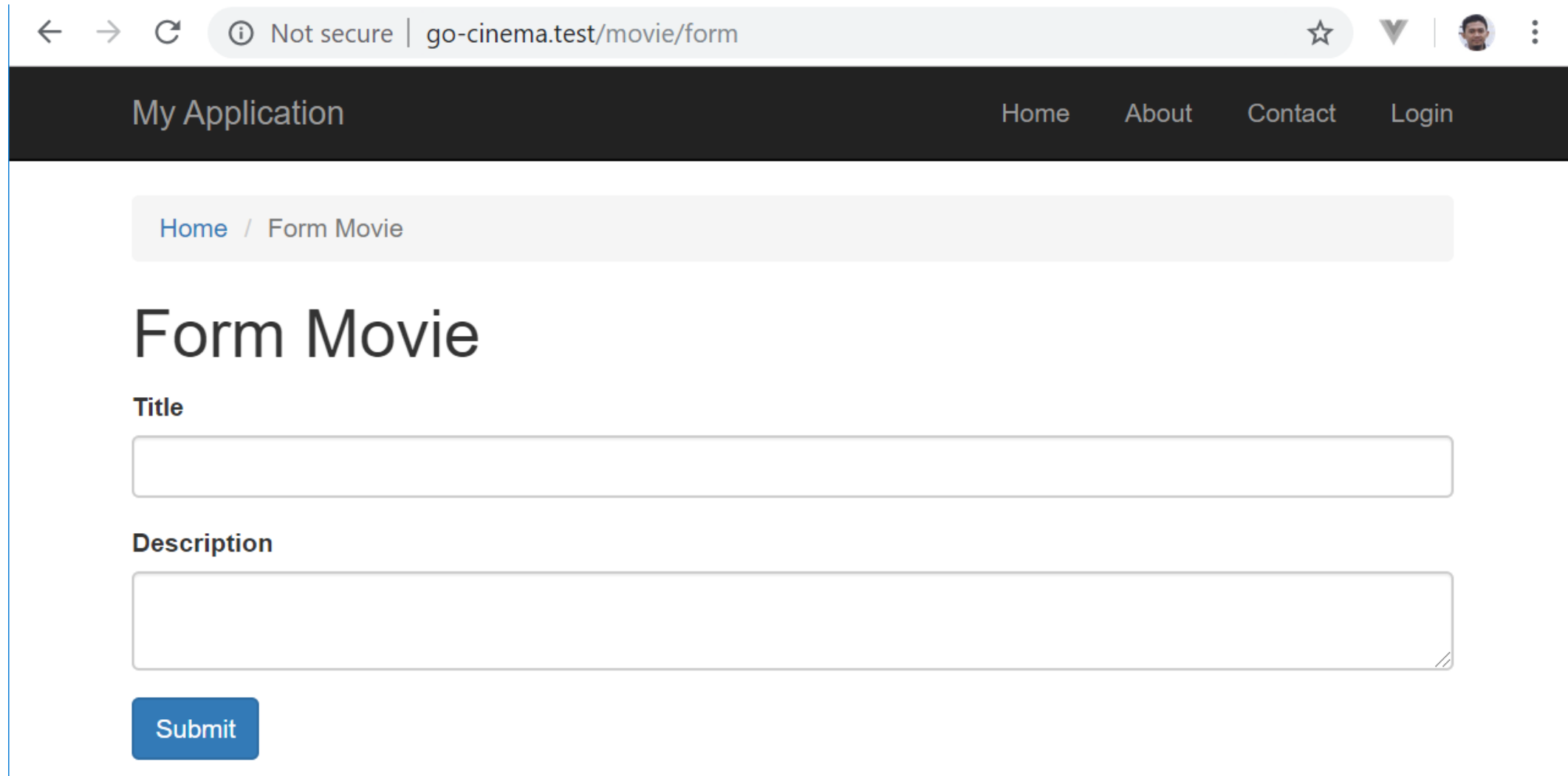
→

↻

ⓘ Not secure | go-cinema.test/movie/form

☆

▼



⋮

My Application

Home

About

Contact

Login

Home / Form Movie

Form Movie

Title

Description

Submit

Memperbaharui actionForm

Pada actionForm tambahkan kode untuk menangani submit data melalui form.

Kode `Yii::$app->request->post()` setara dengan `$_FORM`

Kode `$model->load` digunakan untuk menangkap variable dan memasukkannya pada property model yang sesuai


```
public function actionForm()
{
    $model = new Movie();
    if ($model->load(Yii::$app->request->post())) {
        // perintah untuk menyimpan ke database
    }

    return $this->render('form', [
        'model' => $model,
    ]);
}
```

Penampakan Berikutnya



The screenshot shows a web browser window with the address bar displaying "Not secure | go-cinema.test/movie/form". The page has a dark navigation bar with "My Application" on the left and "Home", "About", "Contact", and "Login" on the right. Below the navigation bar, a breadcrumb trail shows "Home / Form Movie". The main heading is "Form Movie". There are two form fields: "Title" with the value "GUNDALA" and "Description" with the text "Sancaka hidup di jalanan sejak ditinggal ayah dan ibunya. Menghadapi hidup yang keras, Sancaka belajar untuk bertahan hidup dengan tidak peduli dengan orang lain dan hanya mencoba untuk mendapatkan tempat". A blue "Submit" button is at the bottom left.

← → ↻ ⓘ Not secure | go-cinema.test/movie/form ☆ ▼ |  ⋮

My Application Home About Contact Login

Home / Form Movie

Form Movie

Title

GUNDALA

Description

Sancaka hidup di jalanan sejak ditinggal ayah dan ibunya. Menghadapi hidup yang keras, Sancaka belajar untuk bertahan hidup dengan tidak peduli dengan orang lain dan hanya mencoba untuk mendapatkan tempat

Submit

Menambahkan Pesan Notifikasi Simpan

Untuk menampilkan pesan notifikasi, pada Yii terdapat widget Alert yang merupakan bawaan Yii (@/widgets/Alert.php)

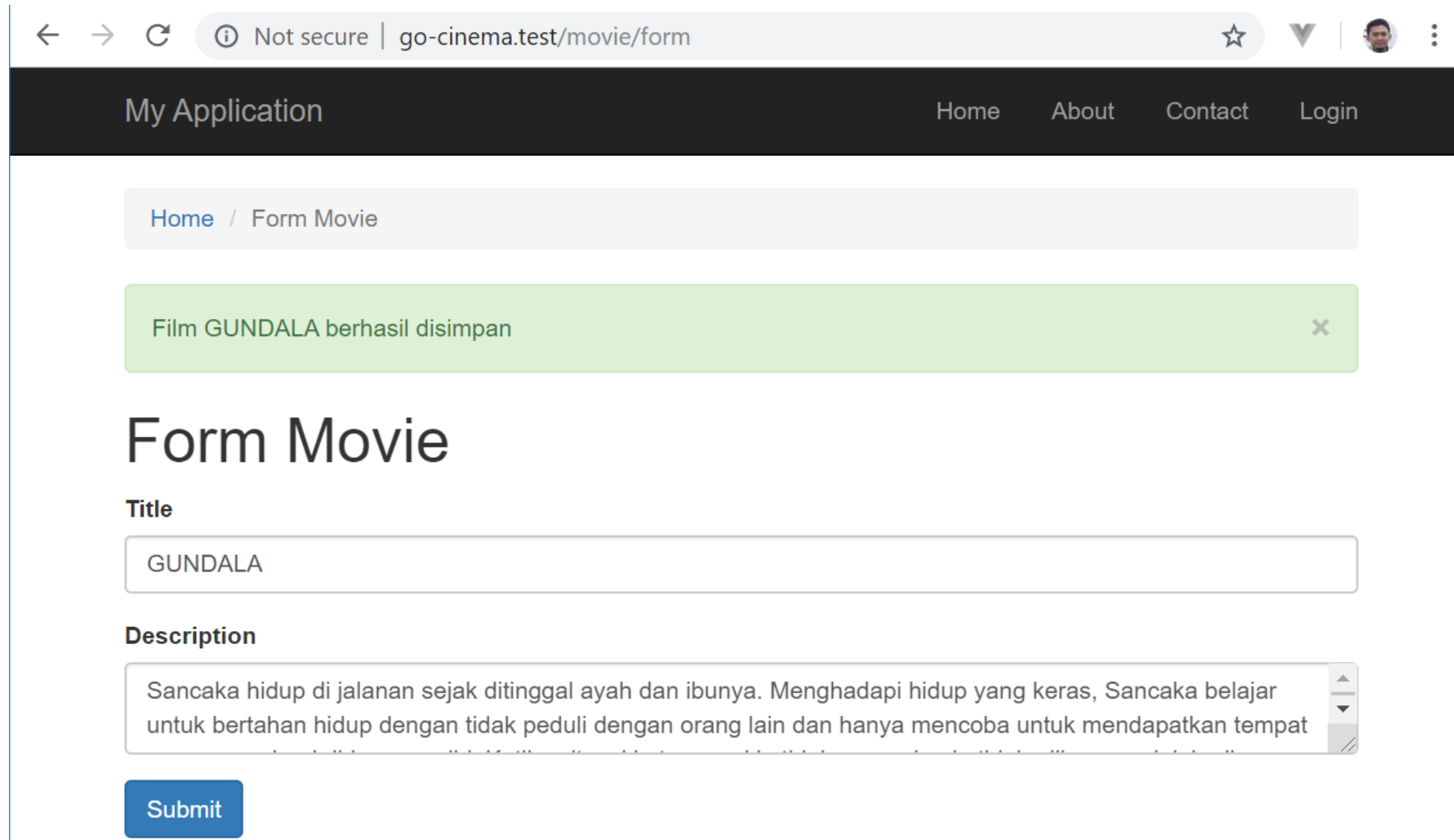
```
if ($model->load(Yii::$app->request->post())) {  
    // perintah untuk menyimpan ke database  
    Yii::$app->session->setFlash('success',  
        'Film '.$model->title.' berhasil disimpan');  
}
```

Untuk menampilkan pesan notifikasi, pada Yii terdapat widget Alert yang merupakan bawaan Yii (@/widgets/Alert.php)

`Yii::$app->session->setFlash(tipe notifikasi, teks notifikasi)`

Ada 3 kemungkinan tipe notifikasi yaitu: success, error, dan info. Pada contohnya, notifikasi akan ditampilkan ketika submit data telah dilakukan. Karena model telah memuat hasil data post maka, kita bisa menggunakan kode `$model->title` untuk menampilkan data judul filmnya

Penampakan Akhir



The screenshot shows a web browser window with the address bar displaying "Not secure | go-cinema.test/movie/form". The page has a dark navigation bar with "My Application" on the left and "Home", "About", "Contact", and "Login" on the right. Below the navigation bar, a breadcrumb trail shows "Home / Form Movie". A green success message box states "Film GUNDALA berhasil disimpan" with a close button. The main heading is "Form Movie". The "Title" field contains "GUNDALA". The "Description" field contains a paragraph about a character named Sancaka. A blue "Submit" button is at the bottom left.

← → ↻ ⓘ Not secure | go-cinema.test/movie/form ☆ ▼ | 👤 ⋮

My Application Home About Contact Login

Home / Form Movie

Film GUNDALA berhasil disimpan ✕

Form Movie

Title

GUNDALA

Description

Sancaka hidup di jalanan sejak ditinggal ayah dan ibunya. Menghadapi hidup yang keras, Sancaka belajar untuk bertahan hidup dengan tidak peduli dengan orang lain dan hanya mencoba untuk mendapatkan tempat

Submit

Mengenal ActiveRecord


+ Implementasi CRUD menggunakan ActiveRecord

Apa itu ActiveRecord (AR)?

- AR merupakan salah satu penerapan konsep dalam design pattern di Yii yang digunakan sebagai antar muka dalam berinteraksi dengan tabel pada database.
- AR akan memudahkan kita melakukan operasi data pada database seperti menampilkan, memfilter, menambahkan, mengedit dan menghapus data.
- Untuk menggunakan AR maka class model yang sebelumnya extends ke class `yii\base\Model`, diubah menjadi extends ke class `\yii\db\ActiveRecord`

Membuat Tabel Movie

Buatlah tabel movie pada database go-cinemadb

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	None		AUTO_INCREMENT
2	title	varchar(255)	latin1_swedish_ci		No	None		
3	description	text	latin1_swedish_ci		No	None		

Modifikasi Model Movie

Modifikasi file Movie.php pada @/models/ dengan extends ke class ActiveRecord yaitu yii\db\ActiveRecord

Property title dan description sudah tidak diperlukan lagi karena by default akan merujuk ke field di database pada tabel movie. Pada kasus inipun, method tableName sebenarnya opsional ketika nama tablenya berbeda dengan nama classnya.

```
<?php
namespace app\models;
use Yii;
use yii\db\ActiveRecord;

class Movie extends ActiveRecord
{
    public static function tableName()
    {
        return 'movie';
    }

    public function rules()
    {
        return [
            [['title','description'], 'safe'],
        ];
    }
}
```

CRUD Query Pada ActiveRecord

- Create
 - Insert into movie (title, description) values ("", "")
- Read
 - Select * from movie; // semua data
 - Select * from movie where genre = 'horor'; // semua data yang genre horror
 - Select * from movie where id=2; // satu data yang id=2
- Update
 - Update movie set title = 'Gundala 2' where id=1; // update judul menjadi Gundala 2 dimana id=1
- Delete
 - Delete from movie where id=1; // hapus data dimana id=1

Create / Input Data Baru

```
// create atau insert into movie(title, description) value ('', '')  
$model = new Movie();  
$model->title = "Gundala";  
$model->description = "Sancaka hidup di jalanan sejak ditinggal ayah dan ibunya.";  
$model->save();
```


Modifikasi actionForm pada MovieController

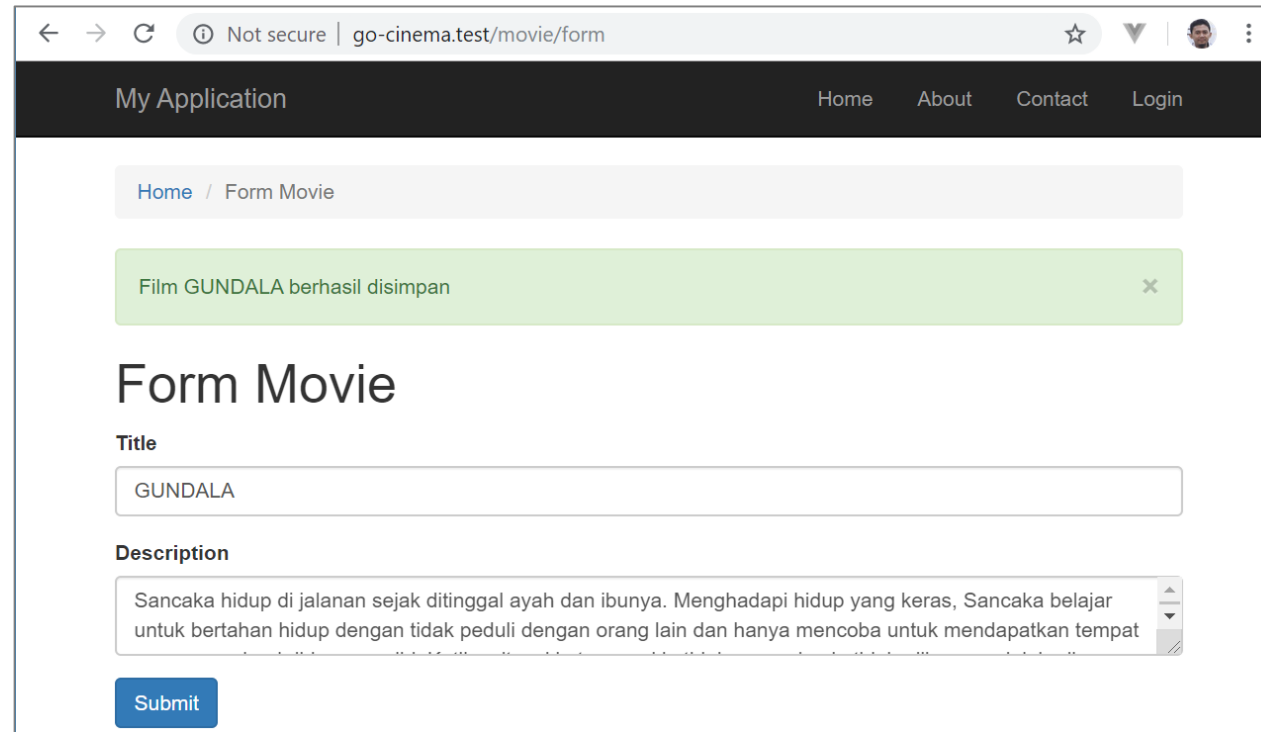
Cukup tambahkan kode
`$model->save();`

Karena baris sebelumnya
sudah ada perintah untuk
menangkap variable post
data `$model->load(Yii::$app->request->post())`

```
public function actionForm()
{
    $model = new Movie();
    if ($model->load(Yii::$app->request->post())) {
        // perintah untuk menyimpan ke database
        $model->save();
        Yii::$app->session->setFlash('success',
            'Film '.$model->title.' berhasil disimpan');
    }

    return $this->render('form', [
        'model' => $model,
    ]);
}
```

Submit Form & Check Database



The screenshot shows a web browser at the URL `go-cinema.test/movie/form`. The page has a dark navigation bar with "My Application" and links for "Home", "About", "Contact", and "Login". Below the navigation bar is a breadcrumb trail "Home / Form Movie". A green success message box states "Film GUNDALA berhasil disimpan". The main form is titled "Form Movie" and contains two fields: "Title" with the value "GUNDALA" and "Description" with the text "Sancaka hidup di jalanan sejak ditinggal ayah dan ibunya. Menghadapi hidup yang keras, Sancaka belajar untuk bertahan hidup dengan tidak peduli dengan orang lain dan hanya mencoba untuk mendapatkan tempat". A blue "Submit" button is located at the bottom of the form.

id	title	description
1	GUNDALA	Sancaka hidup di jalanan sejak ditinggal ayah dan ...

Silakan input beberapa data film

<https://go-tix.id/movies>

Menampilkan Semua Data

```
// menampilkan semua data pada tabel movie
```

```
$model = Movie::find()->all();
```

```
// menampilkan semua data dengan kriteria
```

```
$model = Movie::find()->where(['genre' => 'horror'])->all();
```

Method actionIndex Pada MovieController

Pada MovieController tambahkan method actionIndex untuk menampilkan daftar film

<http://go-cinema.test/movie/index>

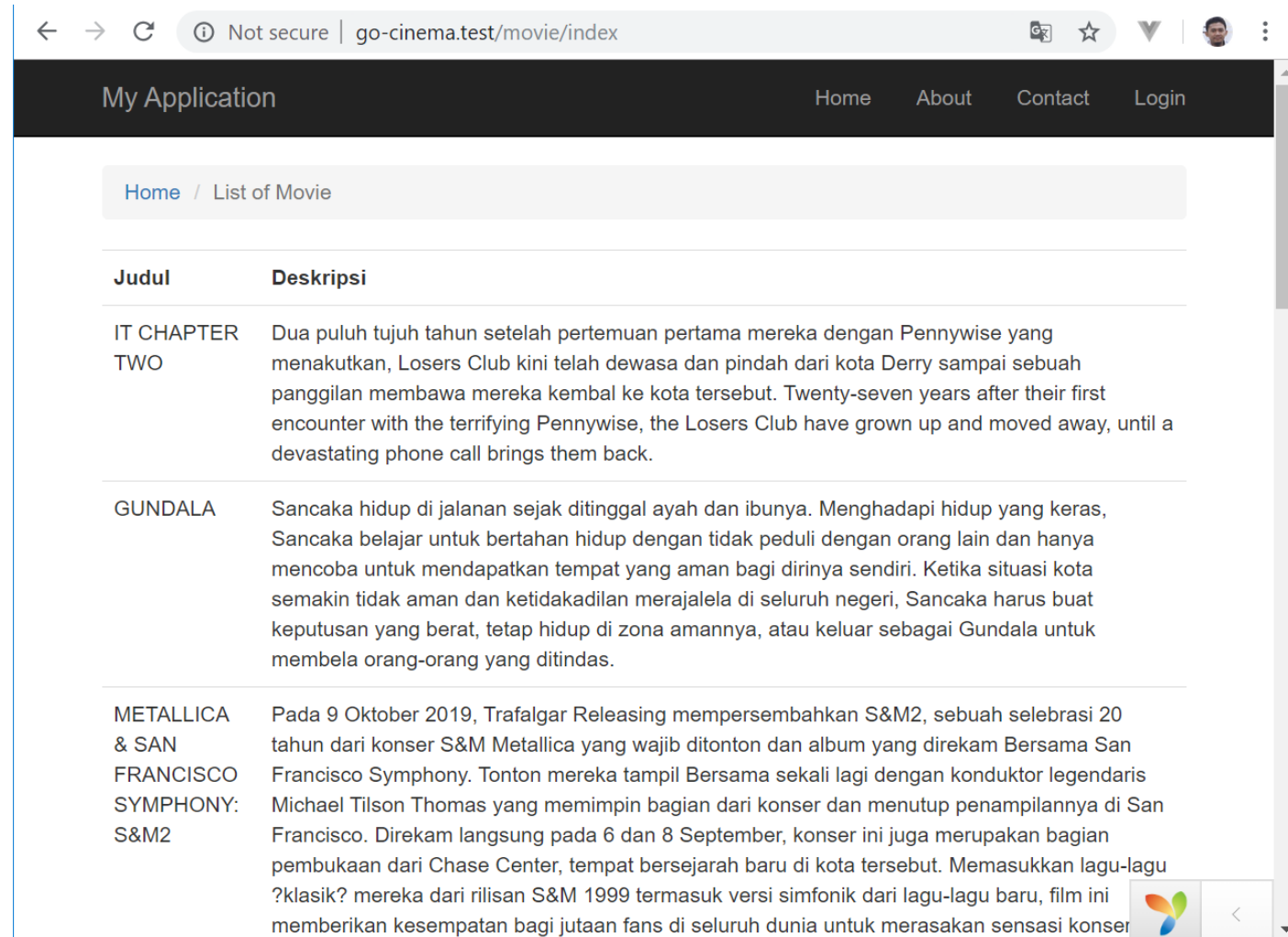
```
public function actionIndex()  
{  
    $model = Movie::find()->all();  
    return $this->render('index', [  
        'model' => $model,  
    ]);  
}
```

View index Pada @/movie/index.php

Tambahkan file index.php
pada @/movie/index.php

```
<div>
  <table class="table">
    <tr>
      <th>Judul</th>
      <th>Deskripsi</th>
    </th>
    <?php
      foreach($model as $movie){
        echo "<tr>";
        echo "<td>".$movie->title."</td>";
        echo "<td>".$movie->description."</td>";
        echo "</tr>";
      }
    ?>
  </table>
</div>
```

Test di Browser /movie/index



Menampilkan Sebuah Data

```
// menampilkan sebuah data
```

```
$model = Movie::find()->one();
```

```
// menampilkan sebuah data dengan kriteria
```

```
$model = Movie::find()->where(['id' => 1])->one();
```


Method actionView Pada MovieController

Pada MovieController tambahkan method actionView untuk menampilkan data sebuah film dengan id sesuai parameter

<http://go-cinema.test/movie/view>

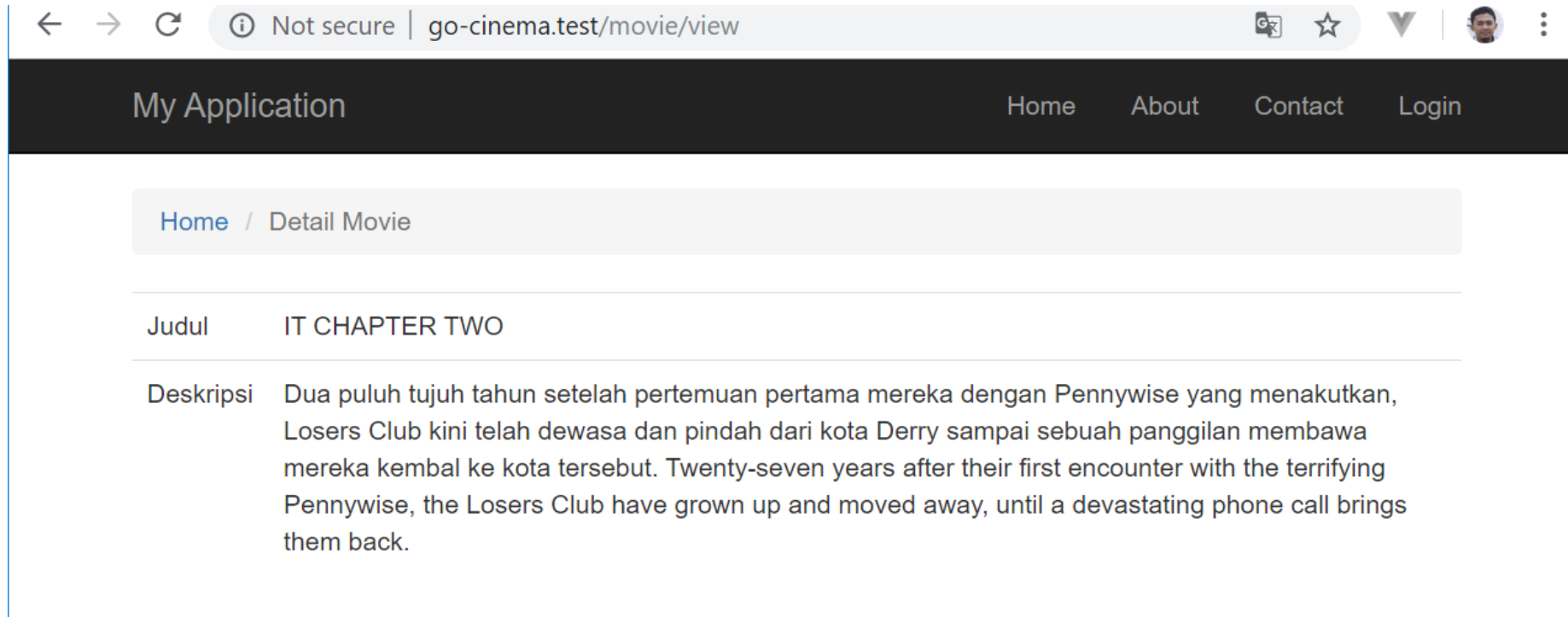
```
public function actionView()  
{  
    $model = Movie::find()->where([  
        'id' => 1  
    ])->one();  
  
    return $this->render('view', [  
        'model' => $model,  
    ]);  
}
```

View view Pada @/movie/view.php

Tambahkan file view.php
pada @/movie/view.php

```
<div>
  <table class="table">
    <?php
    echo "<tr>";
      echo "<td> Judul </td>";
      echo "<td>".$model->title."</td>";
    echo "</tr>";
    echo "<tr>";
      echo "<td> Deskripsi </td>";
      echo "<td>".$movie->description."</td>";
    echo "</tr>";
    ?>
  </table>
</div>
```


Test di Browser /movie/view




Parameter Dinamis Pada actionView

Modifikasi method `actionView` pada `MovieController` untuk menambahkan parameter dinamis sehingga data yang ditampilkan sesuai dengan id yang diinginkan, dimana parameter id tersebut akan dilewatkan melalui URL

<http://go-cinema.test/movie/view?id=1>



```
public function actionView($id)
{
    $model = Movie::find()->where([
        'id' => $id
    ])->one();
    return $this->render('view', [
        'model' => $model,
    ]);
}
```



Link Detail Pada View index

Tambahkan link detail pada setiap baris film di view index, sehingga ketika diklik akan menuju ke detail film,

Adapun kolom deskripsi diubah menjadi kolom detail yang berisi link detail dari film tersebut

Menggunakan helper
Html::a

```
<?php
use yii\helpers\Html;
?>
<div>
    <table class="table">
        <tr>
            <th>Judul</th>
            <th>Detail</th>
        </tr>
        <?php
        foreach($model as $movie){
            echo "<tr>";
            echo "<td>".$movie->title."</td>";
            echo "<td>".
                Html::a('Detail', ['view', 'id'=>$movie->id]).
                "</td>";
            echo "</tr>";
        }
        ?>
    </table>
</div>
```

Membuat Link menggunakan Helper Html

Untuk membuat link dengan kode Html

```
<a href="http://go-cinema.test/movie/view?id=1"> detail </a>
```

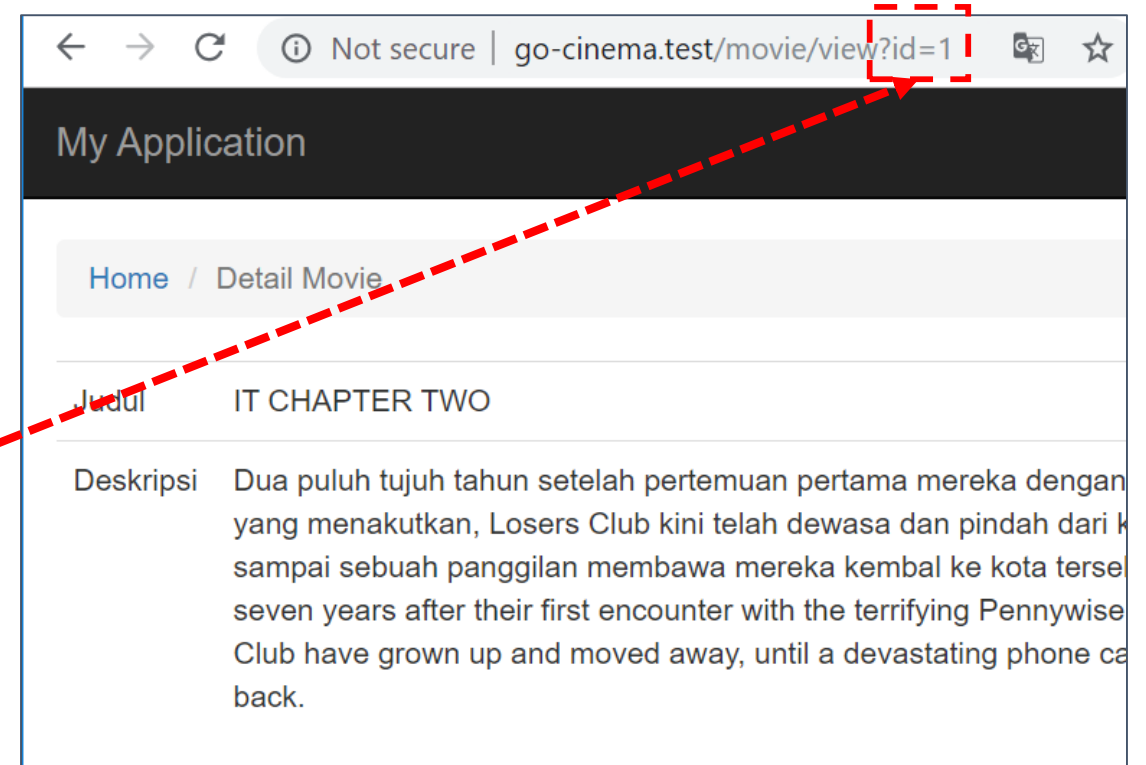
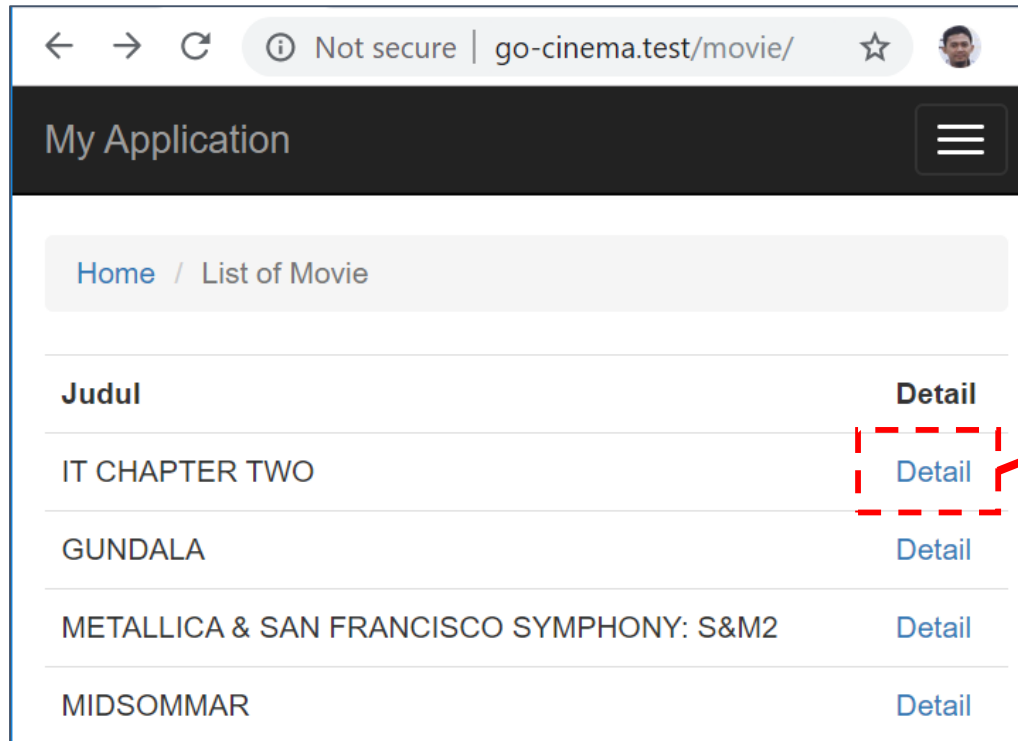
Yii mempunyai class helper Html (yii\helpers\Html) yang salah satu fungsinya untuk membuat link (anchor)

```
echo Html::a(" detail ", [ "movie/view", "id" => 1 ]);
```

Atau jika pada controller yang sama maka nama controller boleh dihilangkan

```
echo Html::a(" detail ", [ "view", "id" => 1 ]);
```

Test di Browser /movie/index



Update Data

```
// update data tabel movie dengan id 1  
$model = Movie::findOne(1);  
$model->title = "Gundala 2";  
$model->save();
```


Tambahkan actionUpdate pada MovieController

Kodenya hampir sama dengan method actionView, perbedaannya pada file rendernya. Di mana kita render pada view form.php yang telah kita buat sebelumnya

```
public function actionUpdate($id)
{
    $model = Movie::find()->where([
        'id' => $id
    ])->one();

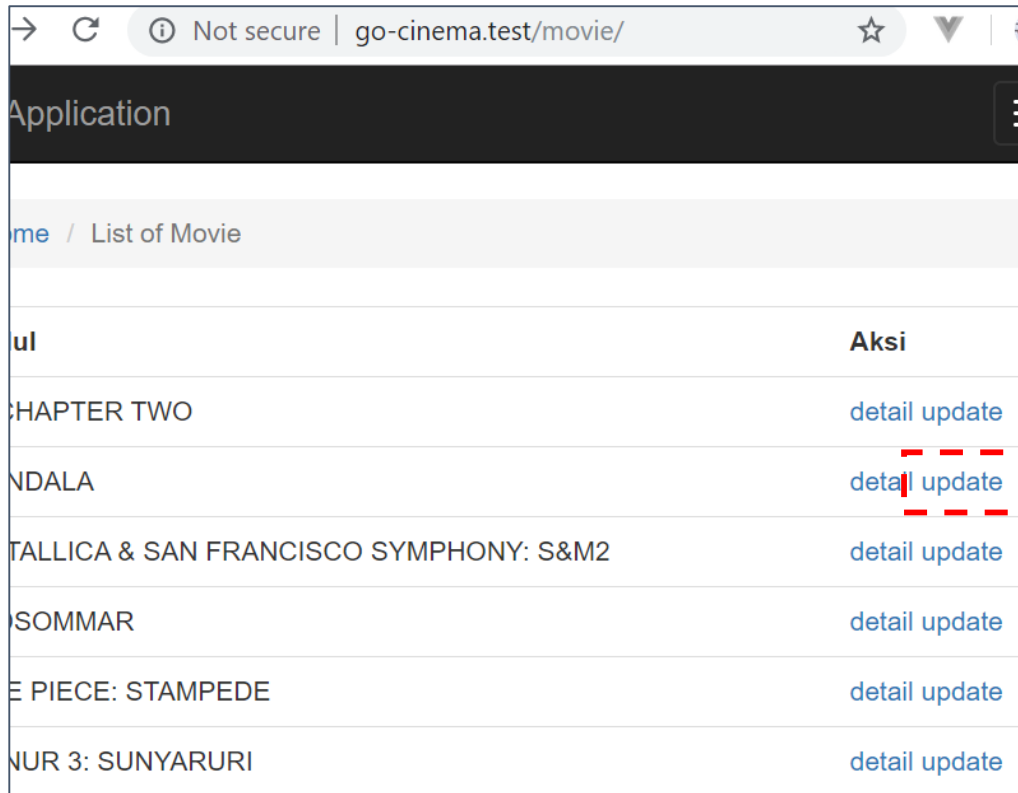
    return $this->render('form', [
        'model' => $model,
    ]);
}
```

Link Update Pada View index

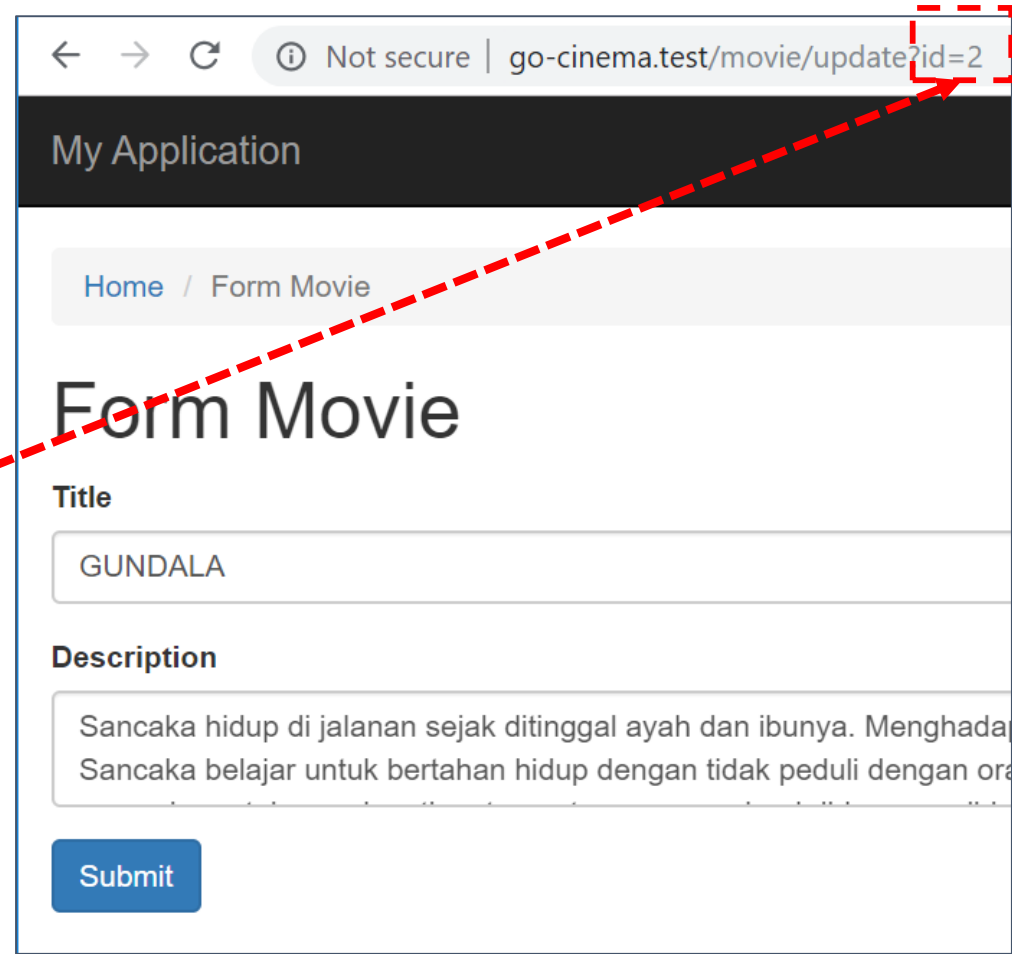
Tambahkan link update pada setiap baris film di view index, sehingga ketika diklik akan menuju ke form update film,

```
<table class="table">
<tr>
    <th>Judul</th>
    <th>Aksi</th>
</tr>
<?php
foreach($model as $movie){
    echo "<tr>";
    echo "<td>".$movie->title."</td>";
    echo "<td>".
        Html::a('detail', ['view', 'id'=>$movie->id]).
        " ".
        Html::a('update', ['update', 'id'=>$movie->id]).
        "</td>";
    echo "</tr>";
}
?>
</table>
```

Test di Browser /movie/index



Detail	Aksi
CHAPTER TWO	detail update
GUNDALA	detail update
TALLICA & SAN FRANCISCO SYMPHONY: S&M2	detail update
SOMMAR	detail update
E PIECE: STAMPEDE	detail update
NUR 3: SUNYARURI	detail update



← → ↻ ⓘ Not secure | go-cinema.test/movie/update?id=2

My Application

Home / Form Movie

Form Movie

Title

Description

Sancaka hidup di jalanan sejak ditinggal ayah dan ibunya. Menghadap
Sancaka belajar untuk bertahan hidup dengan tidak peduli dengan ora

Submit

Modifikasi actionUpdate pada MovieController

Pada actionUpdate kita perlu kode buat menangkap input data dari form dan menyimpannya. Kodanya mirip dengan actionForm yang telah kita buat di awal

```
public function actionUpdate($id)
{
    $model = Movie::find()->where([
        'id' => $id
    ])->one();

    if ($model->load(Yii::$app->request->post())) {
        $model->save();
        Yii::$app->session->setFlash('success',
            'Film '.$model->title.' berhasil disimpan');
    }

    return $this->render('form', [
        'model' => $model,
    ]);
}
```

Test di Browser /movie/update?id=2

← → ↻ ⓘ Not secure | go-cinema.test/movie/update?id=2 ☆ ▼ | [Profile Icon] ⋮

My Application [Menu Icon]

[Home](#) / Form Movie

Film GUNDALA 2 berhasil disimpan ✕

Form Movie

Title

GUNDALA 2

Description


Sancaka hidup di jalanan sejak ditinggal ayah dan ibunya. Menghadapi hidup yang keras, Sancaka belajar untuk bertahan hidup dengan tidak peduli dengan orang lain dan hanya

Submit

Kembali Ke Halaman Index Setelah Update

Modifikasi method actionUpdate agar setelah selesai simpan data langsung kembali ke halaman index (daftar film)

```
if ($model->load(Yii::$app->request->post())) {  
    $model->save();  
    Yii::$app->session->setFlash('success',  
        'Film '.$model->title.' berhasil disimpan');  
    return $this->redirect('index');  
}
```



Test di Browser /movie/update?id=2

← → ↻ ⚠ Not secure | go-cinema.test/movie/update?id=2

My Application

Home / Form Movie

Form Movie

Title

Description



← → ↻ ⓘ Not secure | go-cinema.test/movie/index

My Application

Home / List of Movie

Film GUNDALA REDIRECT berhasil disimpan

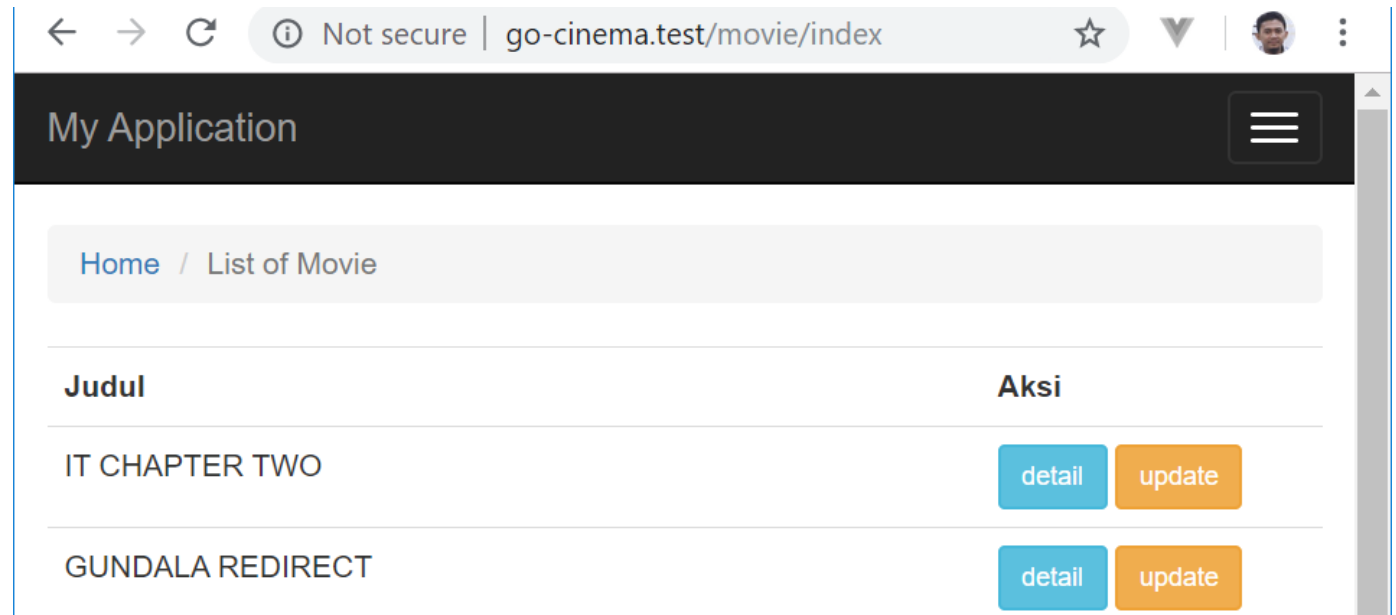
Judul	Aksi
IT CHAPTER TWO	detail update
GUNDALA REDIRECT	detail update
METALLICA & SAN FRANCISCO SYMPHONY: S&M2	detail update
MIDSOMMAR	detail update

Mempercantik Tampilan Tombol Detail & Update

Pada view index, edit tombol detail dan update dengan menambahkan class css button twitter bootstrap

Silakan merujuk ke <https://getbootstrap.com/docs/3.3/css/#buttons>

```
echo "<td>".  
    Html::a('detail', ['view', 'id'=>$movie->id],  
        ['class' => 'btn btn-info btn-sm']).  
    "  
    Html::a('update', ['update', 'id'=>$movie->id],  
        ['class' => 'btn btn-warning btn-sm']).  
"</td>";
```



Hapus Data

```
// delete data tabel movie dengan id 1  
$model = Movie::findOne(1);  
$model->delete();
```

Silakan buat fungsi hapus data!

Terima Kasih