

Pengembangan Aplikasi Berbasis Yii

Chapter II

Agenda

- ❖ Mempersiapkan Struktur Database
- ❖ Menggunakan Code Generator
- ❖ Modifikasi Hasil Code Generator
- ❖ Form Validasi
- ❖ Upload File
- ❖ Menggunakan Extension

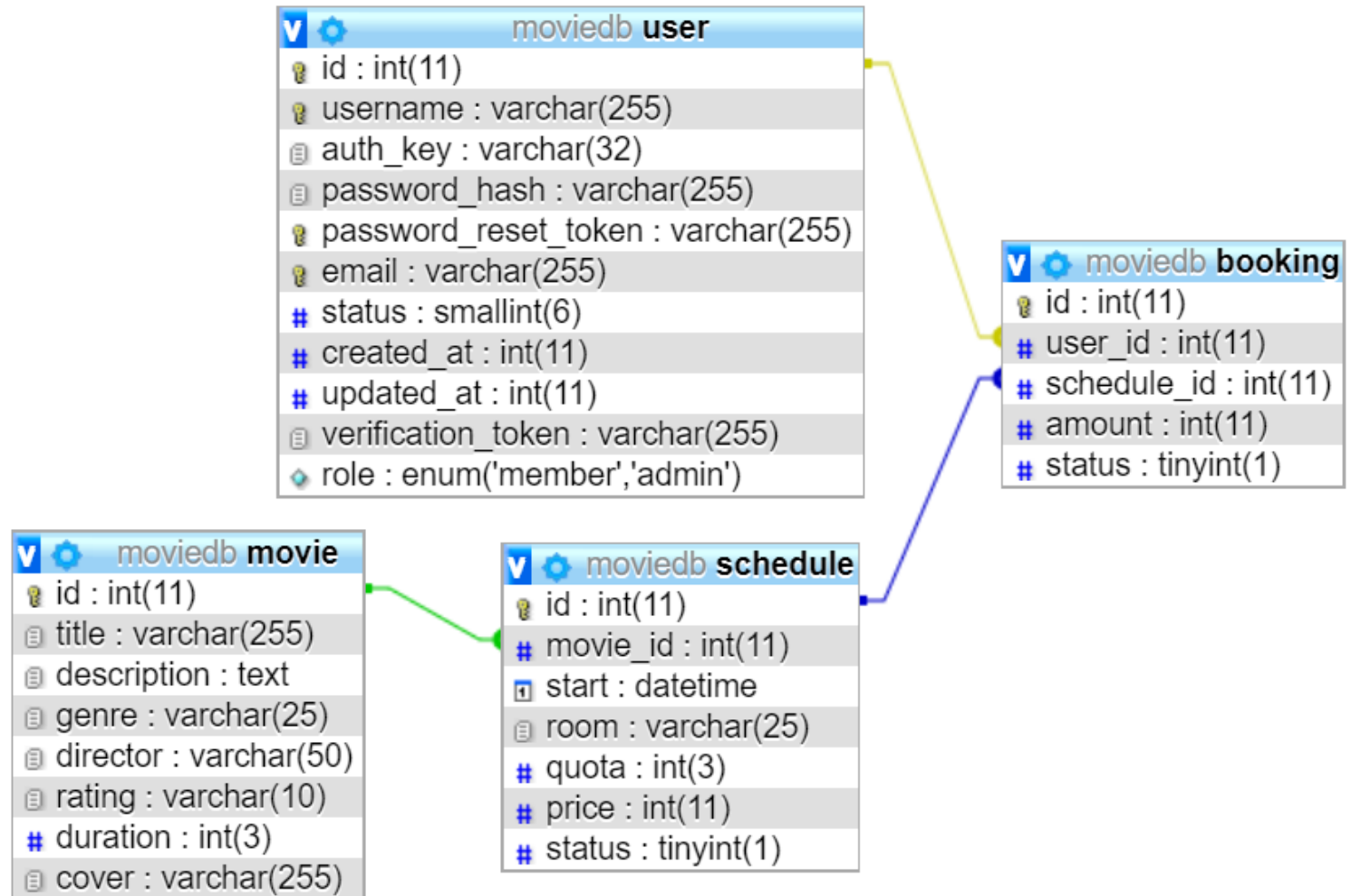
<https://github.com/hscstudio/workshop-yii-palembang>

Mempersiapkan Struktur Database


Nama database: go-cinemadb

Tabel Pada Database go-cinemadb



- Tabel movie
- Tabel schedule
- Tabel booking
- Tabel user






Struktur Tabel Movie

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	<i>None</i>		AUTO_INCREMENT
2	title	varchar(255)	latin1_swedish_ci		No	<i>None</i>		
3	description	text	latin1_swedish_ci		Yes	<i>NULL</i>		
4	genre	varchar(25)	latin1_swedish_ci		Yes	<i>NULL</i>		
5	director	varchar(50)	latin1_swedish_ci		Yes	<i>NULL</i>		
6	rating	varchar(10)	latin1_swedish_ci		Yes	<i>NULL</i>		
7	duration	int(3)			Yes	<i>NULL</i>		
8	cover	varchar(255)	latin1_swedish_ci		Yes	<i>NULL</i>		

Struktur Tabel Schedule

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	<i>None</i>		AUTO_INCREMENT
2	movie_id 	int(11)			No	<i>None</i>		
3	start	datetime			No	<i>None</i>		
4	room	varchar(25)	latin1_swedish_ci		No	<i>None</i>		
5	quota	int(3)			No	<i>None</i>		
6	price	int(11)			No	<i>None</i>		
7	status	tinyint(1)			No	<i>None</i>		

Struktur Tabel Booking

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 	int(11)			No	<i>None</i>		AUTO_INCREMENT
2	user_id 	int(11)			No	<i>None</i>		
3	schedule_id 	int(11)			No	<i>None</i>		
4	amount	int(11)			No	<i>None</i>		
5	status	tinyint(1)			No	0		

Struktur Tabel User

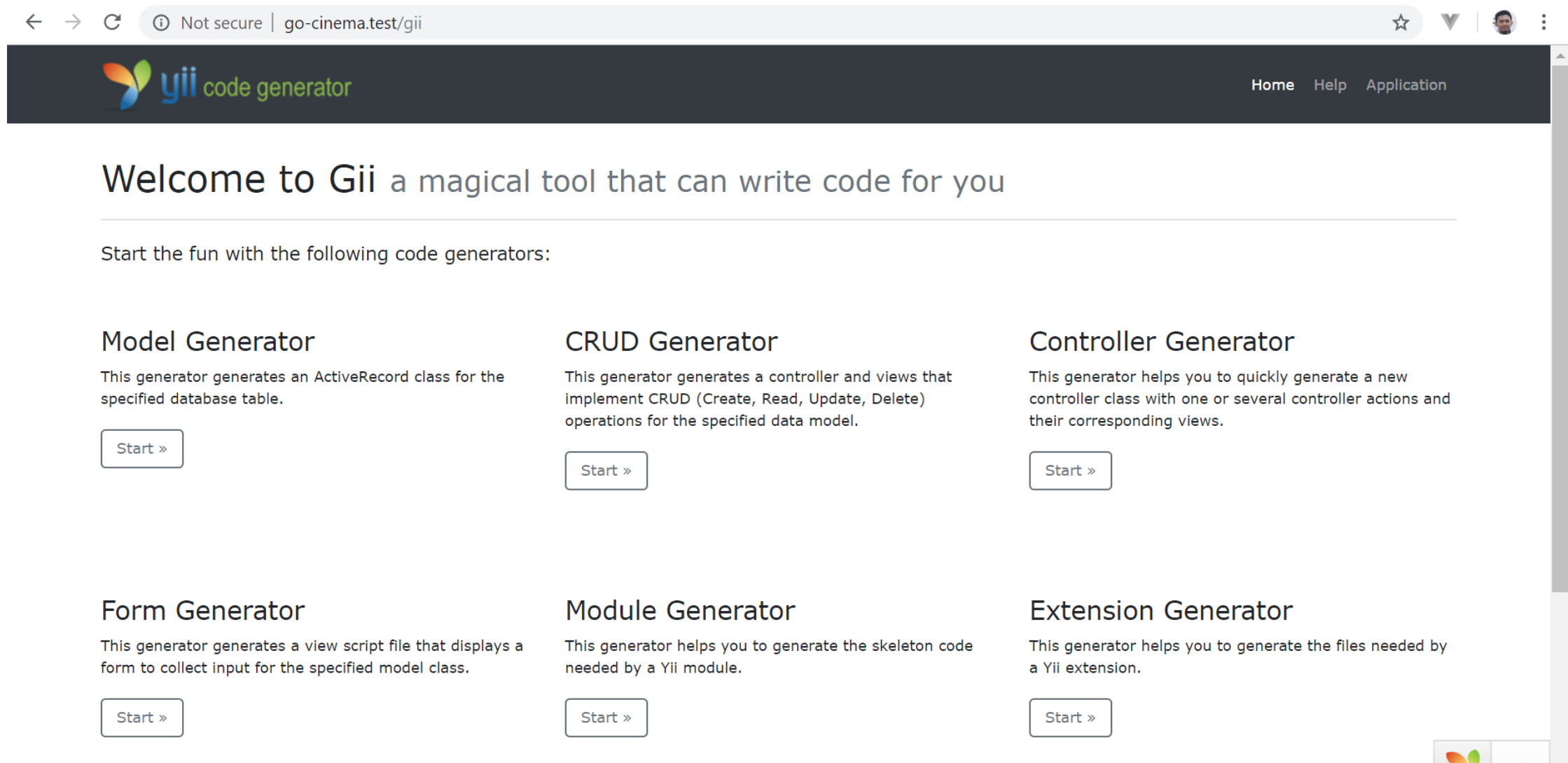
#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra
1	id 🔑	int(11)			No	None		AUTO_INCREMENT
2	username 🔑	varchar(255)	utf8_unicode_ci		No	None		
3	auth_key	varchar(32)	utf8_unicode_ci		No	None		
4	password_hash	varchar(255)	utf8_unicode_ci		No	None		
5	password_reset_token 🔑	varchar(255)	utf8_unicode_ci		Yes	NULL		
6	email 🔑	varchar(255)	utf8_unicode_ci		No	None		
7	status	smallint(6)			No	10		
8	created_at	int(11)			No	None		
9	updated_at	int(11)			No	None		
10	verification_token	varchar(255)	utf8_unicode_ci		Yes	NULL		
11	role	enum('member', 'admin')	utf8_unicode_ci		No	member		

Menggunakan Code Generator

Gii: Code Generator Yii

- Gii = tools official Yii dalam bentuk module untuk menggenerate kode program seperti model, view, controller, dsb.
- Gii bisa diakses melalui <http://go-cinema.test/gii>

http://go-cinema.test/gii



The screenshot shows a web browser window with the address bar displaying "http://go-cinema.test/gii". The page header features the "yii code generator" logo on the left and navigation links "Home", "Help", and "Application" on the right. The main content area has a large heading "Welcome to Gii a magical tool that can write code for you" followed by a subheading "Start the fun with the following code generators:". Below this, there are six generator cards arranged in a 2x3 grid. Each card has a title, a brief description, and a "Start »" button.

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

yii code generator Home Help Application

Welcome to Gii a magical tool that can write code for you

Start the fun with the following code generators:

Model Generator

This generator generates an ActiveRecord class for the specified database table.

Start »

CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, Delete) operations for the specified data model.

Start »

Controller Generator

This generator helps you to quickly generate a new controller class with one or several controller actions and their corresponding views.

Start »

Form Generator

This generator generates a view script file that displays a form to collect input for the specified model class.

Start »

Module Generator

This generator helps you to generate the skeleton code needed by a Yii module.

Start »

Extension Generator

This generator helps you to generate the files needed by a Yii extension.

Start »

6 Generator Gii

- ❑ **Model Generator**

Untuk generate class ActiveRecord (model) untuk tabel tertentu pada basis data.

- ❑ **CRUD Generator**

Untuk generate operasi CRUD berdasarkan class model tertentu

- ❑ **Controller Generator**

Untuk generate class Controller

- ❑ **Form Generator**

Untuk generate tampilan formulir berdasarkan class model tertentu

- ❑ **Module Generator**

Untuk generate template dari module di Yii, jika kita ingin membuat sub aplikasi.

- ❑ **Extension Generator**

Untuk generate template dari extension Yii, untuk membuat extension sendiri.

Model Generator

Model Generator >

CRUD Generator >

Controller Generator >

Form Generator >

Module Generator >

Extension Generator >

Model Generator

This generator generates an ActiveRecord class for the specified database table.

Table Name

movie

Model Class Name

Movie

☐ Standardize Capitals

Code Template

```
default (C:\xampp\htdocs\movieapp\vendor\yiisoft\yii2-gii\src\generators\model\def...
```

Preview

Model Generator

Preview

Generate

Click on the above **Generate** button to generate the files selected below:

Type to filter

☒ Create

☒ Unchanged

☒ Overwrite

Code File	Action	<input checked="" type="checkbox"/>
models\Movie.php diff	overwrite	<input checked="" type="checkbox"/>

Preview

The code has been generated successfully.

Generating code using template "C:\xampp\htdocs\movieapp\vendor\yiisoft\yii2-gii\src\generators\model/default"...
overwrote models\Movie.php
done!

Cek Hasilnya di models/Movie.php

```
<?php

namespace app\models;

use Yii;

/**
 * This is the model class for table "movie".
 *
 * @property int $id
 * @property string $title
 * @property string $description
 * @property string $genre
 * @property string $director
 * @property string $rating
 * @property int $duration
 * @property string $cover
 *
 * @property Schedule[] $schedules
 */
class Movie extends \yii\db\ActiveRecord
{
    /**
     * {@inheritdoc}
     */
    public static function tableName()
    {
        return 'movie';
    }

    /**
     * {@inheritdoc}
     */
    public function rules()
    {
        return [
            [['title'], 'required'],
```

```
            [['description'], 'string'],
            [['duration'], 'integer'],
            [['title', 'cover'], 'string', 'max' => 255],
            [['genre'], 'string', 'max' => 25],
            [['director'], 'string', 'max' => 50],
            [['rating'], 'string', 'max' => 10],
        ];
    }

    /**
     * {@inheritdoc}
     */
    public function attributeLabels()
    {
        return [
            'id' => 'ID',
            'title' => 'Title',
            'description' => 'Description',
            'genre' => 'Genre',
            'director' => 'Director',
            'rating' => 'Rating',
            'duration' => 'Duration',
            'cover' => 'Cover',
        ];
    }

    /**
     * @return \yii\db\ActiveQuery
     */
    public function getSchedules()
    {
        return $this->hasMany(Schedule::className(), ['movie_id' => 'id']);
    }
}
```

CRUD Generator

Model Generator >

CRUD Generator >

Controller Generator >

Form Generator >

Module Generator >

Extension Generator >

CRUD Generator

This generator generates a controller and views that implement CRUD (Create, Read, Update, model).

Model Class

app\models\Movie ←

Search Model Class

app\models\MovieSearch ←

Controller Class

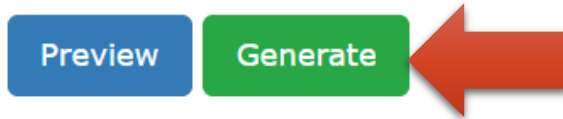
app\controllers\MovieController ←

Code Template

default (C:\xampp\htdocs\movieapp\vendor\yiisoft\yii2-gii\src\generators\crud\defa...

Preview ←

CRUD Generator



Click on the above **Generate** button to generate the files selected below:

☒ Create

☒ Unchanged

☒ Overwrite

Code File	Action	<input checked="" type="checkbox"/>
controllers\MovieController.php	unchanged	
models\MovieSearch.php	create	<input checked="" type="checkbox"/>
views\movie_form.php	create	<input checked="" type="checkbox"/>
views\movie_search.php	create	<input checked="" type="checkbox"/>
views\movie\create.php	create	<input checked="" type="checkbox"/>
views\movie\index.php	unchanged	
views\movie\update.php	create	<input checked="" type="checkbox"/>
views\movie\view.php	unchanged	

CRUD Generator

Preview

The code has been generated successfully.

```
Generating code using template "C:\xampp\htdocs\movieapp\vendor\yiisoft\yii2-gii\src\generators\crud/default"...  
skipped controllers\MovieController.php  
generated models\MovieSearch.php  
generated views\movie\_form.php  
generated views\movie\_search.php  
generated views\movie\create.php  
skipped views\movie\index.php  
generated views\movie\update.php  
skipped views\movie\view.php  
done!
```

Cek Hasilnya

- @app/models/MovieSearch.php
- @app/views/movie/index.php
- @app/views/movie/view.php
- @app/views/movie/create.php
- @app/views/movie/update.php
- @app/views/movie/_form.php
- @app/views/movie/_search.php
- @app/controllers/MovieController.php

Ujicoba Hasil CRUD Generator

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

Go Cinema










HomeAboutContactLogin

Beranda / Movies

Movies

Create Movie

Menampilkan 1-20 dari 21 item.

#	ID	Title	Description	Genre	Director	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	1	The Lord of the Rings: The Return of the King	Gandalf and Aragorn lead the World of Men against Sauron's army to draw his gaze from Frodo and Sam as they approach Mount Doom with the One Ring.	2003	Peter Jackson	  
2	2	The Pianist	A Polish Jewish musician struggles to survive the destruction of the Warsaw ghetto of World War II.	2002	Roman Polanski	  
3	3	Gladiator	A former Roman General sets out to exact vengeance against the corrupt emperor who murdered his family and sent him into slavery.	2000	Ridley Scott	  

Ujicoba Hasil CRUD Generator

Go Cinema

[Beranda](#) / [Movies](#) / Create Movie

Create Movie

Title

Description

Genre

Director

Rating

Go Cinema

[Beranda](#) / [Movies](#) / The Lord of the Rings: The Return of the King

The Lord of the Rings: The Return of the King

UpdateDelete

ID	1
Title	The Lord of the Rings: The Return of the King
Description	Gandalf and Aragorn lead the World of Men against Sauron's Army to reclaim Middle-earth.
Genre	2003
Director	Peter Jackson
Rating	
Duration	0
Cover	the-lord-of-the-rings.jpg

Go Cinema

[Beranda](#) / [Movies](#) / The Lord of the Rings: The Return of the King

Update Movie: The Lord of the Rings: The Return of the King

Title

Description

Genre

Director

Generate Tabel Lainnya

- Schedule
- Booking
- User
 - Khusus tabel user, nama modelnya diganti dengan customer

Model Generator

This generator generates an ActiveRecord class for the specified database table.

Table Name

user

Model Class Name

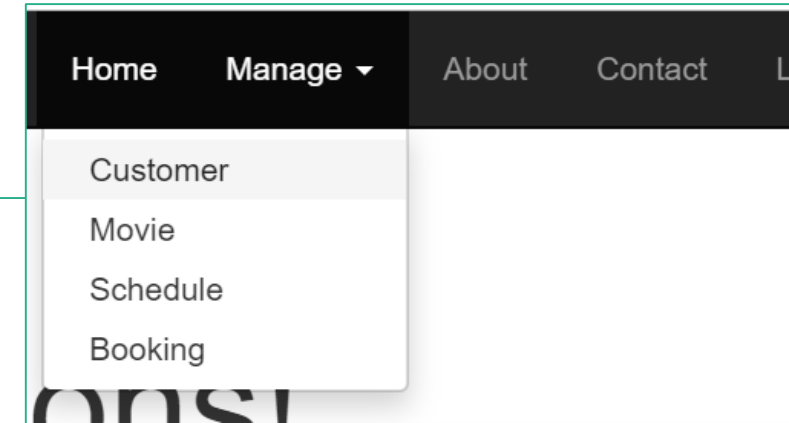
Customer

Modifikasi Code Hasil Generator

Modifikasi View: Menambahkan Menu

- @app\view\layouts\main.php

```
echo Nav::widget([
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => [
        ['label' => 'Home', 'url' => ['/site/index']],
        ['label' => 'Manage', 'items' => [
            ['label' => 'Customer', 'url' => ['/customer/index']],
            ['label' => 'Movie', 'url' => ['/movie/index']],
            ['label' => 'Schedule', 'url' => ['/movie/index']],
            ['label' => 'Booking', 'url' => ['/movie/index']],
        ]
    ]
]),
```



Modifikasi View: Form

File @app\views\movie_form.php

- **textInput**

```
<?= $form->field($model, 'username')->textInput([
    'autofocus' => true
]) ?>
```

- **passwordInput**

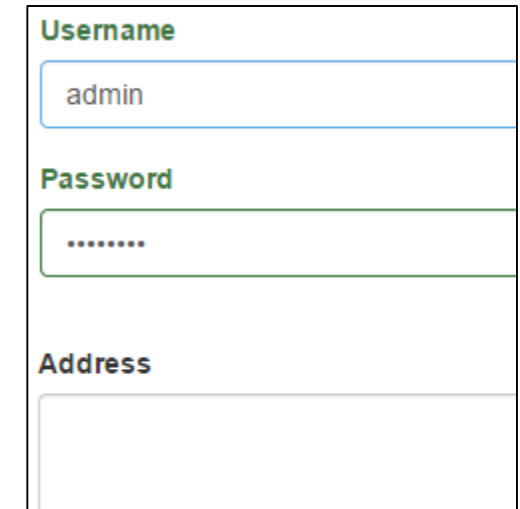
```
<?= $form->field($model, 'password')->passwordInput() ?>
```

- **textarea**

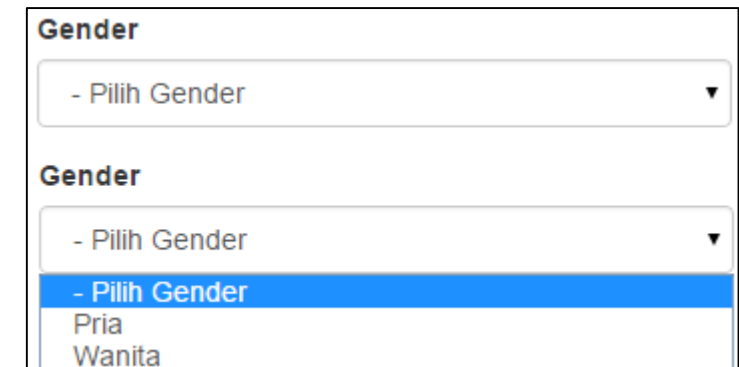
```
<?= $form->field($model, 'body')->textarea([
    'rows' => 3
]) ?>
```

- **dropDownList**

```
<?= $form->field($model, 'gender')->dropDownList([
    '1' => 'Pria', '0' => 'Wanita',
], ['prompt' => '- Pilih Gender']) ?>
```



A form preview with three input fields. The first field is labeled 'Username' and contains the text 'admin'. The second field is labeled 'Password' and contains a series of dots. The third field is labeled 'Address' and is empty.



A form preview showing a dropdown menu labeled 'Gender'. The dropdown menu is open, showing a list of options: '- Pilih Gender', 'Pria', and 'Wanita'. The option '- Pilih Gender' is currently selected and highlighted in blue.

Modifikasi View: Form

- **radioList**

```
<?= $form->field($model, 'gender')->radioList([  
    '1' => 'Pria', '0' => 'Wanita',  
]) ?>
```



Gender

☒ Pria ☐ Wanita

- **listBox -> multiple**

```
<?= $form->field($model, 'hoby')->listBox([  
    1=>'Jalan2', 2 => 'Makan2', 3=>'Joging'  
], ['multiple'=>true]) ?>
```



Hobi

Jalan2
Makan2
Joging

- **checkboxList -> multiple**

```
<?= $form->field($model, 'hoby')->checkboxList([  
    1=>'Jalan2', 2 => 'Makan2', 3=>'Joging'  
], ['multiple'=>true]) ?>
```



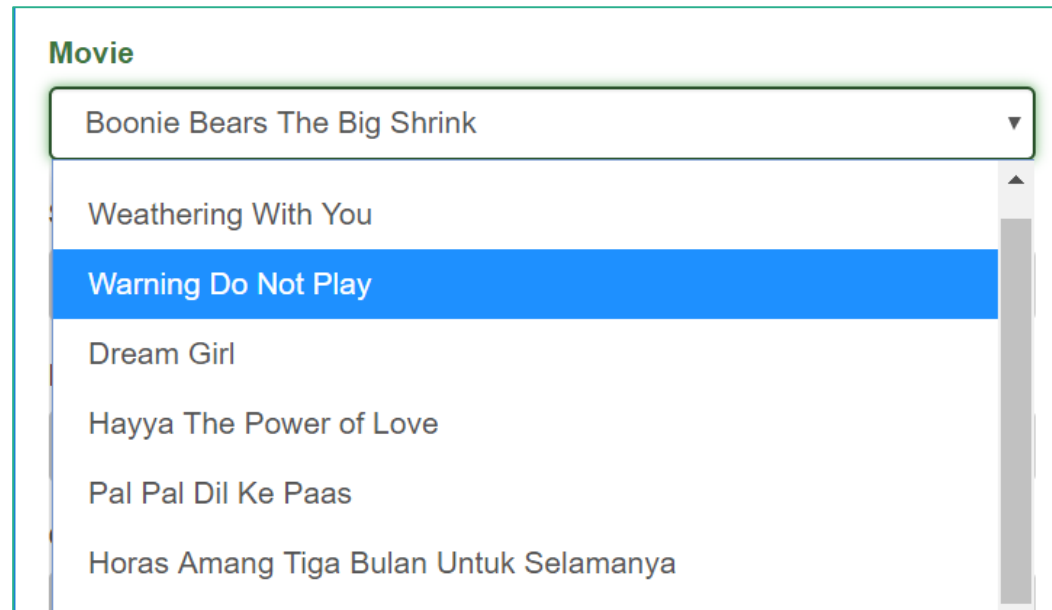
Hobi

☒ Jalan2 ☐ Makan2 ☒ Joging

Modifikasi View: Form

- dropDownList dari Tabel

```
<?php
use \yii\helpers\ArrayHelper;
use \app\models\Movie;
//...
$movies = ArrayHelper::map(
    Movie::find()->all(), "id", "title");
echo $form->field($model, 'movie_id')
    ->dropDownList($movies);
?>
```



Movie

Boonie Bears The Big Shrink

Weathering With You

Warning Do Not Play

Dream Girl

Hayya The Power of Love

Pal Pal Dil Ke Paas

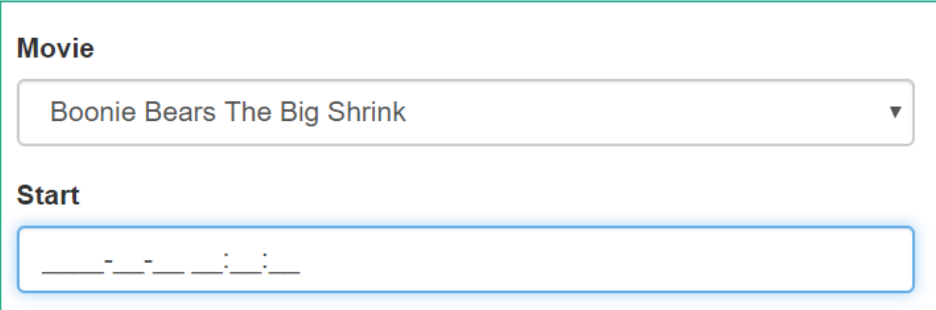
Horas Amang Tiga Bulan Untuk Selamanya

Modifikasi View: Form

- Masking input via widget

```
<?php
use \yii\widgets\MaskedInput;
echo $form->field($model, 'start')->widget(
    MaskedInput::classname(), [
        'mask' => '9999-99-99 99:99:99',
    ]
);
?>
```

- a: karakter alpha (A-Z, a-z)
- 9: karakter numeric (0-9)
- *: karakter alphanumeric (A-Z, a-z, 0-9)
- <https://github.com/RobinHerbots/jquery.inputmask>



The screenshot shows a web form with two fields. The first field, labeled 'Movie', is a dropdown menu with the text 'Boonie Bears The Big Shrink' and a downward arrow. The second field, labeled 'Start', is a text input field with a blue border and a light blue background. It contains a masked date and time format: '____-__-__ __:__:__'. The mask uses '9' for numeric digits, '-' for hyphens, and ':' for colons.

Modifikasi View: Form

- Mengatur posisi field


```
<div class="row">
  <div class="col-md-3">
    <?= $form->field($model, 'born')->textInput();
  </div>
  <div class="col-md-3">
    <?php
    use \yii\widgets\MaskedInput;
    echo $form->field($model, 'birth_day')->widget(
      MaskedInput::classname(), [ 'mask'=>'9999-99-99' ]
    );
  ?>
</div>
</div>
```

Name	
<input type="text"/>	
Born	Birth Day
<input type="text"/>	<input type="text"/>
Gender	
<input type="text"/>	

Modifikasi View: GridView

File @app\views\movie\index.php

```
<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        ['class' => 'yii\grid\SerialColumn'],
        'title',
        'genre',
        ['class' => 'yii\grid\ActionColumn'],
    ],
]); ?>
```






















Go Cinema 

Beranda / Movies

Movies

Create Movie

Menampilkan 1-20 dari 21 item.

#	Title	Genre	
	<input type="text"/>	<input type="text"/>	
1	The Lord of the Rings: The Return of the King	2003	  
2	The Pianist	2002	  
3	Gladiator	2000	  
4	The Green Mile	1999	  
5	Saving Private Ryan	1998	  
6	Good Will Hunting	1997	  
7	Titanic	1997	  

Modifikasi View: GridView

- Menampilkan format tanggal

```
[  
    'attribute'=>'birth_day',  
    'format'=>['date','php:d F Y'],  
    // http://php.net/manual/en/function.date.php  
],
```
- Menampilkan data menggunakan closure

```
[  
    'attribute'=>'gender',  
    'value'=>function($data){  
        return $data->gender==1?"Pria":"Wanita";  
    }  
],
```

Birth Day	Gender
<input type="text"/>	<input type="text"/>
30 April 1986	Pria

Modifikasi View: GridView

- Menampilkan data berelasi menggunakan closure

```
[  
    'attribute'=>'unit_id',  
    'header'=>'Unit',  
    'value'=>function($data){  
        $unit = \app\models\Unit::findOne(['id'=>$data->unit_id]);  
        return $unit->name;  
    }  
],
```

Unit
<input type="text"/>
Direktorat Jenderal Pajak

Modifikasi View: GridView

- Menampilkan link download menggunakan `\yii\helpers\Html`

```
[  
    "attribute" => "certificate",  
    "format" => "html",  
    "value" => function($data){  
        return Html::a("Unduh","uploads/" . $data->certificate);  
    }  
]
```

Modifikasi View: GridView

- Modifikasi Action Column

```
[  
    "class"    => "\yii\grid\ActionColumn",  
    "header"   => "Aksi", // set judul kolom  
    "template" => "{view} {update} {delete}", // set tombol yang muncul  
    "options"  => [  
        "width" => "80px", // set lebar kolom  
    ]  
]
```

Modifikasi View: Gridview

- Modifikasi filter menjadi dropdown
 - [
 "attribute" => "gender",
 "filter" => [1=>"Pria", 0=>"Wanita"],
]
- Menonaktifkan filter
 - [
 "attribute" => "birth_day",
 "filter" => false,
]



Modifikasi View: DetailView

- Hampir sama dengan Gridview

Modifikasi Controller: Flash Message

- Tambahkan set flash message pada pemrosesan data
 - `Yii::$app->session->setFlash('success', 'pesan berhasil disimpan');`
 - `Yii::$app->session->setFlash('error', 'pesan gagal disimpan');`

```
if($model->save()) { // jika berhasil disimpan
    // set flash success
}
else{
    // set flash error
}
```

Modifikasi Controller: Flash Message

```
/*
if ($model->load(Yii::$app->request->post()) && $model->save()) {
    return $this->redirect(['view', 'id' => $model->id]);
}
*/
if ($model->load(Yii::$app->request->post())) {
    if ($model->save()){
        Yii::$app->session->setFlash('success', 'berhasil disimpan!');
        return $this->redirect(['view', 'id' => $model->id]);
    }
    else{
        Yii::$app->session->setFlash('error', 'gagal disimpan!');
    }
}
```
















Modifikasi Controller:

Change Initial Sort & Perpage Gridview

```
public function actionIndex()
{
    $searchModel = new MovieSearch();
    $dataProvider = $searchModel->search(Yii::$app->request->queryParams);
    /* change default sort */
    $dataProvider->sort->defaultOrder = [
        'title' => SORT_ASC
    ];
    // change perpage
    $dataProvider->pagination->pageSize = 10;
    return $this->render('index', [
        'searchModel' => $searchModel,
        'dataProvider' => $dataProvider,
    ]);
}
```

Modifikasi Controller: Change Initial Sort & Perpage Gridview

Menampilkan 1-5 dari 21 item.

#	Title ↓ ¹ ₂	Genre	
	<input type="text"/>	<input type="text"/>	
1	Apocalypse Now	1979	  
2	Avengers: Endgame	2019	  
3	Braveheart	1995	  
4	Fargo	1996	  
5	Forrest Gump	1994	  

« 1 2 3 4 5 »

Modifikasi Model: Behavior

- Timestamp = otomatis menginput waktu insert dan update data. field: created_at dan updated_at (timestamp / int)
- Blameable = otomatis menginput user id yang menginsert dan mengupdate data. field: created_by dan updated_by (int)

```
public function behaviors()
{
    return [
        \yii\behaviors\TimestampBehavior::className(),
        \yii\behaviors\BlameableBehavior::className()
    ];
}
```

Form Validasi

Jenis Validasi Yang Disupport Yii

- ❖ boolean: yii\validators\BooleanValidator
- ❖ captcha: yii\captcha\CaptchaValidator
- ❖ compare: yii\validators\CompareValidator
- ❖ date: yii\validators\DateValidator
- ❖ datetime: yii\validators\DateValidator
- ❖ time: yii\validators\DateValidator
- ❖ default: yii\validators\DefaultValueValidator
- ❖ double: yii\validators\NumberValidator
- ❖ each: yii\validators\EachValidator
- ❖ email: yii\validators\EmailValidator
- ❖ exist: yii\validators\ExistValidator
- ❖ file: yii\validators\FileValidator
- ❖ filter: yii\validators\FilterValidator
- ❖ image: yii\validators\ImageValidator
- ❖ in: yii\validators\RangeValidator
- ❖ integer: yii\validators\NumberValidator
- ❖ match: yii\validators\RegularExpressionValidator
- ❖ required: yii\validators\RequiredValidator
- ❖ safe: yii\validators\SafeValidator
- ❖ string: yii\validators\StringValidator
- ❖ trim: yii\validators\FilterValidator
- ❖ unique: yii\validators\UniqueValidator
- ❖ url: yii\validators\UrlValidator
- ❖ ip: yii\validators\IpValidator

<https://www.yiiframework.com/doc/guide/2.0/en/tutorial-core-validators>

Validasi input melalui method rule di Model

```
public function rules()  
{  
    return [  
        // kode validasi disini  
    ];  
}
```

Validasi Required & Safe

- required: field tidak boleh kosong
[[`'title'`], `'required'`],
- safe: field tidak divalidasi
[[`'genre'`], `'safe'`],

Validasi String

- field harus teks

```
[[ 'title', 'string'],
```

- field maximal 50 char

```
[[ 'director', 'string', 'max' => 50],
```

- field min 5 char

```
[[ 'director', 'string', 'min' => 5],
```

- field min 5 char & max 50 char

```
[[ 'director', 'string', 'length' => [5,50]],
```

Validasi Integer & Double

```
// check field integer
```

```
[['quota'], 'integer'],
```

```
// check age = 17+
```

```
[['age'], 'integer', 'min' => 17],
```

```
// checks field = double (desimal)
```

```
[['salary', 'price'], 'double'],
```

Validasi Boolean & Unik

- Boolean: field harus diantara 1 atau 0 (true atau false)

```
[[ 'status', 'gender'], 'boolean'],
```

- Unique: field harus unik

```
[[ 'username'], 'unique'],
```


Validasi Compare

```
// validasi "password" sama dengan "password_repeat"  
['password', 'compare'],  
  
// sama dengan sebelumnya  
['password', 'compare', 'compareAttribute' => 'password_repeat'],  
  
// validasi umur harus 17+  
['age', 'compare', 'compareValue' => 17, 'operator' => '>=', 'type' => 'number'],
```

Validasi Date, Time, DateTime

```
[['birthday'], 'date'],  
[['start_film'], 'datetime'],  
[['work_start'], 'time'],
```

Validasi Email, URL & IP

- email: field harus berformat email abc@def.ghi
[['email_address'], 'email'],
- URL: field harus berformat URL valid http://example.com
[['homepage'], 'url'],
- ip: field harus berformat IP 127.0.0.1
[['ip_address'], 'ip'],

Validasi IN (Range)

```
// checks if "level" is 1, 2 or 3
```

```
['level', 'in', 'range' => [1, 2, 3]],
```

```
['user_level', 'in', 'range' => ['mimin', 'momod']],
```

Validasi File & Image

```
// checks if "primaryImage" is an uploaded image file in PNG, JPG or GIF format.  
// the file size must be less than 1MB  
['cover', 'file', 'extensions' => ['png', 'jpg', 'gif'], 'maxSize' => 1024*1024],  
  
// checks if "primaryImage" is a valid image with proper size  
['primaryImage', 'image', 'extensions' => 'png, jpg',  
  'minWidth' => 100, 'maxWidth' => 1000,  
  'minHeight' => 100, 'maxHeight' => 1000,  
],
```

Kustomisasi Validasi

```
// Kustomisasi pesan error
[["username"], "unique", "message"=>"Pilih username lain gan!",

// Kustomisasi validasi
[["nip"], function($attribute, $params){
    $length = strlen($this->$attribute);
    if(!in_array($length, [9, 18])){
        $this->addError($attribute, "NIP tidak valid");
        return false;
    }
}
```

<https://pavithratech.wordpress.com/2018/02/21/yii2-custom-validation/>

Upload File

Modifikasi View: Form

❖ fileInput

```
<?= $form->field($model, 'cover')->fileInput() ?>
```

❖ Pada form, tambahkan opsi enctype multipart/form-data

```
<?php $form = ActiveForm::begin([  
    'options' => ['enctype'=>'multipart/form-data']  
]); ?>
```


Modifikasi Controller: File Upload

```
<?php
use \yii\web\UploadedFile;
$file = UploadedFile::getInstance($model, "cover");
if($file){
    $file->saveAs("uploads/" . $file->baseName . "." . $file->extension);
}

// https://github.com/yiisoft/yii2/blob/master/docs/guide/input-file-upload.md
```

Modifikasi Controller: File Upload

```
$old_cover = $model->cover; // simpan old cover
if ($model->load(Yii::$app->request->post())) {
    $model->cover = $old_cover; // pertahankan old cover
    if($model->save()) {
        $file = \yii\web\UploadedFile::getInstance($model, "cover");
        if($file){
            $filename = "/movies/" . $file->baseName . "." . $file->extension;
            $file->saveAs(Yii::getAlias("@webroot").$filename);
            $model->cover = $filename;
            $model->save();
        }
        Yii::$app->session->setFlash('success', 'berhasil disimpan');
    }
    else{
        Yii::$app->session->setFlash('error', 'gagal disimpan');
    }
    return $this->redirect(['view', 'id' => $model->id]);
}
```

Contoh Menampilkan Image Pada DetailView

```
<?= DetailView::widget([
    'model' => $model,
    'attributes' => [
        // attribute lain
        [
            'attribute' => 'cover',
            'format' => 'html',
            'value' => function($data){
                return Html::img(\yii\helpers\Url::base(true).$data->cover, [
                    'width' => '200',
                    'class' => "img-thumbnail"
                ]);
            }
        ]
    ]
]);
```

Menggunakan Extension

Apa itu Extension?

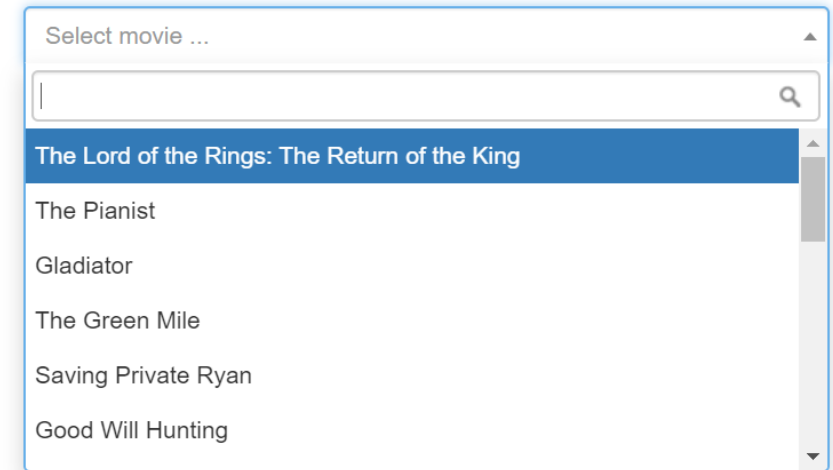
- Extension = Library tambahan diluar core Yii, bentuknya bisa widgets, helpers, module, maupun library PHP biasa
- Contoh: Yii2 Widget buatan Krajee

Yii2 Widget (Krajee Extension)

- Definisi: Kumpulan widget-widget penting untuk mempercantik user interface, yaitu: datepicker, timepicker, select2, depdrop, dll
- URL: <https://github.com/kartik-v/yii2-widgets>
- Instalasi (via CMD)
`composer require kartik-v/yii2-widgets "*"`

Yii2 Widget: Select2

```
<?php
use yii\helpers\ArrayHelper;
use app\models\Movie;
use kartik\widgets\Select2;
// ...
$movies = ArrayHelper::map(Movie::find()->all(),"id","title");
echo $form->field($model, 'movie_id')->widget(Select2::classname(), [
    'data' => $movies,
    'options' => ['placeholder' => 'Select a movie ...'],
    'pluginOptions' => [
        'allowClear' => true,
    ],
]);
?>
```



Yii2 Widget: SwitchInput

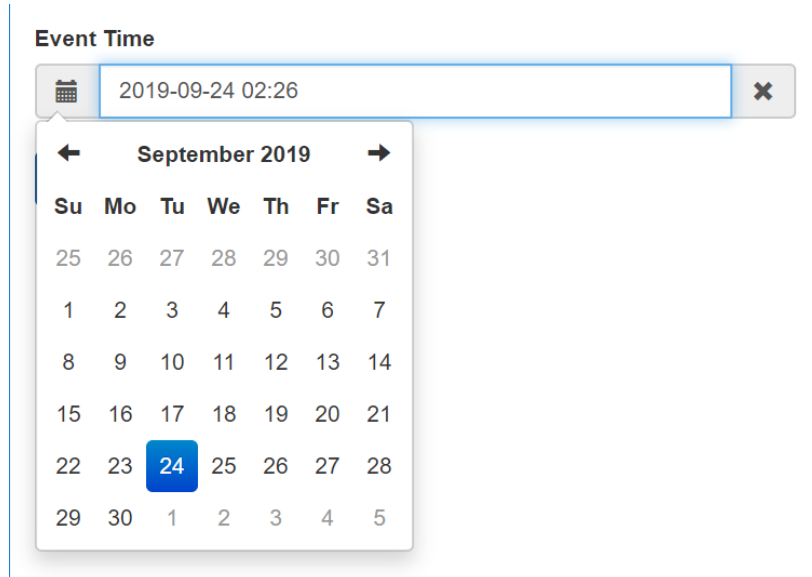
```
<?php
echo $form->field($model, 'gender')->widget(
    SwitchInput::classname(), [
        'type' => SwitchInput::CHECKBOX,
        'pluginOptions'=>[
            'handleWidth'=>60,
            'onText'=>'Pria',
            'offText'=>'Wanita'
        ]
    ]
);
?>
```

Jenis Kelamin

☒ Pria ☐ Wanita

Yii2 Widget: DateTimePicker

```
<?php
use \kartik\widgets\DatePicker;
// ...
echo $form->field($model, 'start')->widget(
    DateTimePicker::classname(), [
        'options' => [
            'placeholder' => 'Enter start time ...'
        ],
        'language' => 'id',
        'pluginOptions' => [
            'autoclose' => true,
            'format' => 'yyyy-mm-dd hh:ii:ss',
            'todayHighlight'=>true,
            'todayBtn'=>true,
        ]
    ]
);
?>
```



Setting Default Time Zone

```
// config/web.php
$config = [
    'id' => 'basic',
    'basePath' => dirname(__DIR__),
    'bootstrap' => ['log'],
    'language' => 'id',
    'timeZone' => 'Asia/Jakarta',
```

Setting Default Value Pada Form

```
// controllers/ScheduleCotroller.php

public function actionCreate()
{
    $model = new Schedule();
    // default value
    $model->start = date('Y-m-d H:i:s');
    $model->quota = 30;
    $model->price = 75000;
    $model->status = true;
}
```

Terima Kasih