Pengembangan Aplikasi Berbasis Yii

Chapter III

Agenda

- Melengkapi Studi Kasus untuk Fitur Customer
- Modifikasi CRUD Customer
- Autentikasi di Yii (Login & Signup)
- Otorisasi di Yii (User Level Access)
- Visualisasi Data

Melengkapi Studi Kasus untuk Fitur Customer

Fitur Customer

- Pencarian Film
- Lihat Jadwal Film
- Pemesanan Tiket Film
- Halaman Carousel Film

Panduan

- Pencarian Film (actionSearch)
  - Contek hasil generate crud Movie (gunakan list view + pjax)
- Lihat Jadwal Film (actionDetail)
  - Detail film (lihat actionView pada MovieController)
  - Jadwal film (ihat actionIndex pada ScheduleController)
- Pemesanan Tiket Film (actionBooking)
  - Lihat actionCreate pada Booking Controller
- Halaman Carousel Film (actionIndex)
  - Lihat getbootstrap carousel

Catatan: semua di SiteController

Mari Kita Kerjakan Bersama ☺

Modifikasi CRUD Customer

Perintah Dasar Membuat Data User

```
$user = new \app\models\User();
$user->username = 'admin';
$user->email = 'admin@gmail.com';

$user->password_hash = Yii::$app->security->generatePasswordHash('123456');
$user->auth_key = Yii::$app->security->generateRandomString();
$user->save();
```

Modifikasi Model Customer

- Tambahkan property $password

public $password;

- Tambahkan pada method rules, aturan ini

[['password'], 'string', 'length' => [6,25]],

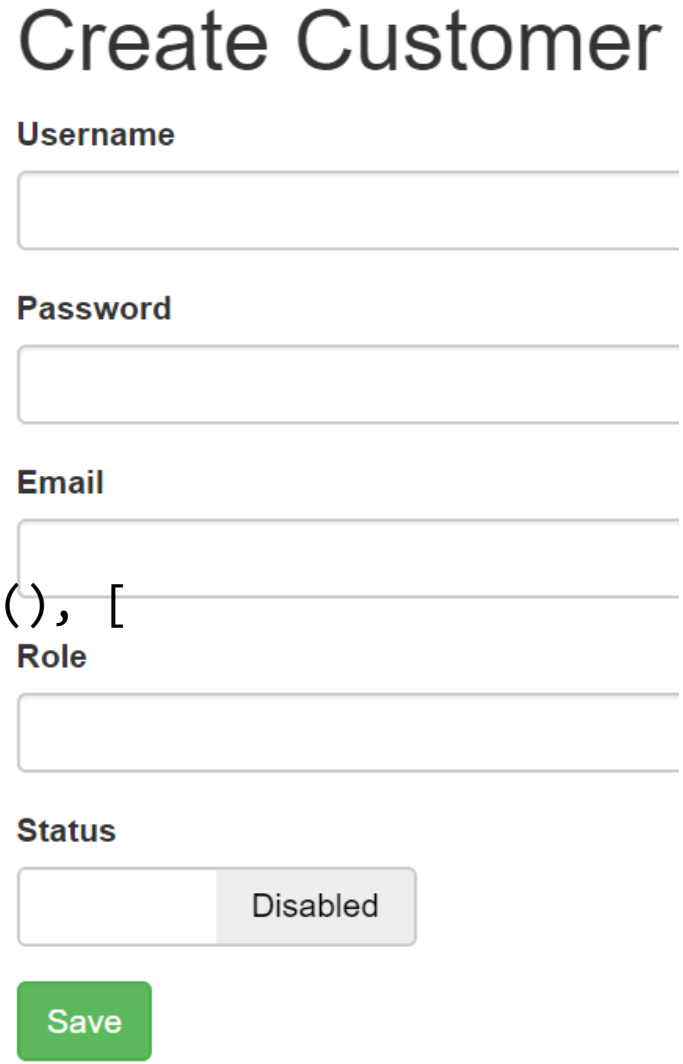
- Modifikasi method rules dimana field yang harus diisi hanya username dan email,

[['username', 'email'], 'required'],

Modifikasi View (_form) Form Customer

- Modifikasi form customer sehingga hanya 5 field: username, password, email, role, status

```
<?= $form->field($model, 'username')->textInput() ?>
<?= $form->field($model, 'password')->passwordInput() ?>
<?= $form->field($model, 'email')->textInput() ?>
<?= $form->field($model, 'role')->dropDownList([
    'customer' => 'Customer', 'admin' => 'Admin'
], ['prompt' => '']) ?>
<?php
echo $form->field($model, 'status')->widget(SwitchInput::classname(), [
    'type' => SwitchInput::CHECKBOX,
    'pluginOptions'=>[
        'handleWidth'=>60,
        'onText'=>'Active',
        'offText'=>'Disabled'
    ]
]);
?>
```



Modifikasi Controller Customer

```
public function actionCreate()
{
    $model = new Customer();
    $security = Yii::$app->security;
    if ($model->load(Yii::$app->request->post())) {
        if(!empty($model->password)){
            $model->password_hash = $security->generatePasswordHash($model->password);
        }
        $model->auth_key = $security->generateRandomString();
        if($model->save()) {
            Yii::$app->session->setFlash('success', 'berhasil disimpan');
        }
        else{
            Yii::$app->session->setFlash('error', 'gagal disimpan');
        }
        return $this->redirect(['view', 'id' => $model->id]);
    }
    // kode lainnya
}
```

Buat minimal 2 user dengan role berbeda yaitu admin & customer

Autentikasi di Yii

Login & Signup

Fitur Login

Fungsi Dasar Login & Logout Yii

- Yii::$app->user->login($user)
- Yii::$app->user->logout();

- $user merupakan objek user

```
$user = User::find()->where([
    'username' => 'hafid',
    'password_hash' => ' $2y$13$n2ZFORGMtdQJzMQah8odYO1Rclmi'
])->one();
```

- Password hash di dapat dari

```
Yii::$app->security->generatePasswordHash('123456');
```

Login

File-file yang terlibat

- Method actionLogin (SiteController)
- Model LoginForm
- View login.php (form login)
- Model User

Login dengan Database

- Modifikasi model User di @app\models\User.php
- Ubah extends \yii\base\Object menjadi \yii\db\ActiveRecord

```
class User extends  \yii\base\BaseObject
```

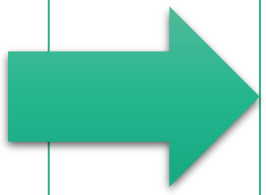
```
class User extends \yii\db\ActiveRecord
```

Login dengan Database

- Hapus semua property, ganti menjadi method static tableName (opsional)

```php
public $id;
public $username;
public $password;
public $authKey;
public $accessToken;

private static $users = [
    '100' => [
        'id' => '100',
        'username' => 'admin',
        'password' => 'admin',
        'authKey' => 'test100key',
        'accessToken' => '100-token',
    ],
```



```php
public static function tableName()
{
    return 'user';
}
```

Login dengan Database

- Ubah fungsi findIdentity & findByUserName, menjadi bentuk ActiveRecord

```php
public static function findIdentity($id)
{
    return isset(self::$users[$id]) ?
        new static(self::$users[$id]) :
        null;
}
```

```php
public static function findIdentity($id)
{
    return static::find()->where([
        'id' => $id, 'status' => 1
    ])->one();
}
```

```php
public static function findByUsername($username)
{
  foreach (self::$users as $user) {
    if(strcasecmp($user['username'],$username)===0) {
      return new static($user);
    }
  }
  return null;
}
```

```php
public static function findByUsername($username)
{
    return static::find()->where([
        'username' => $username, 'status' => 1
    ])->one();
}
```

Login dengan Database

- Sesuaikan nama property pada method getAuthKey & validateAuthKey sesuai dengan nama field di tabel user

```php
public function getAuthKey()
{
    return $this->authKey;
}
```

```php
public function getAuthKey()
{
    return $this->auth_key;
}
```

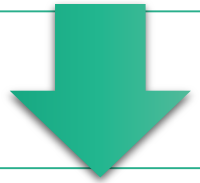
```php
public function validateAuthKey($authKey)
{
    return $this->authKey === $authKey;
}
```

```php
public function validateAuthKey($authKey)
{
    return $this->auth_key === $authKey;
}
```

Login dengan Database

- Ubah method validatePassword menggunakan standard keamanan password Yii

```
public function validatePassword($password)
{
    return $this->password === $password;
}
```



```
public function validatePassword($password)
{
    //return $this->password === $password;
    return Yii::$app->security->validatePassword($password, $this->password);
}
```

Silakan Ujicoba Login Sebagai Customer atau Admin

Fitur Register

Fungsi Dasar Register

```
$user = new \app\models\User();
$user->username = 'admin';
$user->email = 'admin@gmail.com';
$user->password_hash = Yii::$app->security->generatePasswordHash('123456');
$user->auth_key = Yii::$app->security->generateRandomString();
if($user->save()){
  Yii::$app->session->setFlash('success','registrasi sukses');
}
else{
  Yii::$app->session->setFlash('error','registrasi gagal');
}
return $this->redirect(["index"]);
```

Yang diperlukan?

- Model RegisterForm
- Method actionRegister (SiteController)
- View register.php (folder views/site)

Model RegisterForm

```php
<?php
namespace app\models;
class RegisterForm extends \yii\base\Model
{
    public $username;
    public $email;
    public $password;
    public $password_repeat;

    public function rules()
    {
        return [
            [['username', 'email', 'password', 'password_repeat'], 'required'],
            ['password', 'compare'],
        ];
    }
}
```

Method actionRegister (SiteController)

```php
public function actionRegister()
{
    if (!Yii::$app->user->isGuest) {
        return $this->goHome();
    }

    $model = new \app\models\RegisterForm();
    if ($model->load(Yii::$app->request->post())) {

        // create new customer
        $customer = new \app\models\Customer();
        $customer->username = $model->username;
        $customer->email = $model->email;
        $security = Yii::$app->security;
        $customer->password_hash = $security->generatePasswordHash($model->password);
        $customer->auth_key = $security->generateRandomString();
        $customer->role = 'customer';
        $customer->status = 1;
```

Method actionRegister (SiteController)

```
            if($customer->save()) {
                Yii::$app->session->setFlash('success', 'registrasi sukses');
            }
            else{
                Yii::$app->session->setFlash('error', 'registrasi gagal');
            }
            return $this->goBack();
        }

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

View register.php (folder views/site)

```
<?php $form = ActiveForm::begin(); ?>
<div class="row">
<div class="col-md-4">
    <?= $form->field($model, 'username')->textInput(['autofocus' => true]) ?>

    <?= $form->field($model, 'email')->textInput() ?>

    <?= $form->field($model, 'password')->passwordInput() ?>

    <?= $form->field($model, 'password_repeat')->passwordInput() ?>

    <div class="form-group">
        <?= Html::submitButton('Register', ['class' => 'btn btn-success']) ?>
    </div>
</div>
</div>
<?php ActiveForm::end(); ?>
```

Autorisasi di Yii

User Level Access

Pendahuluan

- Otorisasi pengguna adalah pengaturan hak akses *pengguna* terhadap aplikasi. Hal ini berbeda dengan otentikasi yang hanya mengecek apakah *pengguna* sudah *login* atau belum saja, *otorisasi* mengatur apa yang boleh dan tidak boleh dilakukan oleh *pengguna* yang telah *login* atau belum pada aplikasi
- *Otorisasi* cocok diterapkan untuk aplikasi yang memiliki beberapa level *pengguna*, misalnya level *guest* yang hanya bisa *view*, level *member* bisa mengunduh, dan level *admin* yang bisa melakukan apapun yang bisa dilakukan oleh dua level *pengguna* sebelumnya.
- Yii menyediakan dua metode *otorisasi*, yaitu *Access Control Filter* (ACF) dan *Role-Based Access Control* (RBAC). Pada dasarnya keduanya sama-sama *access control*,

ACF (Access Control Filter)

- ACF adalah metode *otorisasi* yang sederhana yang cocok digunakan untuk aplikasi yang membutuhkan pengaturan hak akses yang *simple*. Sesuai dengan namanya, ACF bekerja dengan menyaring setiap *action* pada suatu *controller* ketika *pengguna* melakukan *request* terhadap suatu *action*.
- Class yang digunakan [[yii\filters\AccessControl]]
- Penerapannya pada Controller fungsi behaviors()

Mengatur Hak Akses Controller (Semua)

```
public function behaviors()
{
    return [
        'access' => [
            'class' => \yii\filter\AccessControl::className(),
            'rules' => [
                [
                    'allow' => true,
                    'roles' => ['@'],
                ],
            ],
        ],
    ];
}
```

Catatan:
Roles ['@'] artinya user yang sudah login yang diizinkan mengakses, pada contoh ini semua action pada controller ini hanya bisa diakses oleh user yang sudah login

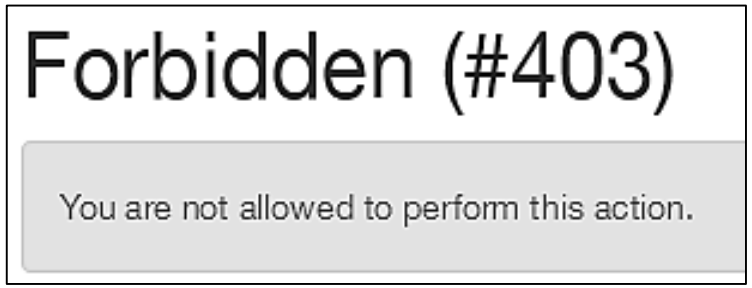
Mengatur Hak Akses Controller (Sebagian)

```
public function behaviors()
{
    return [
        'access' => [
            'class' => \yii\filter\AccessControl::className(),
            'rules' => [
                [
                    'actions' => ['index', 'view'],
                    'allow' => true,
                    'roles' => ['?'],
                ],
                [
                    'allow' => true,
                    'roles' => ['@'],
                ],
            ],
        ],
    ];
```

Catatan:
Roles ['?'] artinya baik user sudah login maupun belum login, semua diizinkan mengakses, pada contoh ini semua action pada controller ini hanya bisa diakses oleh user yang sudah login, kecuali index dan view

Fungsi Balik dari ACF

- Ada dua hal yang akan terjadi ketika *pengguna* mengakses suatu *action* yang mana dia tidak memiliki hak akses akan *action* tersebut.
  1. Jika *pengguna* belum *login* atau *guest*, maka ACF akan memanggil fungsi yii\web\User::loginRequired() untuk mengalihkan alamat *web browser* ke halaman *login*.
  2. Jika *pengguna* sudah *login* atau *authenticated user* maka ACF akan menjalankan [[yii\web\ForbiddenHttpException]].



Multi Level User menggunakan ACF

- Untuk kasus yang sederhana misalnya kita mempunya tiga level *user* yaitu *guest*, *customer* dan *admin*, maka kita bisa memaksimalkan ACF yaitu dengan menggunakan parameter matchCallback pada *rules*
- Misalnya, untuk membedakan level user, kita menggunakan field role pada tabel user yang isinya "customer" atau "admin"

```
[
    'allow' => true,
    'roles' => ['@'],
    'matchCallback' => function($rule, $action){
        $user = \Yii::$app->user->identity;
        if ($user->role == 'admin'){
            return true;
        }
    }
],
```

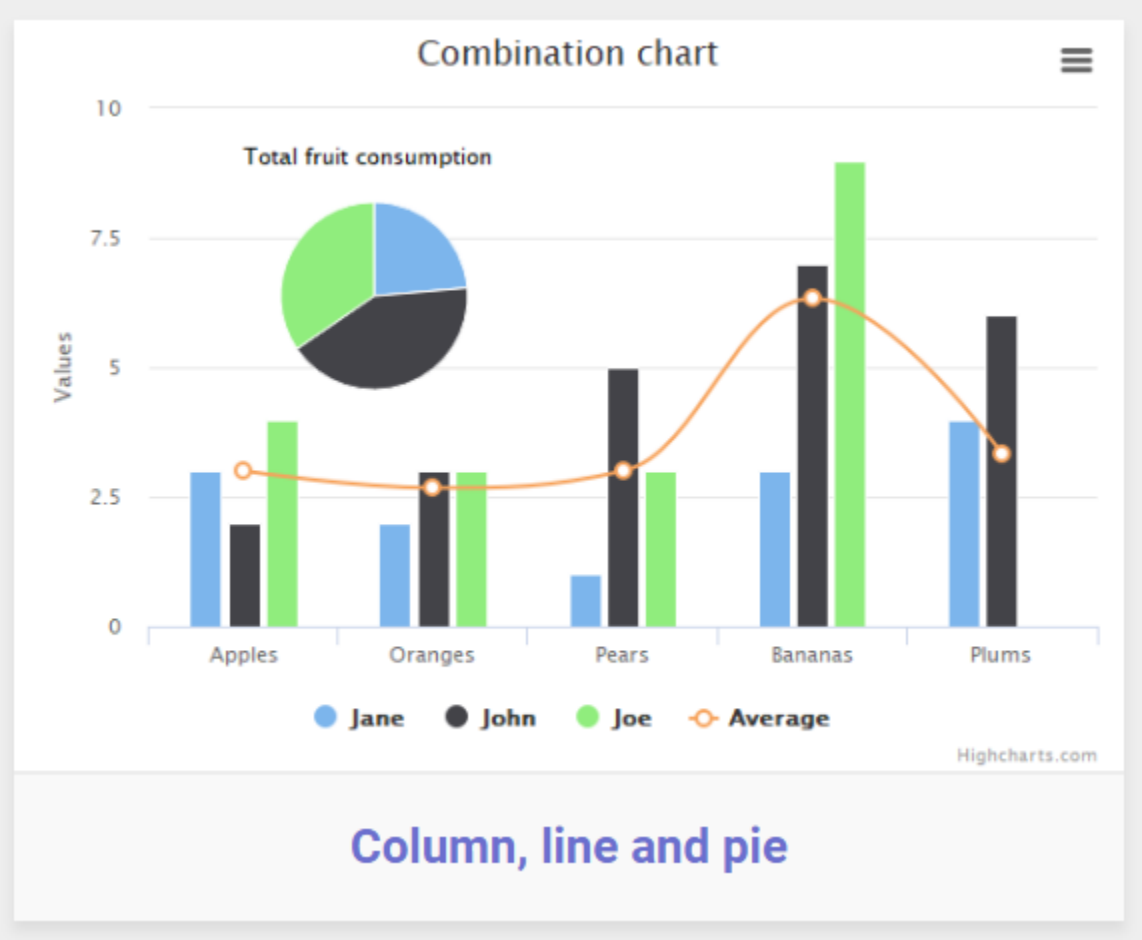
Lakukan Proteksi Akses Untuk Fitur Manage Customer, Movie, Schedule, Booking menggunakan ACF

Visualisasi Data

Visualiasi Data

- Visualisasi data yang dimaksud di sini adalah menampilkan data dalam bentuk grafik atau chart. Ada banyak library yang bisa digunakan untuk membuat tampilan grafik, salah satunya yang cukup populer adalah Highchart (http://www.highcharts.com/). Library ini berbayar untuk tujuan proyek aplikasi commercial.
- Highchart banyak digunakan oleh perusahaan besar, semisal Facebook, Twitter, Yahoo, dsb.

Contoh Tampilan Demo Highchart

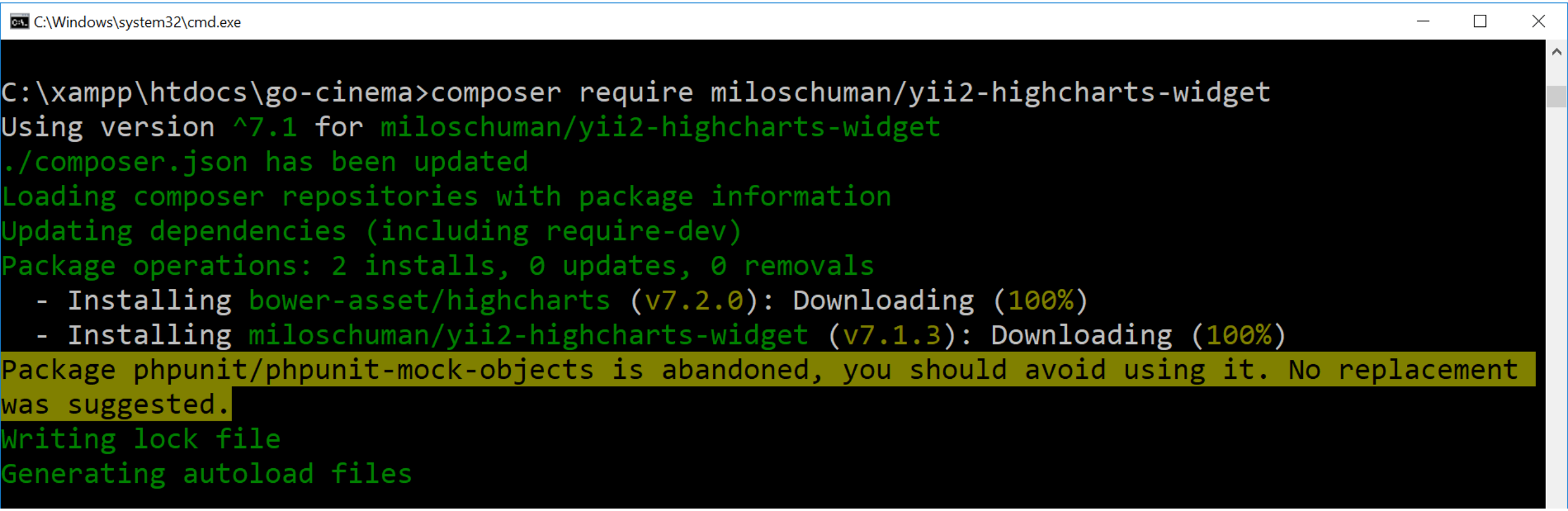


Install Highchart pada Yii2

- Salah satu extension Yii yang mem-wrap library ini dan cukup populer adalah yii2-highcharts (https://github.com/miloschuman/yii2-highcharts). Tidak hanya Highchart, kita juga bisa mengintegrasikan library lain yang berafiliasi dengan Highchart yaitu Highstock dan Highmaps.

```
composer require miloschuman/yii2-highcharts-widget
```

Progress Instalasi Highchart

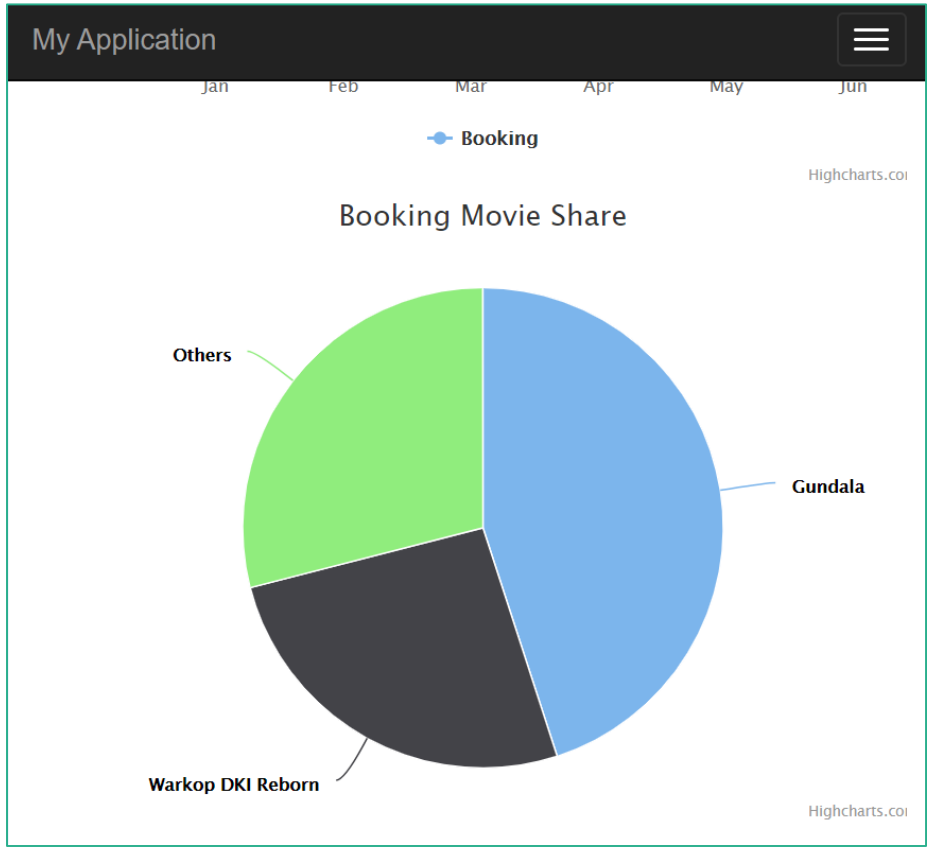


Implementasi Pada View - Line Chart

```php
<?php
use miloschuman\highcharts\Highcharts;
echo Highcharts::widget([
    'options' => [
    'chart' => ['type' => 'line' /* column */],
        'title' => ['text' => 'Booking Tickets Growth'],
        'xAxis' => [
            'categories' => ['Jan', 'Feb', 'Mar', 'Apr',
                             'May', 'Jun']
        ],
        'yAxis' => [
            'title' => ['text' => 'Total Booking']
        ],
        'series' => [
            ['name' => 'Booking', 'data' => [36, 79, 45, 83, 91, 102]],
        ]
    ]
]);
```

Implementasi Pada View - Pie Chart

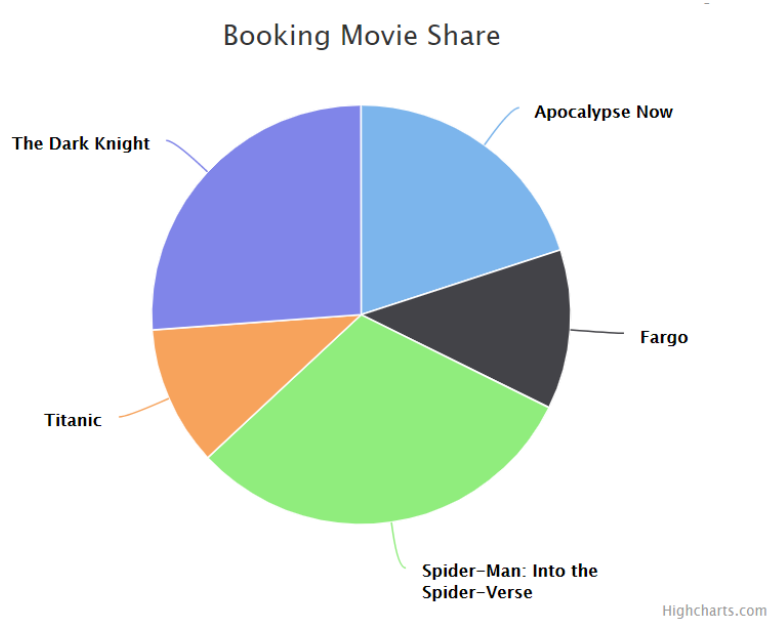
```php
<?php
use miloschuman\highcharts\Highcharts;
echo Highcharts::widget([
    'options' => [
        'title' => ['text' => 'Booking Movie Share'],
        'series' => [
            [
                'type' => 'pie',
                'name' => 'Movie',
                'data' => [
                    ['Gundala', 45],
                    ['Warkop DKI Reborn', 26],
                    ['Others', 29]
                ],
            ]
        ],
    ]
]);
```



Dynamic Data Chart

```php
public function actionCharts()
{
    $movies = \app\models\Movie::find()
        ->select('title, id AS sales')
        ->asArray()
        ->orderBy('RAND()')
        ->limit(5)
        ->all();
    $movie_data = [];
    foreach($movies as $movie){
        $movie_data[] = [
            $movie['title'], (int)$movie['sales'],
        ];
    }
    return $this->render('charts', [
        'movie_data' => $movie_data
    ]);
}
```

```php
        'series' => [
            [
                'type' => 'pie',
                'name' => 'Movie',
                'data' => $movie_data,
            ]
        ],
```



Terima Kasih