# YII FRAMEWORK 2.0

The Fast, Secure & Professional Framework

# #2

# MVC YII & DATABASE

**Let's start it!**

# Outline

- Model View Controller
  - konsep MVC
  - MVC di Yii
  - alur kerja aplikasi
- Bekerja dengan Database
  - konfigurasi database
  - active record dasar
  - CRUD sederhana

# Model View Controller

Yii mengimplementasikan konsep *Model-View-Controller* (MVC) *design pattern*

# Konsep MVC

*Model-View-Controller design pattern* merupakan konsep populer yang umum digunakan oleh *framework* pemrograman modern.

MVC adalah sebuah pola arsitektur perangkat lunak yang membagi aplikasi menjadi tiga bagian yang saling berhubungan.

MVC memisahkan antara data atau informasi, bagaimana data itu ditampilkan, dan aturan bisnis terkait data tersebut.

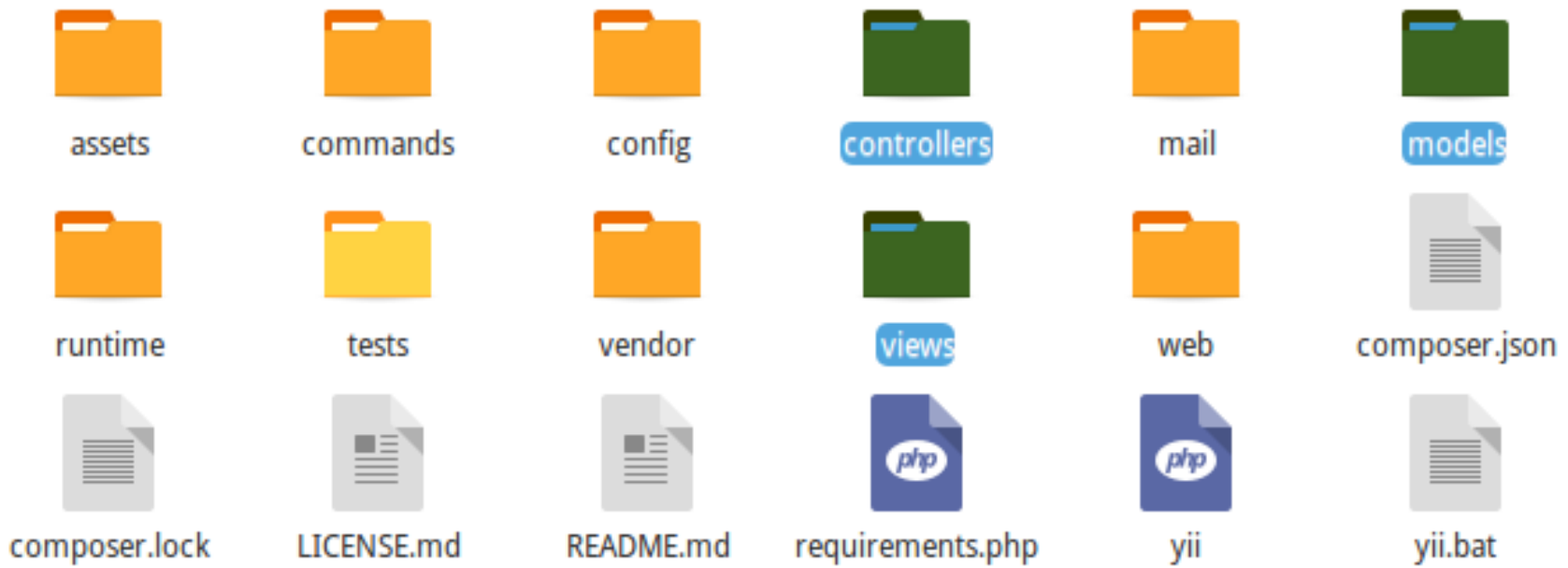
- *Model*, format data yang digunakan
- *View*, tampilan data
- *Controller*, bagaimana data ditampilkan

# Keuntungan Penerapan MVC

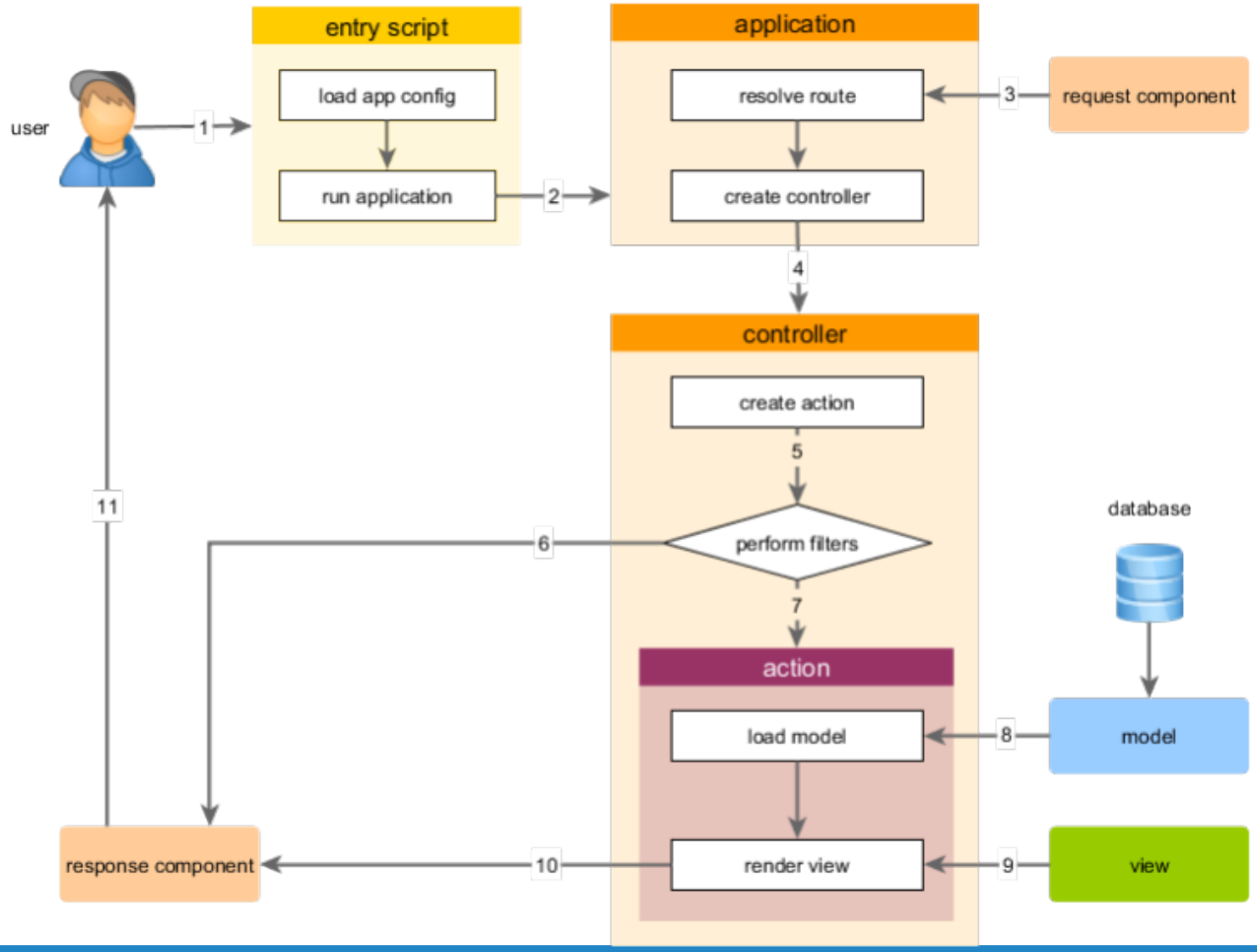
- Struktur aplikasi akan lebih rapi dan mudah dipahami terutama untuk proyek aplikasi yang kompleks
- Lebih mudah *maintan* ketika terjadi perubahan data, bisnis proses maupun tampilan
- Lebih mudah dalam *tracking* dan *handling error*
- Mudah dalam membagi pekerjaan jika proyek aplikasi dikerjakan oleh tim

# MVC di Yii

MVC di Yii juga tercermin pada struktur aplikasinya, yaitu terdapat *folder models* (berisi semua *file model*), *views* (berisi semua *file view*), dan *controllers* (berisi semua *file controller*).



# Alur Kerja Aplikasi



# Membuat *Hello Word*

Setelah belajar tentang konsep MVC, kini saatnya kita mengimplementasikannya di Yii.

Untuk menampilkan teks "*hello word*" di Yii caranya cukup mudah. Caranya tambahkan fungsi bernama actionHello() pada SiteController, contohnya sebagai berikut.

@app/controllers/SiteController.php

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

# Konsep Dasar Routing

*Routing* adalah pemetaan antara URL dengan konten.

Konsep *routing* di Yii sangat-sangat sederhana, karena sifatnya otomatis berdasarkan nama fungsi *action* di *controller* dan tidak perlu kita definisikan secara manual.

# Aturan Dasar Routing

- *Routing* di Yii sifatnya *case sensitive* atau membedakan huruf besar dan huruf kecil.
- *Routing* di Yii otomatis terbentuk sesuai dengan *controller* dan fungsi *action*-nya. Antara *controller* dan fungsi *action* dipisahkan dengan tanda slash /.
- Nama fungsi pada *controller* yang akan menjadi *routing* adalah yang diawali dengan kata *action*, contoh : actionIndex
- Jika nama *controller* lebih dari satu kata, misal BelajarYiiController, maka *routing* menggunakan *separator dash* ( - ) untuk memisahkan dua kata tersebut.

```
~ index.php?r=belajar-yii
```

# Aturan Dasar Routing (cont)

- Demikian juga, jika nama *action* lebih dari satu kata actionJumlahRoda, maka *routing* juga akan menggunakan *separator dash* ( - ) untuk memisahkan dua kata tersebut

```
~ index.php?r=mobil/jumlah-roda
```

- Penulisan nama *controller* harus menggunakan format *StudlyCaps*
- Penulisan nama fungsi *action* menggunakan format *camelCase*

# Membuat *Hello Word* Tingkat Lanjut

Kata "Hello Word" yang ditampilkan pada bagian sebelumnya hanyalah teks polos, artinya *layout* utama Yii tidak ikut tampil. Tidak cukup dengan menggunakan *controller* saja melainkan menggunakan *view* juga.

@app/controllers/SiteController.php

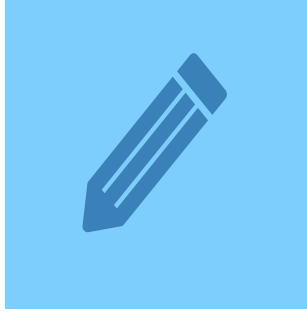
```
public function actionTampil()
{
      return $this->render('hello');
}
```

# *Link* Antar Halaman

Belajar tentang bagaimana menghubungkan *link* antar halaman yang telah kita buat.

Namun sebelumnya kita harus mengetahui terlebih dahulu tentang bagaimana membuat URL di Yii.

# Membuat URL



Yii membuat *class* [[yii\helpers\Url]] untuk menangani pembuatan URL.

@app/views/site/hello.php

```
<?php
use \yii\helpers\Url;

echo Url::home();
echo "<br>";
echo Url::to();
echo "<br>";
echo Url::to(['create']);
echo "<br>";
echo Url::to(['person/index']);
echo "<br>";
echo Url::to(['person/index', 'nama'=>'Edo']);
```

# Membuat *Hyperlink*

*Hyperlink* merupakan tautan atau *link* yang terdapat pada halaman *website*, dengan tujuan untuk mengarahkan pengunjung web ke suatu target.

@app/views/site/hello.php

```
<?php
use \yii\helpers\Url;
?>
<a href="<?= Url::to(['person/index', 'nama'=>'Edo']) ?>">Data Person</a>
```

Atau menggunakan fungsi-fungsi HTML *helpers* melalui class [[yii\helpers\Html]]

```
<?php
use \yii\helpers\Html;
?>
echo Html::a('Example', 'http://www.example.com');
echo Html::a('Data Person', ['person/index']);
```

# Penerapan MVC pada *Form*

Berikut ini kita membuat tampilan *form* komentar dengan menggunakan Yii

@app/models/Komentar.php

```
<?php
namespace app\models;
class Komentar extends \yii\base\Model
{
    public $nama;
    public $pesan;

    public function rules()
    {
        return [
            [['nama', 'pesan'], 'required', 'message'=>'{attribute} tidak
boleh kosong.'],
        ];
    }
}
```

# Penerapan MVC pada *Form* (cont)

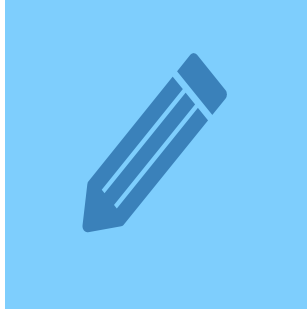
Selanjutnya membuat fungsi actionKomentar pada *controller* (misal : SiteController)

@app/controllers/SiteController.php

```
public function actionKomentar()
{
    $model = new \app\models\Komentar();

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

# Penerapan MVC pada *Form* (cont)



Buat tampilan *form*-nya pada

@app/views/site/komentar.php

```
<?php
use yii\helpers\Html;
use yii\bootstrap\ActiveForm;
?>
<h1>Komentar</h1>
<?php $form = ActiveForm::begin(['layout' => 'horizontal']); ?>
<?= $form->field($model, 'nama')->label('Nama Anda') ?>
<?= $form->field($model, 'pesan')->label('Pesan Anda') ?>
<?= Html::submitButton('Simpan', ['class'=>'btn btn-primary']) ?>
<?php ActiveForm::end(); ?>
```

Uji coba melalui browser!

# Penerapan MVC pada *Form* (cont)

Melakukan pemrosesan atas *input* data dari user. Tambahan pada actionKomentar

@app/controllers/SiteController.php

```
public function actionKomentar()
{
    $model = new \app\models\Komentar();
    // tambahan
    if(Yii::$app->request->post()){
        $model->load(Yii::$app->request->post());
        if($model->validate()){
            Yii::$app->session->setFlash('success', 'Terima kasih');
        }else{
            Yii::$app->session->setFlash('error', 'Maaf, salah!');
        }
        return $this->render('hasil_komentar', [
            'model'=>$model,
        ]);
    }else{
        return $this->render('komentar', [
            'model'=>$model,
        ]);
    }
}
```

# Penerapan MVC pada *Form* (cont)

## Buat view hasil komentar

@app/views/site/hasil_komentar.php

```
<?php
if(Yii::$app->session->hasFlash('success')){
    echo '<div class="alert alert-success">';
    echo Yii::$app->session->getFlash('success');
    echo '</div>';
    echo '<br>Nama : '.$model->nama;
    echo '<br>Pesan : '.$model->pesan;
}else if (Yii::$app->session->hasFlash('error')){
    echo '<div class="alert alert-danger">';
    echo Yii::$app->session->getFlash('error');
    echo '</div>';
}
```

Uji coba melalui browser!

# Membuat *Widget* Sederhana

*Widget* adalah blok atau bagian dari tampilan yang dapat digunakan kembali.

*Widget* digunakan pada *view* untuk membuat elemen antar muka yang kompleks namun bisa dikonfigurasi dalam gaya pendekatan berbasis *object*.

Konsep DRY atau *Don't Repeat Your Self*.

# *Widget Alert*

Kode *file* Widget Alert, sebagai berikut

@app/widgets/Alert.php

```
<?php
namespace app\widgets;
use Yii;
class Alert extends \yii\bootstrap\Widget
{
   public function init()
   {
      parent::init();
      if(Yii::$app->session->hasFlash('success')){
         echo '<div class="alert alert-success">';
         echo Yii::$app->session->getFlash('success');
         echo '</div>';
      }else if (Yii::$app->session->hasFlash('error')){
         echo '<div class="alert alert-danger">';
         echo Yii::$app->session->getFlash('error');
         echo '</div>';
      }
   }
}
```

# Penggunaan *Widget Alert*

## Implementasikan pada view hasil komentar

@app/views/site/hasil_komentar.php

```
<?php
use app\widgets\Alert;
echo Alert::widget();

if(Yii::$app->session->hasFlash('success')){
   echo '<br>Nama : '.$model->nama;
   echo '<br>Pesan : '.$model->pesan;
}
```

# Penggunaan *Widget Alert* (cont)

Agar berlaku global, bisa diimplementasikan pada layout utama

@app/views/layouts/main.php

```
<div class="container">
    <?= Breadcrumbs::widget([
        'links' => isset($this->params['breadcrumbs']) ? $this-
>params['breadcrumbs'] : [],
    ]) ?>
    <!-- TAMBAHKAN WIDGET ALERT -->
    <?= \app\widgets\Alert::widget(); ?>
    <?= $content ?>
</div>
```

@app/views/site/hasil_komentar.php

```
<?php
if(Yii::$app->session->hasFlash('success')){
    echo '<br>Nama : '.$model->nama;
    echo '<br>Pesan : '.$model->pesan;
}
```

# Pengaturan *Pretty* URL

*by default* URL pada Yii memiliki format sebagai berikut

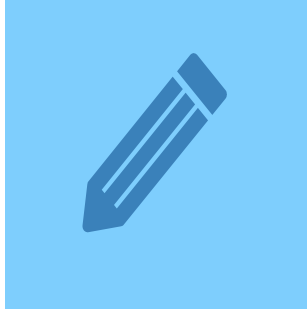
```
~ index.php?r=controllerID/actionID
```

Untuk mempercantik URL address caranya adalah sebagai berikut

@app/config/web.php

```
'components' => [
    ...
    'urlManager' => [
        'enablePrettyUrl' => true,
    ]
    ...
],
```

# Pengaturan *Pretty* URL (cont)



## Menghilangkan index.php dari URL

@app/config/web.php

```
'components' => [
    ...
    'urlManager' => [
        'enablePrettyUrl' => true,
        'showScriptName' => false,
    ]
    ...
],
```

@app/web/.htaccess

```
RewriteEngine on

# If a directory or a file exists, use it directly
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
# Otherwise forward it to index.php
RewriteRule . index.php
```

# Bekerja dengan Database

*Database* merupakan bagian penting pada sebuah aplikasi yang berfungsi untuk menyimpan data

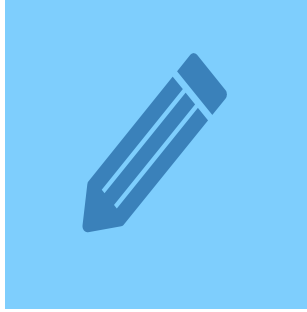
# Persiapan *Database*

*Framework* Yii secara *default* mendukung banyak *database* populer seperti MySQL, PostgreSQL, Ms SQL Server, Oracle dll.

Yii menggunakan teknologi PDO PHP *extension* dalam hal koneksi *database*nya.

Sebagai latihan, kita akan menggunakan MySQL membuat database bernama "yii2basic", kemudian membuat tabel "employee".

# Persiapan *Database* (cont)



Perintah *query* nya sebagai berikut

```
CREATE database yiibasic;

use yiibasic;

CREATE TABLE employee (
id int primary key auto_increment,
name varchar(50) NOT NULL,
age int(3) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

INSERT INTO employee VALUES (default, 'Edo', 25);
INSERT INTO employee VALUES (default, 'Arief', 28);
INSERT INTO employee VALUES (default, 'Dewi', 32);
INSERT INTO employee VALUES (default, 'Wahyu', 30);
```

# Konfigurasi Database Pada Yii

## Konfigurasi *Global*

@app/config/db.php

```
<?php
return [
    'class' => 'yii\db\Connection',
    'dsn' => 'mysql:host=localhost;dbname=yii2basic',
    'username' => 'root',
    'password' => '',
    'charset' => 'utf8',
];
```

## Konfigurasi Lokal

```
$db = new \yii\db\Connection([
   'dsn' => 'mysql:host=localhost;dbname=yii2basic',
   'username' => 'root',
   'password' => '',
   'charset' => 'utf8',
]);
```

# Uji Coba *Query*

Untuk menguji apakah koneksi *database* kita benar-benar berjalan dengan baik adalah dengan mencoba beberapa *query* untuk mengakses *database*.

## *Query Select*

@app/controllers/SiteController.php

```
public function actionQuery()
{
    $db = Yii::$app->db;
    $command = $db->createCommand('SELECT * FROM employee');
    $employees = $command->queryAll();
    // Ekstrak data
    foreach($employees as $employee){
        echo "<br>";
        echo $employee['id']." ";
        echo $employee['name']." ";
        echo "(".$employee['age'].") ";
    }
}
```

# Uji Coba *Query* (cont)

## Contoh lain

@app/controllers/SiteController.php

```
public function actionQuery2()
{
    $db = Yii::$app->db;
    // return a single row
    $employee = $db->createCommand('SELECT * FROM employee where id=1')->queryOne();
    echo $employee['id']." ";
    echo $employee['name']." ";
    echo "(".$employee['age'].") ";
    echo "<hr>";

    // return a single column (the first column)
    $names = $db->createCommand('SELECT name FROM employee')->queryColumn();
    print_r($names);
    echo "<hr>";

    // return a scalar
    $count = $db->createCommand('SELECT COUNT(*) FROM employee')->queryScalar();
    echo "Jumlah Employee ".$count;
    echo "<hr>";
}
```

# Active Record Dasar

Active Record adalah sebuah class antar muka yang digunakan untuk mengakses dan memanipulasi data yang disimpan pada *database*.

Konsep Active Record di Yii sama dengan konsep *Active Record Pattern* pada umumnya (*Object Relational Model/ORM).*

Class Active Record pada Yii merupakan *model* pada konsep MVC.

# *Model Active Record*

Berikut ini contoh deklarasi minimal dari *model Active Record* dari employee

@app/models/Employee.php

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

class Employee extends ActiveRecord
{
   public static function tableName()
   {
      return 'employee';
   }
}
```

# Implementasi *Active Record*

Berikut ini contoh Class Active Record untuk *query select* data

@app/controllers/SiteController.php

```
public function actionActiveRecord()
{
   $employees = \app\models\Employee::find()->all();
   foreach($employees as $employee){
      echo "<br>";
      echo $employee['id']." ";
      echo $employee['name']." ";
      echo "(".$employee['age'].") ";
   }
}
```