



# Web Programming



Hafid Mukhlisin



# Agenda

---

1. Konsep OOP di PHP
2. Briefing Proyek
3. Persiapan Lingkungan Kerja
4. Pengenalan Yii Framework
5. Konsep MVC di Yii
6. Bekerja Dengan Database
7. Otentikasi
8. Gii: Code Generator
9. Modifikasi CRUD
10. Bekerja dengan Extension
11. Impor & Ekspor Data
12. Visualisasi Data
13. Otorisasi
14. Finishing Proyek



**40 Jam  
5 hari**



# Konsep OOP di PHP

---

- ❖ Introduction
- ❖ Class, Property, Method
- ❖ Setter vs Getter
- ❖ Constructor vs Destructor
- ❖ Inheritance / Pewarisan
- ❖ Polymorphisme / Method Overriding
- ❖ Ruang Lingkup Property & Method
- ❖ Static, Self & Parent
- ❖ Method Chaining
- ❖ Visibility / Ruang Lingkup
- ❖ Abstract vs Interface
- ❖ Namespace
- ❖ Autoloading



# Konsep OOP di PHP

---

## Introduction

- Object Oriented Programming
- Object = Segala sesuatu yang bisa diciptakan dan dimusnahkan
- Object vs Prosedural



# Konsep OOP di PHP

---

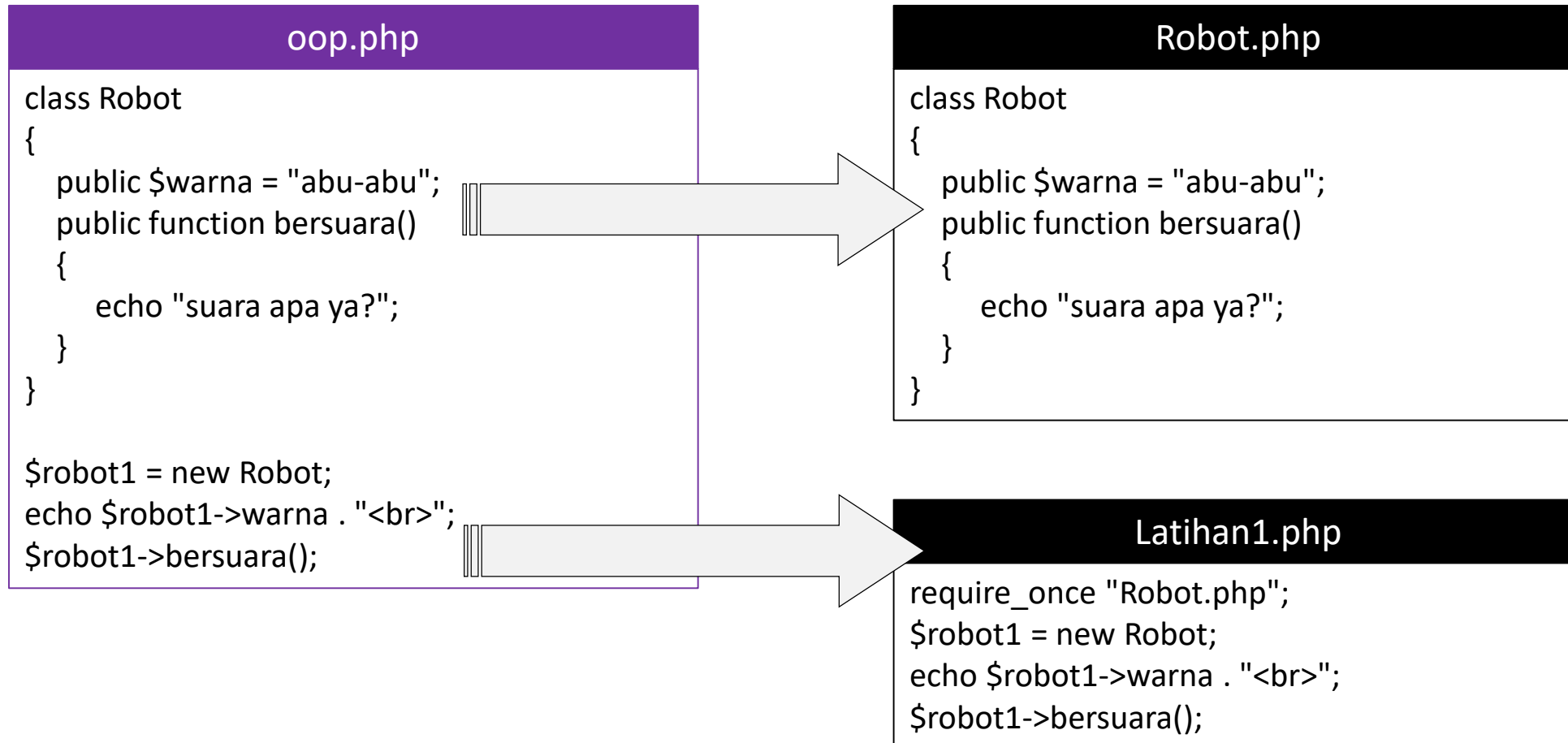
## Class, Property, Method

- Class = Blueprint / Design dari Object
- Property = Atribut dari Class (Variabel & Constant)
- Method = Kemampuan/Fungsi dari Class
- Object = Instance dari Class



# Konsep OOP di PHP

## Class, Property, Method (Praktik)





# Konsep OOP di PHP

---

## PHP Standard Recommendation (PSR)

- Format penulisan nama class **HARUS** adalah StudlyCaps
- Nama file class sebaiknya sama dengan nama classnya
- Format penulisan nama method **HARUS** camelCase
- Format penulisan nama variabel bebas tapi konsisten
- Kode PHP **HARUS** berada diantara `<?php ?>` dan atau `<?= ?>`
- Jika satu file hanya berisi kode PHP maka **JANGAN** ditutup `?>`



## Referensi

- <http://www.php-fig.org/psr/psr-1/>
- <http://www.php-fig.org/psr/psr-2/>



# Konsep OOP di PHP

---

## Setter vs Getter

- Setter = Method untuk mendefinisikan nilai suatu property
- Getter = Method untuk mendapatkan nilai suatu property





# Konsep OOP di PHP

## Setter vs Getter (Praktik)

### Robot.php

```
class Robot
{
    public $warna;
    public function setWarna($warna)
    {
        $this->warna = $warna;
    }
    public function getWarna()
    {
        return $this->warna;
    }
}
```

### Latihan2.php

```
require_once "Robot.php";
$robot1 = new Robot;
$robot->setWarna("MERAH");
echo $robot1->getWarna() . "<br>";
```



# Konsep OOP di PHP

---

## Constructor vs Destructor

- Constructor = Method yang pertama kali jalan ketika object dibuat
- Destructor = Method yang pertama kali jalan ketika object dihapus
- Keyword = `__construct`, `__destruct`



# Konsep OOP di PHP

## Constructor vs Destructor (Praktik)

### Robot.php

```
class Robot
{
    public $warna;
    public function __construct($warna)
    {
        $this->warna = $warna;
    }
    public function getWarna()
    {
        return $this->warna;
    }
}
```

### Latihan3.php

```
require_once "Robot.php";
$robot1 = new Robot("HITAM");
echo $robot1->getWarna() . "<br>";
```



# Konsep OOP di PHP

---

## Inheritance / Pewarisan

- Inheritance = class child memiliki sifat yang sama dengan parent
- Keyword = extends



# Konsep OOP di PHP

## Inheritance / Pewarisan (Praktik)

### Robot.php

```
class Robot
{
    public $warna;
    public function __construct($warna)
    {
        $this->warna = $warna;
    }
    public function getWarna()
    {
        return $this->warna;
    }
}
```

### Burung.php

```
require_once "Robot.php";
class Burung extends Robot
{
    public function terbang()
    {
        echo "Kepakkan sayap";
    }
}
```

### Latihan4.php

```
require_once "Burung.php";
$robot_burung1 = new Burung("HITAM");
$robot_burung1->terbang();
// menggunakan method & property parent
echo $robot_burung1->warna . "<br>";
echo $robot_burung1->getWarna() . "<br>";
```



# Konsep OOP di PHP

---

## Polymorphisme / Method Overriding

- Polymorphisme = Method yang sama pada class child bisa memiliki implementasi yang berbeda dengan method pada class parent



# Konsep OOP di PHP

## Polymorphisme / Method Overriding (Praktik)

### Robot.php

```
class Robot
{
    public $warna;
    public function __construct($warna)
    {
        $this->warna = $warna;
    }
    public function getWarna()
    {
        return $this->warna;
    }
}
```

### Burung.php

```
require_once "Robot.php";
class Burung extends Robot
{
    public function terbang()
    {
        echo "Kepakkan sayap";
    }
    public function getWarna()
    {
        echo "Warnanya: " . $this->warna;
    }
}
```

### Latihan5.php

```
require_once "Burung.php";
$robot_burung1 = new Burung("HITAM");
$robot_burung1->getWarna() . "<br>";
```



# Konsep OOP di PHP

---

## Static, Self & Parent

- Static = cara memanggil property atau method tanpa membuat object
  - Self = Mengakses property dan method dari class tersebut
  - Parent = Mengakses property dan method dari class bapaknya





# Konsep OOP di PHP

## Static, Self & Parent (Praktik)

### Robot.php

```
class Robot
{
    public static $warna;
    public function __construct($warna)
    {
        $this->warna = $warna;
    }
    public static function getWarna()
    {
        return self::$warna;
    }
}
```

### Burung.php

```
require_once "Robot.php";
class Burung extends Robot
{
    public function terbang()
    {
        echo "Kepakkan sayap";
    }
    public function getWarna()
    {
        echo "Warnanya: " . self::$warna;
    }
    public static function tampilWarna()
    {
        echo parent::getWarna() . "<br>";
        echo self::getWarna() . "<br>";
    }
}
```

### Latihan6.php

```
require_once "Burung.php";

Burung::$warna = "HIJAU";
Burung::tampilWarna();

// HIJAU
// Warna HIJAU
```



# Konsep OOP di PHP

---

## Method Chaining

- Method chaining = cara memanggil lebih dari satu method dalam satu perintah
- Contoh:  
`$nama_object->method_1()->method_2()->method_n();`



# Konsep OOP di PHP

## Method Chaining (Praktik)

### Komputer.php

```
class Komputer
{
    public function beli()
    {
        echo "1. Beli hardware komputer <br>";
        return $this;
    }
    public function rakit()
    {
        echo "2. Rakit komputer <br>";
        return $this;
    }
    public function install()
    {
        echo "3. Install OS <br>";
        return $this;
    }
}
```

### Latihan7.php

```
require_once "Komputer.php";

$pc = new Komputer;

$pc->beli();
$pc->rakit();
$pc->install();

$pc->beli()->rakit()->install();
```



# Konsep OOP di PHP

---

## Visibility / Ruang Lingkup

- Visibility = keteraksesan method dan property pada suatu class dan childnya
  - Public = method & property bisa diakses secara global baik dari dalam class, object, maupun dari class childnya
  - Private = method & property HANYA bisa diakses dari class tersebut
  - Protected = method & property bisa diakses dari class tersebut dan child-nya, TIDAK BISA diakses dari object



# Konsep OOP di PHP

## Visibility / Ruang Lingkup (Praktik)

### User.php

```
class User
{
    protected $email = "admin@gmail.com";
    private $password = "123456";

    public function getEmail()
    {
        return $this->email;
    }

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

### Latihan8.php

```
require_once "User.php";

$user1 = new User;
//echo $user1->password . "<br>";
//echo $user1->email . "<br>";
echo $user1->getEmail() . "<br>";
echo $user1->getPassword() . "<br>";
```



# Konsep OOP di PHP

---

## Visibility / Ruang Lingkup (Praktik)

### Employee.php

```
require_once "User.php";

class Employee extends User
{
    public function test()
    {
        echo parent::$password . "<br>";
        echo $this->email . "<br>";
    }
}
```

### Latihan9.php

```
require_once "Employee.php";

$employee1 = new Employee;
echo $employee1->test();
```



# Konsep OOP di PHP

---

## Abstract vs Interface

- Abstract = class / method yang methodnya parent diwariskan dan harus dideklarasikan di class child (ada method yang bukan abstract)
- Interface = class yang semua methodnya parent harus dideklarasikan atau diimplementasikan di class child



# Konsep OOP di PHP

## Abstract vs Interface (Praktik)

### Hewan.php

```
abstract class Hewan
{
    abstract protected function bersuara();
    public function berjalan()
    {
        echo "berjalan dg kaki";
    }
}
```

### Kucing.php

```
require_once "Hewan.php";
class Kucing extends Hewan
{
    public function bersuara()
    {
        return "meooong";
    }
}
```

### Latihan10.php

```
require_once "Kucing.php";

$ kucing1 = new Kucing;
echo $ kucing1->bersuara();
```





# Konsep OOP di PHP

## Abstract vs Interface (Praktik)

### Kemampuan.php

```
interface Kemampuan
{
    public function melompat();
}
```

### Latihan11.php

```
require_once "Kuda.php";

$kuda1 = new Kuda;
echo $kuda1->melompat();
```

### Kuda.php

```
require_once "Hewan.php";
require_once "Kemampuan.php";
class Kuda extends Hewan implements Kemampuan
{
    public function bersuara()
    {
        return "kikikikik";
    }

    public function melompat()
    {
        return "lompat tinggi";
    }
}
```



# Konsep OOP di PHP

## Namespace

- Namespace = keyword untuk memberi ruang khusus pada memori yang mengisolasi penamaan dari luar

### Folder directory

```
+ app
- librari
- librari2
```

### Latihan12.php

```
namespace app;
require "librari/Mobil.php";
require "librari2/Mobil.php";

use app\librari\Mobil;
use app\librari2\Mobil as Mobil2;

$mobil1 = new Mobil;
$mobil2 = new Mobil2;
```

### Mobil.php

```
namespace app\librari;
class Mobil
{

}
```

### Mobil.php

```
namespace app\librari2;
class Mobil
{

}
```



# Konsep OOP di PHP

---

## Autoloading

- Autoloading = Include class otomatis ketika class diciptakan

### Latihan13.php

```
spl_autoload_register(function ($class_name) {  
    require $class_name . ".php";  
    // require str_replace('\\', '/', $ class_name) . ".php";  
});  
  
$robot1 = new Robot("MERAH");  
echo "Warna : " . $robot1->getWarna();
```



# Briefing Proyek

---

## Studi Kasus: (Sistem Informasi Kepegawaian) SIMPEG

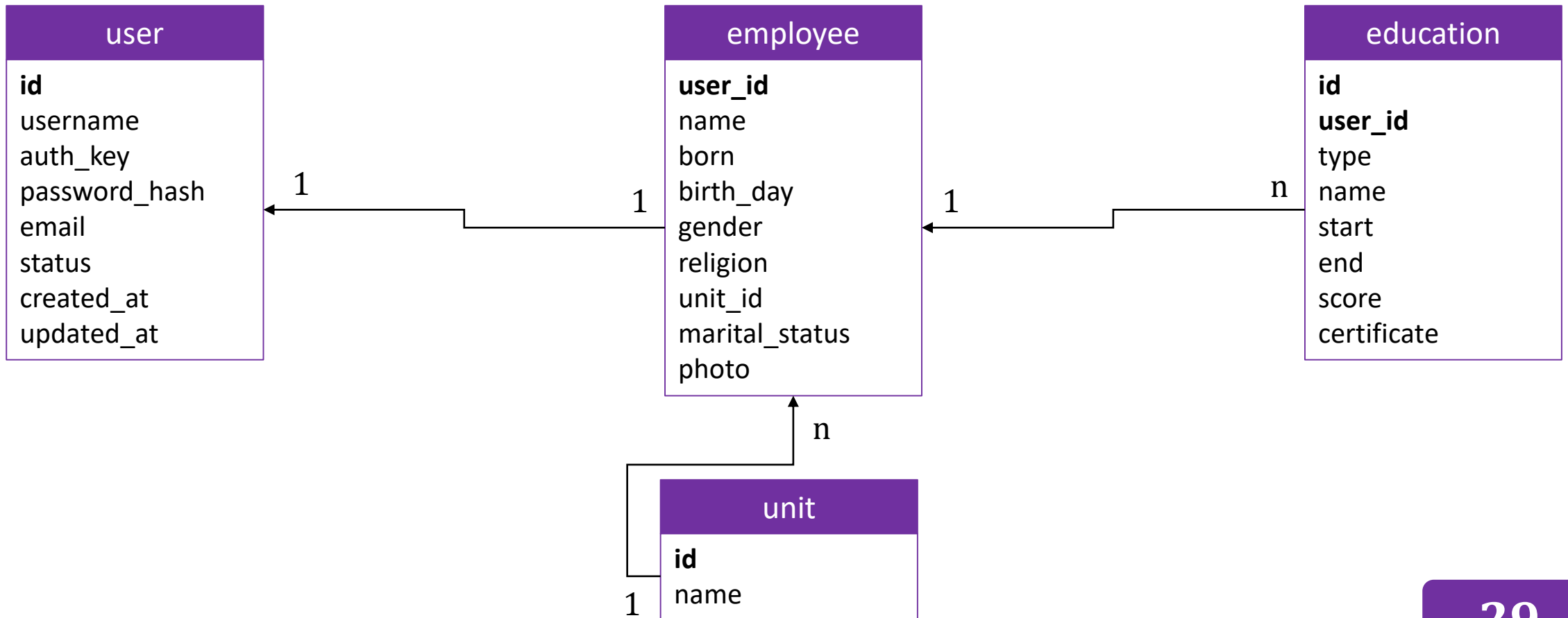
- Fitur:

Pegawai	Admin
Login	Login
Update Profile Pegawai	Kelola Data Pegawai
Kelola Data Riwayat Pendidikan	Impor & Ekspor Data Pegawai
Pencarian Data Pegawai Lain	
Cetak Profile Pegawai	
Upload & Download Ijazah	
Upload Foto	



# Briefing Proyek

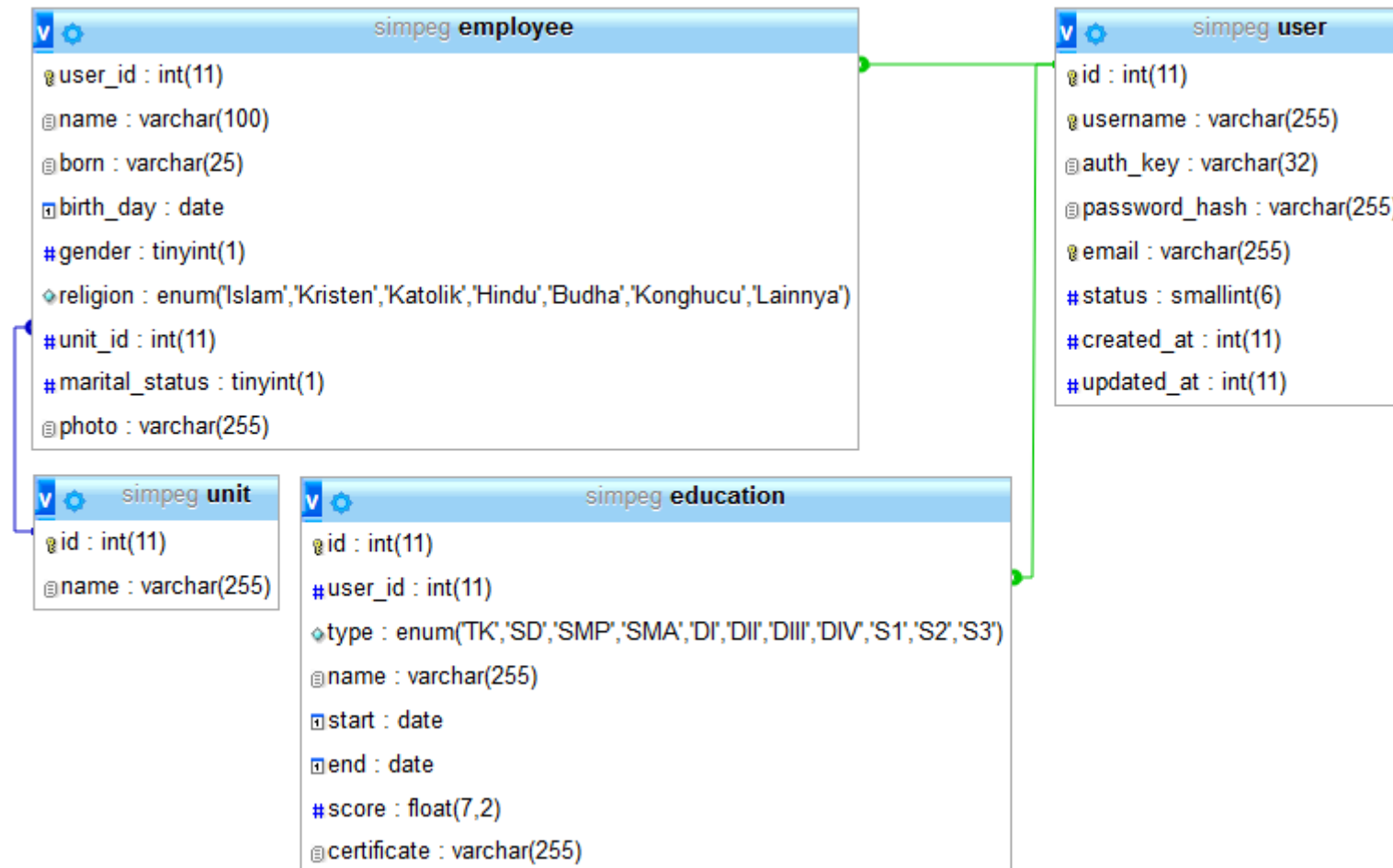
## Struktur Database SIMPEG





# Briefing Proyek

## Struktur Database SIMPEG





# Persiapan Lingkungan Kerja

---

## Checklist

- PHP in CMD
- Composer
- Github



# Persiapan Lingkungan Kerja

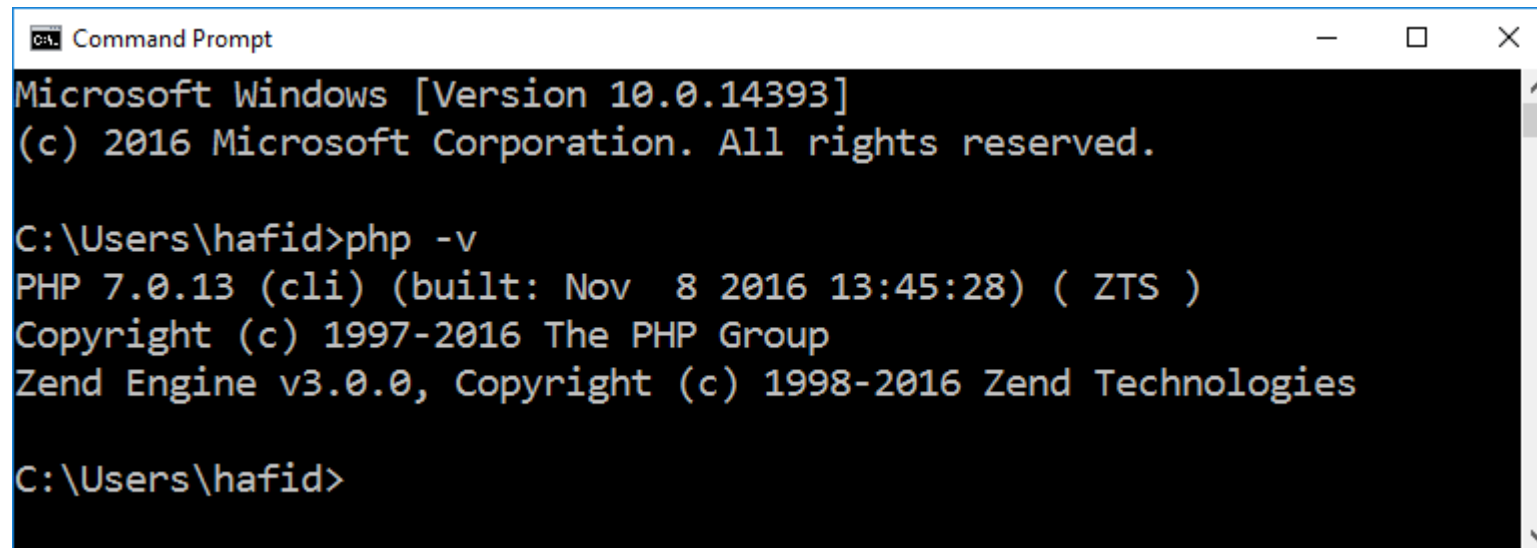
---

## PHP in CMD

- Setting PHP sebagai variabel PATH agar bisa dijalankan via CMD hanya dengan perintah "php"

`SETX /M PATH "%PATH%;C:\xampp\php;"`

`php -v`



```
Command Prompt
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\hafid>php -v
PHP 7.0.13 (cli) (built: Nov  8 2016 13:45:28) ( ZTS )
Copyright (c) 1997-2016 The PHP Group
Zend Engine v3.0.0, Copyright (c) 1998-2016 Zend Technologies

C:\Users\hafid>
```





# Persiapan Lingkungan Kerja

## Composer

- **Definisi:** Dependecy manager untuk PHP, berfungsi sebagai installer online yang mirip apt-get di ubuntu.
- **Cara Kerja:** Composer akan mendownload file2 yang dibutuhkan dari server repository. Repository utama composer adalah packagist.org.
- **Instalasi:**

```
php -r "readfile('https://getcomposer.org/installer');" | php
```

- **Tips trik**

- Pastikan modul open-ssl aktif di php.ini
- Jika koneksinya via proxy, maka perlu di set via CMD

```
set http_proxy=http://username:password@host:port
set https_proxy=https://username:password@host:port
```





# Persiapan Lingkungan Kerja

## Instalasi Composer

```
Command Prompt
C:\Users\hafid>php -r "readfile('https://getcomposer.org/installer');" | php
All settings correct for using Composer
Downloading...

Composer (version 1.3.2) successfully installed to: C:\Users\hafid\AppData\Local\Composer\composer.phar
Use it: php composer.phar

C:\Users\hafid>
```

```
Command Prompt
C:\Users\hafid>composer

Composer version 1.3.1 2017-01-07 18:08:51

Usage:
  command [options] [arguments]
```

### Tips

echo @php "%~dp0composer.phar" %\*>composer.bat  
Daftarkan lokasi composer sebagai variabel PATH



# Persiapan Lingkungan Kerja

---

## Github

- **Definisi:**

- Git merupakan tools yang berfungsi untuk mengontrol revisi secara terdistribusi dan manajemen source code (distributed revision control and source code management atau SCM).
- Sedangkan Github (<https://github.com/>) merupakan layanan berbasis web hosting untuk proyek-proyek pengembangan perangkat lunak yang menggunakan sistem kontrol revisi git.

- **Mengapa Github?:**

Kode-kode sumber librari PHP umumnya dihosting di github dan untuk mendownloadnya, composer perlu berkomunikasi dengan API github

- **Setting OAuth Github (opsional):**

- Buka laman setting application <https://github.com/settings/applications>
- Generate OAuth token (personal access token) <https://github.com/settings/tokens/new>
- `composer config -g github-oauth.github.com <oauthtoken>`





# Pengenalan Yii Framework

---

## Topic

- Selayang Pandang
- Instalasi Yii
- Testing Hasil Instalasi Yii
- Arsitektur Yii



# Pengenalan Yii Framework

---

## Selayang Pandang

- Yii adalah framework PHP, kumpulan kode PHP sebagai dasar untuk membangun berbagai jenis aplikasi berbasis web, misalnya: web portal, forum, CMS, e-commerce, web service, dll
- Fitur: MVC, support banyak database, ORM, caching, RESTful, code generator, security, application template, dsb
- Official Site: <http://www.yiiframework.com>
- Sejarah:
  - Mulai dikembangkan 2008, oleh Qiang Xue (sebelumnya dev framework Prado)
  - Versi 1.x 2008 – 2014 disupport hingga akhir 2018
  - Versi 2.x oktober 2014 – sekarang



# Pengenalan Yii Framework

---

## Application Template

Feature	Basic	Advanced
Project structure	✓	✓
Site controller	✓	✓
User login/logout	✓	✓
Forms	✓	✓
DB connection	✓	✓
Console command	✓	✓
Asset bundle	✓	✓
Codeception tests	✓	✓
Twitter Bootstrap	✓	✓
Front- and back-end apps		✓
Ready to use User model		✓
User signup and password restore		✓



# Pengenalan Yii Framework

## Instalasi Yii

- Perintah Instalasi
  - `composer global require "fxp/composer-asset-plugin:^1.2.0"`
  - `composer create-project --prefer-dist yiisoft/yii2-app-basic nama_folder`

```
Command Prompt
C:\Users\hafid>composer global require "fxp/composer-asset-plugin:^1.2.0"
Changed current directory to C:/Users/hafid/AppData/Roaming/Composer
```

```
Command Prompt - composer create-project --prefer-dist yiisoft/yii2-app-basic simpeg
C:\Users\hafid>cd c:\xampp\htdocs
c:\xampp\htdocs>composer create-project --prefer-dist yiisoft/yii2-app-basic simpeg
1/10: http://263d18f2bf64db02c0329a3
2/10: http://b1914b94f529c00c53a14ee
3/10: http://0ef5d7c1c0ae6e83c7bca74
4/10: http://c6210bb35e3be552a36926f
1/2: http://packagist.org/p/provider-latest$66261f27c4c75
8d72d38a9936aa29766f01b6cace0879b150530eeaf6cf95805.json
2/2: http://packagist.org/p/provider-2017-01$632074e1b35b
48aafcac76fafae8885e92883add411eccc813e7da4701bfb1e.json
```



# Pengenalan Yii Framework

## Testing Hasil Instalasi Yii

- Akses [http://localhost/nama\\_folder/web](http://localhost/nama_folder/web)

My Company

HomeAboutContactLogin

# Congratulations!

You have successfully created your Yii-powered application.

Get started with Yii

### Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea

### Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea

### Heading

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea





# Pengenalan Yii Framework

## Testing Hasil Instalasi Yii

- Akses [http://localhost/nama\\_folder/requirements.php](http://localhost/nama_folder/requirements.php)

### Yii Application Requirement Checker

#### Description

This script checks if your server configuration meets the requirements for running Yii application. It checks if the server is running the right version of PHP, if appropriate PHP extensions have been loaded, and if php.ini file settings are correct.

There are two kinds of requirements being checked. Mandatory requirements are those that have to be met to allow Yii to work as expected. There are also some optional requirements being checked which will show you a warning when they do not meet. You can use Yii framework without them but some specific functionality may be not available in this case.

#### Conclusion

Your server configuration satisfies the minimum requirements by this application.

Please pay attention to the warnings listed below and check if your application will use the corresponding features.

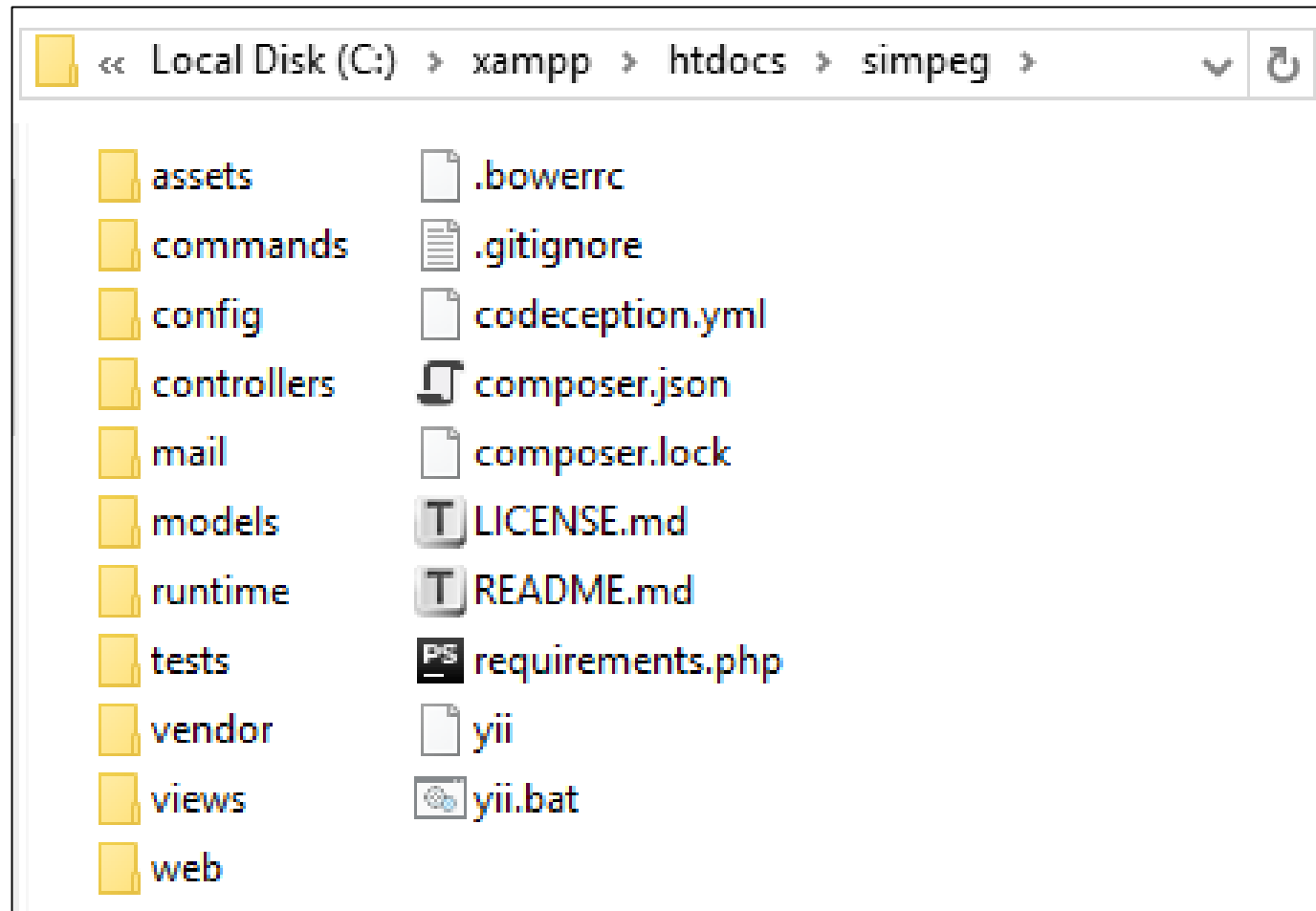
#### Details

Name	Result	Required By	Memo
PHP version	Passed	Yii Framework	PHP 5.4.0 or higher is required.
Reflection extension	Passed	Yii Framework	
PCRE extension	Passed	Yii Framework	
SPL extension	Passed	Yii Framework	



# Pengenalan Yii Framework

## Arsitektur Yii





# Pengenalan Yii Framework

## Arsitektur Yii

Nama File/Folder	Keterangan
<b>assets/</b>	berisi definisi assets yang digunakan pada aplikasi (css, js)
<b>commands/</b>	berisi contoh penerapan aplikasi console (CMD)
<b>config/</b>	berisi file konfigurasi aplikasi
<b>console.php</b>	file konfigurasi aplikasi console
<b>web.php</b>	file konfigurasi aplikasi web
<b>db.php</b>	file konfigurasi basis data
<b>controllers/</b>	berisi file class-class controllers aplikasi
<b>mail/</b>	berisi contoh template email
<b>models/</b>	berisi file class-class model aplikasi
<b>runtime/</b>	berisi file-file yang di-generate oleh Yii selama runtime, seperti logs dan cache
<b>tests/</b>	Berisi file-file untuk kebutuhan uji coba aplikasi
<b>vendor/</b>	berisi paket yang diinstalasi oleh Composer, termasuk core Yii sendiri.
<b>views/</b>	berisi file-file views aplikasi yang dikelompokkan menggunakan folder sesuai dengan nama controller-nya
<b>web/</b>	root dari aplikasi web, atau file yang boleh diakses oleh pengguna



# Konsep MVC di Yii

---

## Definisi MVC

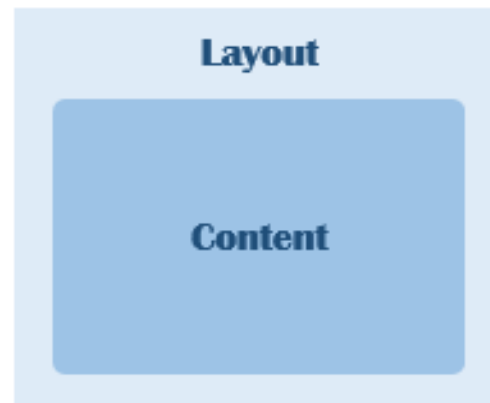
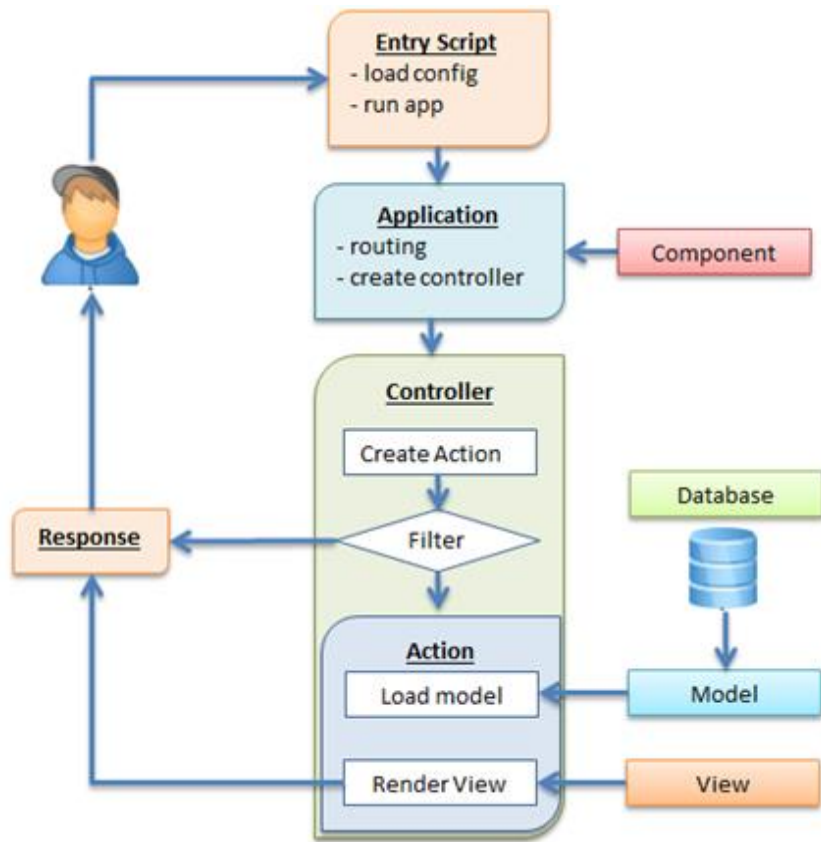
- MVC (Model – View – Controller) merupakan konsep dalam pengembangan aplikasi modern yaitu membagi aplikasi menjadi 3 bagian yaitu:
  - Model, data yang digunakan (database)
  - View, tampilan data (user interface)
  - Controller, pemroses dan pengendali input dan output (logic / bisnis proses)
- MVC di Yii tercermin dalam struktur aplikasinya, yaitu terdapat folder models (berisi semua file models), views (berisi semua file views), dan controllers (berisi semua file controller)



# Konsep MVC di Yii

## Alur Kerja 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`





# Konsep MVC di Yii

---

## Membuat Hello World (Only Controller)

- Tambahkan method actionHelloworld di SiteController

@app\controllers\SiteController.php

```
public function actionHelloworld()  
{  
    echo "Halo dunia!";  
}
```

- Akses [http://localhost/nama\\_folder/web/index.php?r=site/helloworld](http://localhost/nama_folder/web/index.php?r=site/helloworld)





# Konsep MVC di Yii

---

## Aturan Routing

- Controller
  - Format: `NamaController`
  - URL: `index.php?r=nama`
  - Contoh: `SiteController` → `index.php?r=site`
- Action pada Controller
  - Format: `actionNama`
  - URL: `index.php?r=namacontroller/nama`
  - Contoh: `SiteController` → `actionOke` → `index.php?r=site/oke`
- Penulisan URL
  - Format URL menggunakan lowercase
  - Jika dua kata gunakan format camelCase
  - Contoh: `actionHelloWorld` → `index.php?r=site/hello-world`



# Konsep MVC di Yii

## Membuat Hello World Lanjutan (Controller + View)

- Tambahkan method actionHelloWorld di SiteController & view halo

@app\controllers\SiteController.php

```
public function actionHelloWorld()  
{  
    return $this->render("halo");  
}
```

@app\views\site\halo.php

```
<h1>Hallo dunia!</h1>  
<p> Lorem ipsum sit dolor amert ..</p>
```

- Akses [http://localhost/nama\\_folder/web/index.php?r=site/hello-world](http://localhost/nama_folder/web/index.php?r=site/hello-world)







# Konsep MVC di Yii

## Membuat Hello World Lanjutan (Controller + View)

- Tambahkan parameter nama pada method actionHelloWorld

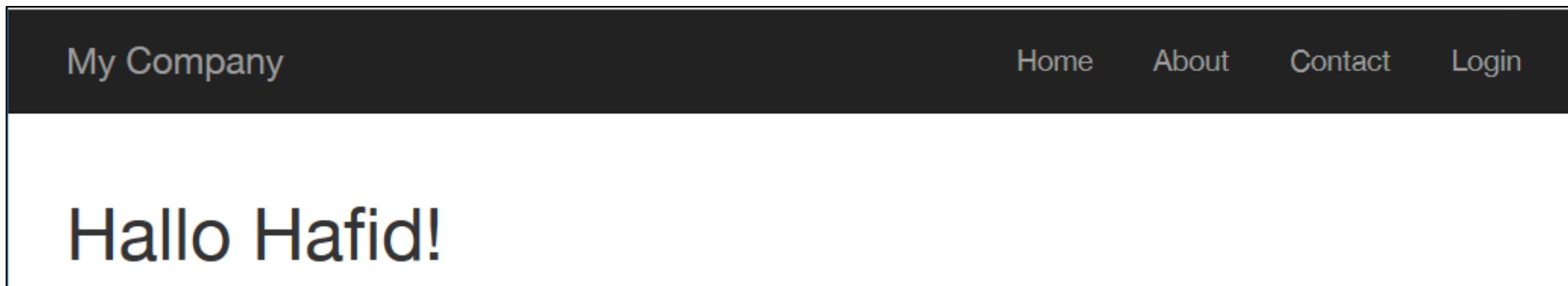
@app\controllers\SiteController.php

```
public function actionHelloWorld($nama)
{
    return $this->render("halo",[
        "nama" =>$nama
    ]);
}
```

@app\views\site\halo.php

```
<h1>Hallo <?= $nama ?></h1>
<p> Lorem ipsum sit dolor amert ..</p>
```

- Akses [http://localhost/nama\\_folder/web/index.php?r=site/hello-world&nama=Hafid](http://localhost/nama_folder/web/index.php?r=site/hello-world&nama=Hafid)





# Konsep MVC di Yii

---

## Membuat Formulir Komentar (Controller + View + Model)

- Desain form komentar dengan dua input nama dan pesan

**Tinggal komentar Anda!**

Nama	<input type="text" value="Agung Hercules"/>
Pesan	<input type="text" value="Hallo apa kabar?"/>



# Konsep MVC di Yii

---

## Membuat Formulir Komentar (Controller + View + Model)

- Buat model Komentar

@app\models\Komentar.php

```
namespace app\models;

class Komentar extends \yii\base\Model
{
    public $nama;
    public $pesan;

    public function rules()
    {
        return [
            [['nama', 'pesan'], 'required']
        ];
    }
}
```



# Konsep MVC di Yii

---

## Membuat Formulir Komentar (Controller + View + Model)

- Buat controller baru untuk menangani formulir komentar
- Buat method actionForm untuk menampilkan formulir komentar

@app\controllers\KomentarController.php

```
namespace app\controllers;
use app\models\Komentar;
class KomentarController extends \yii\web\Controller
{
    public function actionForm()
    {
        $model = new Komentar;
        return $this->render("form",["model" => $model]);
    }
}
```



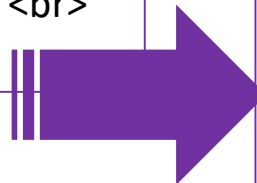
# Konsep MVC di Yii

## Membuat Formulir Komentar (Controller + View + Model)

- Buat view untuk menampilkan formulir komentar

@app\views\komentar\form.php

```
<h1>Tinggalkan Komentar Anda!</h1>
<form method="POST" action="index.php?komentar/kirim">
Nama   <input type="text" name="Komentar[nama]"> <br>
Pesan  <textarea name="Komentar[pesan]"></textarea> <br>
<button type="submit">Simpan</button>
```



@app\views\komentar\form.php

```
<?php
use yii\bootstrap\ActiveForm;
use yii\helpers\Html;
?>
<h1>Tinggalkan Komentar Anda!</h1>
<?php
$form = ActiveForm::begin([
    "action" => ["komentar/kirim"]
]);
echo $form->field($model, "nama");
echo $form->field($model, "pesan")->textarea();
echo Html::submitButton("Kirim",[
    "class"=>"btn btn-primary"
]);
\yii\bootstrap\ActiveForm::end();
```



# Konsep MVC di Yii

## Membuat Formulir Komentar (Controller + View + Model)

- Testing formulir komentar (kosong)

[http://localhost/nama\\_folder/web/index.php?r=komentar/form](http://localhost/nama_folder/web/index.php?r=komentar/form)

**Tinggalkan Komentar Anda!**

**Nama**

  
**Pesan**

Kirim

**Tinggalkan Komentar Anda!**

**Nama**

Nama cannot be blank.

**Pesan**

Pesan cannot be blank.

Kirim



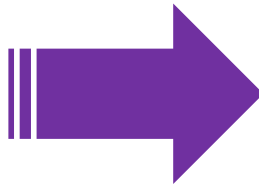
# Konsep MVC di Yii

## Membuat Formulir Komentar (Controller + View + Model)

- Tambahkan method actionKirim pada controller Komentar, untuk memproses hasil submit form

@app\controllers\KomentarController.php

```
public function actionKirim()
{
    print_r($_POST);
}
```



@app\controllers\KomentarController.php

```
public function actionKirim()
{
    $model = new Komentar;
    $model->load(Yii::$app->request->post());
    return $this->render("hasil",[
        "model" => $model
    ]);
}
```



# Konsep MVC di Yii

---

## Membuat Formulir Komentar (Controller + View + Model)

- Buat view untuk menampilkan hasilnya

@app\views\komentar\hasil.php

```
<?php  
echo "Nama : ".$model->nama . "<br>";  
echo "Pesan : ".$model->pesan;
```





# Konsep MVC di Yii

## Membuat Formulir Komentar (Controller + View + Model)

- Testing formulir komentar

[http://localhost/nama\\_folder/web/index.php?r=komentar/form](http://localhost/nama_folder/web/index.php?r=komentar/form)

My Company

## Tinggalkan Komentar Anda!

Nama

Agung Hercules

Pesan

Halo apa kabar?

Kirim

My Company

Nama : Agung Hercules

Pesan : Halo apa kabar?



# Bekerja Dengan Database

## Pengaturan Koneksi Database

- Sesuaikan nama database, username dan password database

@app\config\db.php

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

- Buat tabel komentar & isi beberapa data dummy

tabel komentar

id	primary key
nama	varchar(50)
pesan	text



# Bekerja Dengan Database

---

## Menjalankan Query Tanpa Model

- Menampilkan data dengan query select
  - Menampilkan semua data pada tabel komentar (select \* from komentar)  
`$komentar = \Yii::$app->db->createCommand('SELECT * FROM komentar')->queryAll();`
  - Menampilkan data komentar yang namanya "hafid"  
`$komentar = \Yii::$app->db->createCommand('SELECT * FROM komentar WHERE nama="hafid"')->queryOne();`
  - Menampilkan jumlah data komentar  
`$jumlah_komentar = \Yii::$app->db->createCommand('SELECT count(id) FROM komentar')->queryScalar();`
  - Menampilkan list data field nama komentar  
`$nama_komentars = \Yii::$app->db->createCommand('SELECT nama FROM komentar')->queryColumn();`



# Bekerja Dengan Database

---

## Menjalankan Query Tanpa Model

- Memanipulasi data dengan query insert, update, delete
  - Menambah data tabel komentar (insert into komentar(nama, pesan) values('Hafid','halo'))  

```
\Yii::$app->db->createCommand()->insert('komentar',['nama'=>'Hafid', 'pesan'=>'Halo'])  
->execute();
```
  - Menambah data tabel komentar secara batch  

```
\Yii::$app->db->createCommand()->batchInsert('komentar',['nama', 'pesan'],[  
    ['Hafid', 'Halo'], [ 'Budi', 'Hola'], [ 'Arfan', 'Hehe']  
])->execute();
```
  - Mengedit data tabel komentar yang ID nya 5  

```
\Yii::$app->db->createCommand()->update('komentar',['pesan'=>'Halo bandung'],'id=5')  
->execute();
```
  - Menghapus data tabel komentar yang ID nya 5  

```
\Yii::$app->db->createCommand()->delete('komentar','id=5')->execute();
```



# Bekerja Dengan Database

---

## Menjalankan Query Tanpa Model

- Praktik menampilkan data

```
@app\controllers\QueryController.php

public function actionSelect()
{
    $komentars = \Yii::$app->db->createCommand('SELECT * FROM komentar')->queryAll();
    foreach($komentars as $komentar){
        echo $komentar['nama']. " = " . $komentar['pesan']. " <br> ";
    }
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=query/select](http://localhost/nama_folder/web/index.php?r=query/select)



# Bekerja Dengan Database

---

## Menjalankan Query Tanpa Model

- Praktik menambah data

@app\controllers\QueryController.php

```
public function actionInsert($nama, $pesan)
{
    \Yii::$app->db->createCommand()->insert('komentar',['nama'=>$nama, 'pesan'=>$pesan])->execute();
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=query/insert&nama=Hafid&pesan=halo](http://localhost/nama_folder/web/index.php?r=query/insert&nama=Hafid&pesan=halo)



# Bekerja Dengan Database

---

## Menjalankan Query Tanpa Model

- Praktik menambah data

@app\controllers\QueryController.php

```
public function actionInsert($nama, $pesan)
{
    \Yii::$app->db->createCommand()->insert('komentar',['nama'=>$nama, 'pesan'=>$pesan])->execute();
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=query/insert&nama=Hafid&pesan=halo](http://localhost/nama_folder/web/index.php?r=query/insert&nama=Hafid&pesan=halo)



# Bekerja Dengan Database

---

## Menjalankan Query Tanpa Model

- Praktik mengedit data

@app\controllers\QueryController.php

```
public function actionUpdate($id, $nama, $pesan)
{
    \Yii::$app->db->createCommand()->insert('komentar',['nama'=>$nama, 'pesan'=>$pesan], 'id'=>$id)->execute();
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=query/update&id=5&nama=Hafid&pesan=halo](http://localhost/nama_folder/web/index.php?r=query/update&id=5&nama=Hafid&pesan=halo)





# Bekerja Dengan Database

---

## Menjalankan Query Tanpa Model

- Praktik menghapus data

@app\controllers\QueryController.php

```
public function actionDelete($id)
{
    \Yii::$app->db->createCommand()->delete('komentar',['id'=>$id])->execute();
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=query/delete&id=5](http://localhost/nama_folder/web/index.php?r=query/delete&id=5)



# Bekerja Dengan Database

## Menghubungkan Model dengan Database

- Tambahkan fungsi tableName, dan extends ActiveRecord

```
@app\models\Komentar.php

namespace app\models;
class Komentar extends \yii\db\ActiveRecord
{
    public static function tableName()
    {
        return "komentar";
    }

    public function rules()
    {
        return [
            [ "nama", "pesan", "required" ]
        ];
    }
}
```



# Bekerja Dengan Database

---

## Menjalankan Query Menggunakan Model: Active Record

- Mengakses data atau query select
  - Menampilkan semua data pada tabel komentar (select \* from komentar)  
`Komentar::find()->all();` atau `Komentar::findAll();`
  - Menampilkan semua data komentar dengan urutan  
`Komentar::find()->orderBy("nama")->all();`
  - Menampilkan semua data komentar hanya field nama  
`Komentar::find()->select("nama")->all();`
  - Menampilkan data komentar yang namanya "hafid"  
`Komentar::find()->where(["nama"=>"hafid"])->one();`  
atau  
`Komentar::findOne(["nama"=>"hafid"]);`



# Bekerja Dengan Database

## Menjalankan Query Menggunakan Model: Active Record

- Praktik menampilkan data

@app\controllers\ActiveRecordController.php

```
public function actionSelect()
{
    $komentars = Komentar::find()->all();
    foreach($komentars as $komentar){
        echo $komentar->nama. " = " . $komentar->pesan. " <br> ";
    }
}
```

- Catatan: karena kita menggunakan model Komentar, maka pastikan telah mendeklarasikan model Komentar beserta namespace lengkapnya dibagian atas kode.
  - use app\models\Komentar;
- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=active-record/select](http://localhost/nama_folder/web/index.php?r=active-record/select)



# Bekerja Dengan Database

---

## Menjalankan Query Menggunakan Model: Active Record

- Menambahkan data atau query insert

```
$komentar = new Komentar;
```

```
$komentar->nama = "Hafid Mukhlasin";
```

```
$komentar->pesan = "Halo halo bandung!";
```

```
$komentar->save();
```



# Bekerja Dengan Database

---

## Menjalankan Query Menggunakan Model: Active Record

- Praktik menambahkan data komentar

@app\controllers\ActiveRecordController.php

```
public function actionInsert($nama, $pesan)
{
    $komentar = new Komentar;
    $komentar->nama = $nama;
    $komentar->pesan = $pesan;
    $komentar->save();
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=active-record/insert&nama=X&pesan=Y](http://localhost/nama_folder/web/index.php?r=active-record/insert&nama=X&pesan=Y)



# Bekerja Dengan Database

---

## Menjalankan Query Menggunakan Model: Active Record

- Mengupdate data atau query update

```
$komentar = Komentar::findOne(["id"=>2]);  
$komentar->nama = "Wida";  
$komentar->pesan = "hehehehe";  
$komentar->save();
```



# Bekerja Dengan Database

---

## Menjalankan Query Menggunakan Model: Active Record

- Praktik mengupdate data komentar

@app\controllers\ActiveRecordController.php

```
public function actionUpdate($id, $nama, $pesan)
{
    $komentar = Komentar::findOne(["id"=>$id]);
    $komentar->nama = $nama;
    $komentar->pesan = $pesan;
    $komentar->save();
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=active-record/update&id=2&nama=X&pesan=Y](http://localhost/nama_folder/web/index.php?r=active-record/update&id=2&nama=X&pesan=Y)





# Bekerja Dengan Database

---

## Menjalankan Query Menggunakan Model: Active Record

- Menghapus data atau query delete  
\$komentar = Komentar::findOne(["id"=>2]);  
\$komentar->delete();



# Bekerja Dengan Database

---

## Menjalankan Query Menggunakan Model: Active Record

- Praktik menghapus data komentar

@app\controllers\ActiveRecordController.php

```
public function actionDelete($id)
{
    $komentar = Komentar::find(["id"=>$id]);
    $komentar->delete();
}
```

- Akses
  - [http://localhost/nama\\_folder/web/index.php?r=active-record/delete&id=2](http://localhost/nama_folder/web/index.php?r=active-record/delete&id=2)



# Bekerja Dengan Database

---

## Menampilkan Data Komentar

- Tambahkan actionTampil pada KomentarController

@app\controllers\KomentarController.php

```
public function actionTampil()
{
    $model = Komentar::findAll();
    return $this->render("tampil",[
        "model" => $model
    ]);
}
```



# Bekerja Dengan Database

## Mengatur Tampilan Daftar Komentar

- Tambahkan View

@app\views\komentar\tampil.php

```
<h1>Daftar Komentar!</h1>
<table class="table">
<tr>
  <th>No</th>
  <th>Nama</th>
  <th>Pesan</th>
</tr>
<?php
$no=1;
foreach($model as $komentar){
  echo "<tr>";
  echo "<td>".$no++."</td>";
  echo "<td>".$komentar->nama."</td>";
  echo "<td>".$komentar->pesan."</td>";
  echo "</tr>";
}
?>
</table>
```



# Bekerja Dengan Database

## Testing Menampilkan Data

- Akses [http://localhost/nama\\_folder/web/index.php?r=komentar/tampil](http://localhost/nama_folder/web/index.php?r=komentar/tampil)

My Company		
Daftar Komentar!		
No	Nama	Pesan
1	Agung Hercules	Halo apa kabar?
2	Hafid	Good bye



# Bekerja Dengan Database

## Menampilkan Data Dengan Batasan Per Halaman

- use yii\data\Pagination;

```
@app\controllers\KomentarController.php

public function actionIndex()
{
    $query = Komentar::find();
    $count = $query->count();
    $pagination = new Pagination([
        'totalCount' => $count,
        'defaultPageSize' => 5,
    ]);
    $model = $query->offset($pagination->offset)
        ->limit($pagination->limit)
        ->all();
    return $this->render("index",[
        'model' => $model,
        'pagination' => $pagination,
    ]);
}
```



# Bekerja Dengan Database

## Menampilkan Data Dengan Batasan Per Halaman

- modifikasi view
- use `\yii\widgets\LinkPager`;

@app\views\komentar\index.php

```
<table class="table">
...
<?php
$no=1;
foreach($model as $komentar){
    echo "<tr>";
    echo "<td>".$no++."</td>";
    echo "<td>".$komentar->nama."</td>";
    echo "<td>".$komentar->pesan."</td>";
    echo "</tr>";
}
?>
</table>
<?php
echo LinkPager::widget([
    'pagination' => $pagination,
]);
```



# Bekerja Dengan Database

---

## Menyimpan Data Ke Database Menggunakan Active Record

- Ubah actionKirim pada KomentarController

@app\controllers\KomentarController.php

```
public function actionKirim()
{
    $model = new Komentar;
    $model->load(\Yii::$app->request->post());
    $model->save();
    return $this->render("hasil",[
        "model" => $model
    ]);
}
```





# Bekerja Dengan Database

## Testing

- Kirim isi formulir kemudian cek database tabel komentar

[http://localhost/nama\\_folder/web/index.php?r=komentar/form](http://localhost/nama_folder/web/index.php?r=komentar/form)

My Company

## Tinggalkan Komentar Anda!

Nama

Bambang Pamungkas

Pesan

Futsal Yuuk

Kirim

My Company

Nama : Bambang Pamungkas  
Pesan : Futsal Yuuk

+ Opsi				id	nama	pesan
<input type="checkbox"/>	Ubah	Salin	Hapus	1	Agung Hercules	Halo apa kabar?
<input type="checkbox"/>	Ubah	Salin	Hapus	3	Hafid	Good bye
<input type="checkbox"/>	Ubah	Salin	Hapus	4	Bambang Pamungkas	Futsal Yuuk



# Bekerja Dengan Database

## Mengeset Informasi Proses

- Modifikasi actionKirim pada KomentarController, dengan mengeset flash message

@app\controllers\KomentarController.php

```
public function actionKirim()
{
    $model = new \app\models\Komentar;
    $model->load(\Yii::$app->request->post());
    if($model->save()){
        // berhasil di simpan
    }
    else{
        // gagal disimpan
    }
    return $this->render("hasil",[
        "model" => $model
    ]);
}
```

```
if($model->save()){
    \Yii::$app->session->setFlash("success", "Berhasil disimpan");
}
else{
    \Yii::$app->session->setFlash("error", "Gagal disimpan");
}
```



# Bekerja Dengan Database

## Menampilkan Informasi Proses

- Modifikasi view hasil, untuk menampilkan flash message

@app\views\komentar\hasil.php

```
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>';  
}
```

### CEK SESSION FLASH MESSAGE?

```
if (\Yii::$app->session->hasFlash('success')){
```

### TAMPILKAN FLASH MESSAGE

```
    echo '<div class="alert alert-success">';  
    echo \Yii::$app->session->getFlash('success');  
    echo '</div>';
```



# Bekerja Dengan Database

## Testing Flash Message

- Akses: [http://localhost/nama\\_folder/web/index.php?r=komentar/form](http://localhost/nama_folder/web/index.php?r=komentar/form)

My Company

### Tinggalkan Komentar Anda!

**Nama**

**Pesan**

**Kirim**

My Company

Berhasil disimpan

Nama : Novi  
Pesan : Makan makan yuk!



# Bekerja Dengan Database

## Membuat Widget "Flash Message"

@app\widgets\FlashMessage.php

```
namespace app\widgets;
class FlashMessage 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>';
        }
    }
}
```



# Bekerja Dengan Database

---

## Cara Menggunakan Widget "Flash Message" (Lokal)

@app\views\komentar\hasil.php

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



# Bekerja Dengan Database

---

## Cara Menggunakan Widget "Flash Message" (Global)

@app/views/layouts/main.php

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



# Otentikasi

## Otentikasi Menggunakan Database

- Modifikasi model User di @app\models\User.php
  1. Ubah extends \yii\base\Object menjadi \yii\db\ActiveRecord
  2. Hapus semua property, ganti menjadi fungsi static tableName()

@app/models/User.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',
```



@app/models/User.php

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





# Otentikasi

---

## Otentikasi Menggunakan Database

### 3. Ubah fungsi findIdentity, menjadi bentuk ActiveRecord

@app/models/User.php

```
public static function findIdentity($id)
{
    return static::findOne(['id' => $id, 'status' => 1]);
}
```

### 4. Lakukan hal yang sama untuk fungsi findByUsername

@app/models/User.php

```
public static function findByUsername($username)
{
    return static::findOne(['username' => $username, 'status' => 1]);
}
```



# Otentikasi

---

## Otentikasi Menggunakan Database

5. Sesuaikan nama property yang dipakai pada fungsi `getAuthKey`.

@app/models/User.php

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

6. Lakukan hal yang sama untuk fungsi `validateAuthKey`

@app/models/User.php

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



# Otentikasi

## Otentikasi Menggunakan Database

7. Ubah fungsi validatePassword menggunakan standard keamanan password Yii

@app/models/User.php

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

8. Tambahkan fungsi behaviors untuk memasukkan TimestampBehavior

@app/models/User.php

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



# Otentikasi

## Otentikasi Menggunakan Database

- Untuk testing, buat fungsi untuk membuat admin baru

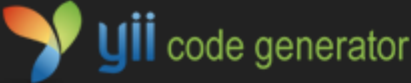
@app/controllers/SiteController.php

```
public function actionCreateAdmin()
{
    $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','Admin berhasil dibuat');
    }
    else{
        Yii::$app->session->setFlash('error','Admin gagal dibuat');
    }
    return $this->redirect(["index"]);
}
```



# Gii: Code Generator

- Gii = tools official Yii dalam bentuk module untuk menggenerate kode program Yii seperti model, view, controller, dsb.
- [http://localhost/nama\\_folder/web/index.php?r=gii](http://localhost/nama_folder/web/index.php?r=gii)

 Home

## 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 controller class with one or several their corresponding views.

Start »



# Gii: Code Generator

---

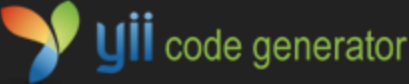
Ada 6 generator kode yaitu:

1. **Model Generator**  
Untuk generate class ActiveRecord (model) untuk tabel tertentu pada basis data.
2. **CRUD Generator**  
Untuk generate operasi CRUD berdasarkan class model tertentu
3. **Controller Generator**  
Untuk generate class Controller
4. **Form Generator**  
Untuk generate tampilan formulir berdasarkan class model tertentu
5. **Module Generator**  
Untuk generate template dari module di Yii, jika kita ingin membuat sub aplikasi.
6. **Extension Generator**  
Untuk generate template dari extension Yii, untuk membuat extension sendiri.



# Gii: Code Generator

## 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

unit

Model Class

Unit

Namespace

app\models

#### Code Template

default (C:\xampp\htdocs\simpeg\vendor\yiisoft\yii2-gii\generators\model/default)

Preview



# Gii: Code Generator


## Model Generator

PreviewGenerate

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

☒ Create☒ Unchanged☒ Overwrite

Code File	Action	<input checked="" type="checkbox"/>
models\Unit.php	create	<input checked="" type="checkbox"/>



Preview

The code has been generated successfully.

Generating code using template "C:\xampp\htdocs\simpeg\vendor\yiisoft\yii2-gii\generators\model/default"...  
generated models\Unit.php  
done!





# Gii: Code Generator

## Model Generator

@app/models/Unit.php

```
namespace app\models;
use Yii;
class Unit extends \yii\db\ActiveRecord
{
    public static function tableName()
    {
        return 'unit';
    }

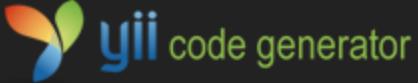
    public function rules()
    {
        return [
            [['name'], 'required'],
            [['name'], 'string', 'max' => 255],
        ];
    }
}
```

```
public function attributeLabels()
{
    return [
        'id' => 'ID',
        'name' => 'Name',
    ];
}
```



# Gii: Code Generator

## CRUD Generator

Home Help Application

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, Delete) operations for the specified data model.

**Model Class**

**Search Model Class**

**Controller Class**

**View Path**



# Gii: Code Generator

## CRUD Generator

PreviewGenerate

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

☒ Create ☒ Unchanged ☒ Overwrite

Code File	Action	<input checked="" type="checkbox"/>
controllers\UnitController.php	create	<input checked="" type="checkbox"/>
models\UnitSearch.php	create	<input checked="" type="checkbox"/>
views\unit\_form.php	create	<input checked="" type="checkbox"/>
views\unit\_search.php	create	<input checked="" type="checkbox"/>
views\unit\create.php	create	<input checked="" type="checkbox"/>
views\unit\index.php	create	<input checked="" type="checkbox"/>
views\unit\update.php	create	<input checked="" type="checkbox"/>
views\unit\view.php	create	<input checked="" type="checkbox"/>



# Gii: Code Generator

---

## CRUD Generator

Preview

The code has been generated successfully.

```
Generating code using template "C:\xampp\htdocs\simpeg\vendor\yiis  
generated controllers\UnitController.php  
generated models\UnitSearch.php  
generated views\unit\_form.php  
generated views\unit\_search.php  
generated views\unit\create.php  
generated views\unit\index.php  
generated views\unit\update.php  
generated views\unit\view.php  
done!
```



# Gii: Code Generator

## Testing Hasil CRUD Generator

[http://localhost/nama\\_folder/web/index.php?r=unit](http://localhost/nama_folder/web/index.php?r=unit)

My Company

HomeAboutContactLogin

Home / Units

## Units

Create Unit

#	ID	Name	
	<input type="text"/>	<input type="text"/>	

No results found.



# Gii: Code Generator

## Testing Hasil CRUD Generator

Masuk ke create unit

My Company [Home](#) [About](#) [Contact](#) [Login](#)

[Home](#) / [Units](#) / Create Unit

## Create Unit

**Name**

My Company [Home](#) [About](#) [Contact](#) [Login](#)

[Home](#) / [Units](#) / Direktorat Jenderal Pajak

## Direktorat Jenderal Pajak

ID	1
Name	Direktorat Jenderal Pajak



# Gii: Code Generator

## Testing Hasil CRUD Generator

kembali ke index unit

My Company




HomeAboutContactLogin

Home / Units

## Units

Create Unit

Showing 1-1 of 1 item.

#	ID	Name	
	<input type="text"/>	<input type="text"/>	
1	1	Direktorat Jenderal Pajak	  



# Modifikasi CRUD

---

- Modifikasi Layout
  - Modifikasi Menu
  - Modifikasi Warna Navbar
- Modifikasi View
  - Modifikasi Form
  - Modifikasi Gridview
  - Modifikasi DetailView
- Modifikasi Controller
- Modifikasi Model





# Modifikasi CRUD

## Modifikasi Layout : Menu Aplikasi

@app/views/layouts/main.php

```
$status_menu = (Yii::$app->user->isGuest)?false:true;
echo Nav::widget([
    'options' => ['class' => 'navbar-nav navbar-right'],
    'items' => [
        ['label' => 'Home', 'url' => ['/site/index']],

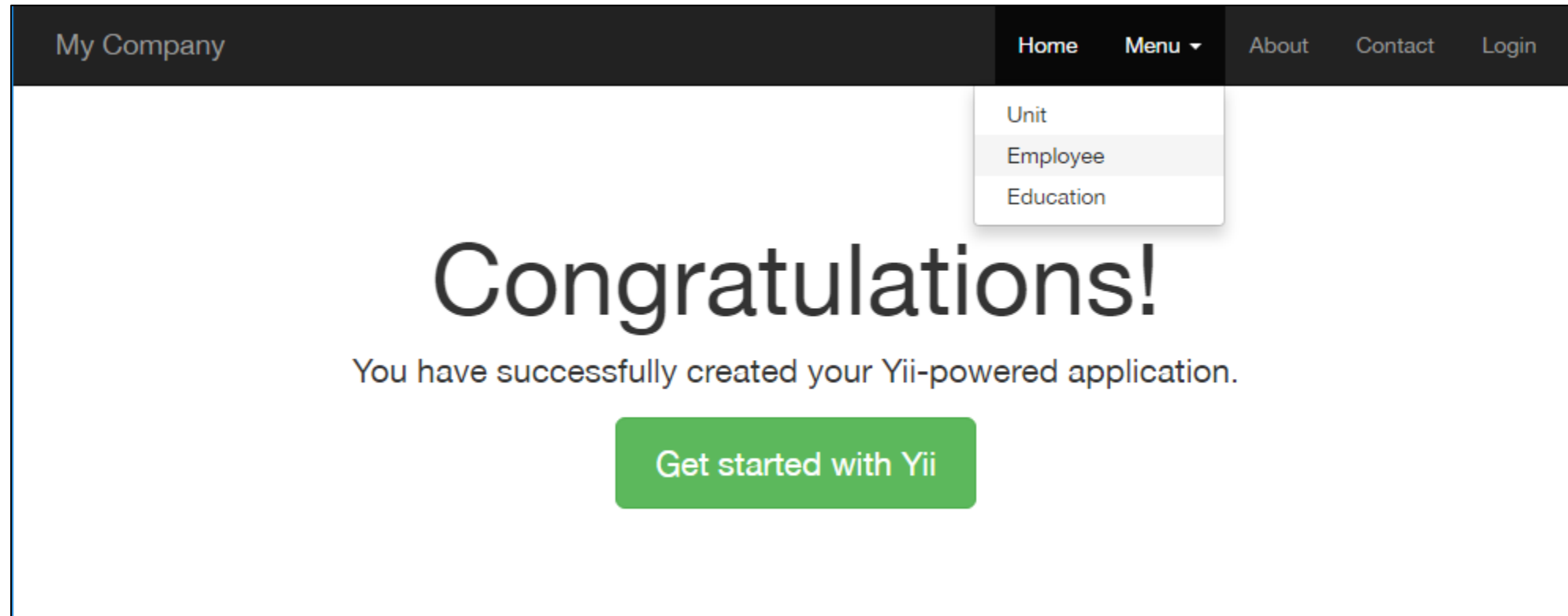
        ['label' => 'Menu', 'visibility'=>$status_menu,'items' => [
            ['label' => 'Unit', 'url' => ['/unit/index']],
            ['label' => 'Employee', 'url' => ['/employee/index']],
            ['label' => 'Education', 'url' => ['/education/index']],
        ]],

        ['label' => 'About', 'url' => ['/site/about']],
        ['label' => 'Contact', 'url' => ['/site/contact']],
        Yii::$app->user->isGuest ? (
            ['label' => 'Login', 'url' => ['/site/login']]
        ) : (
            ['label' => 'Logout', 'url' => ['/site/logout']],
            ['label' => 'Unit', 'url' => ['/unit/index']],
            ['label' => 'Employee', 'url' => ['/employee/index']],
            ['label' => 'Education', 'url' => ['/education/index']],
        )
    ]
]);
```



# Modifikasi CRUD

## Modifikasi Layout : Menu Aplikasi





# Modifikasi CRUD

---

## Modifikasi Layout : Warna Navbar

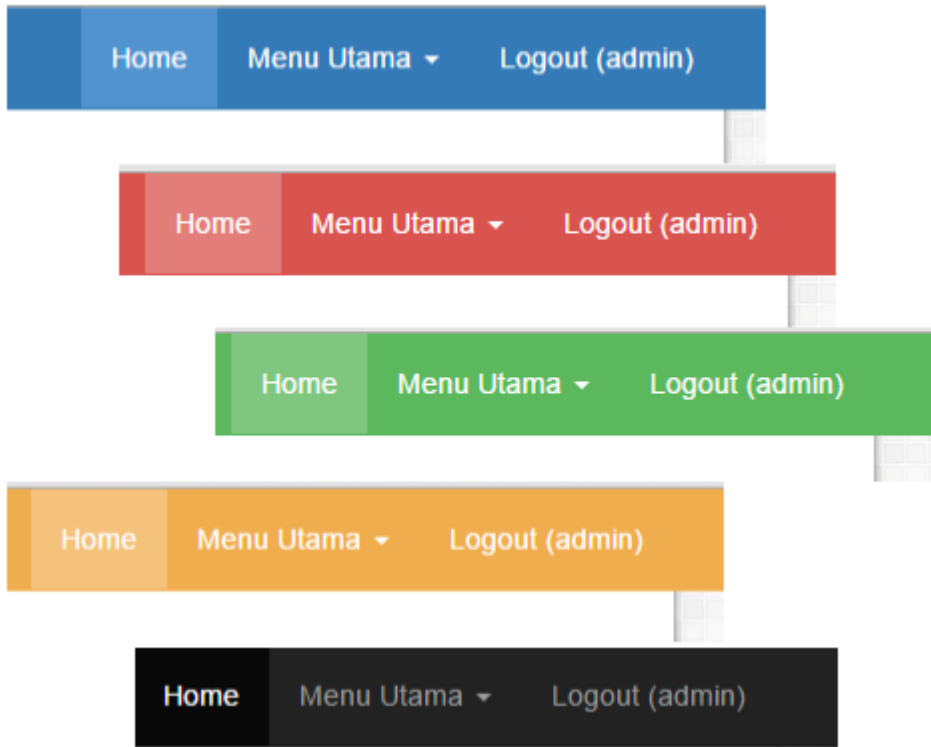
- Gunakan generator <https://work.smarchal.com/twbscolor/>
- Tentukan warna pilihan kita, bisa mengikuti aturan di <http://getbootstrap.com/css/#less-variables-colors>
- Copy CSS yang dihasilkan
- Ubah navbar-default menjadi navbar-inverse
- Letakkan CSS tsb di @app\web\site.php





# Modifikasi CRUD

## Modifikasi Layout : Warna Navbar



```
@brand-primary: darken(#428bca, 6.5%); // #337ab7  
@brand-success: #5cb85c;  
@brand-info: #5bc0de;  
@brand-warning: #f0ad4e;  
@brand-danger: #d9534f;
```



# Modifikasi CRUD

## Modifikasi Layout : Warna Navbar

@app/web/site.css

```
/* PANEL */
.navbar-inverse{
  background-color: #337ab7;
  border-color: #337ab7;
}

/* FONT */
.navbar-inverse .navbar-brand,
.navbar-inverse .btn-link,
.navbar-inverse .navbar-nav > li > a{
  color: #fff;
}

/* BG PANEL SOFT */
.navbar-inverse .navbar-nav > .active > a,
.navbar-inverse .navbar-nav > .active > a:hover,
.navbar-inverse .navbar-nav > .active > a:focus,
.navbar-inverse .navbar-nav > .open > a,
.navbar-inverse .navbar-nav > .open > a:hover,
.navbar-inverse .navbar-nav > .open > a:focus{
  color:#fff;
  background-color: #5093CE;
}

/* ITEM DROPDOWN */
.dropdown-menu > .active > a,
.dropdown-menu > .active > a:hover,
.dropdown-menu > .active > a:focus{
  background-color: #337ab7;
}
```



# Modifikasi CRUD

## Modifikasi View: Form

File @app\views\nama\_controller\\_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 fields: 'Username' containing 'admin', 'Password' containing '.....', and 'Address' which is empty.

A form preview showing two 'Gender' dropdown menus. The first dropdown is closed, showing '- Pilih Gender'. The second dropdown is open, showing the same prompt and two options: 'Pria' and 'Wanita'.



# Modifikasi CRUD

## 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 CRUD

---

## Modifikasi View: Form

- dropDownList dari Tabel

```
<?php
use \yii\helpers\ArrayHelper;
use \app\models\Unit;
$units = ArrayHelper::map(Unit::find()->all(),"id","name");
echo $form->field($model, 'unit_id')->dropDownList($units);
?>
```

**Unit ID**  

Direktorat Jenderal Pajak

▼





# Modifikasi CRUD

---

## Modifikasi View: Form

- Masking input via widget

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

- 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>



# Modifikasi CRUD

---

## Modifikasi View: Form

- fileInput

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

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

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

**Certificate**  
 No file selected.



# Modifikasi CRUD

## 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 CRUD

## Modifikasi View: GridView

File @app\views\nama\_controller\index.php



- Default

```
<?= GridView::widget([
    'dataProvider' => $dataProvider,
    'filterModel' => $searchModel,
    'columns' => [
        ['class' => 'yii\grid\SerialColumn'],
        'user_id',
        'name',
        'born',
        'birth_day',
        'gender',
        // 'religion',
        // 'unit_id',
        // 'marital_status',
        // 'photo',
        ['class' => 'yii\grid\ActionColumn'],
    ],
]); ?>
```

### Employees

Create Employee

Showing 1-1 of 1 item.

#	User ID	Name	Born	Birth Day	Gender	
	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	
1	1	Hafid Mukhlasin	Jember	1986-04-30	1	 



# Modifikasi CRUD

---

## 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 CRUD

---

## 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 CRUD

---

## 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 CRUD

---

## 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 CRUD

---

## Modifikasi View: Gridview

- Modifikasi filter menjadi dropdown

```
[  
  "attribute" => "gender",  
  "filter"    => [1=>"Pria", 0=>"Wanita"],  
]
```



- Menonaktifkan filter

```
[  
  "attribute" => "birth_day",  
  "filter"    => false,  
]
```



# Modifikasi CRUD

---

## Modifikasi View: DetailView

- Hampir sama dengan Gridview



# Modifikasi CRUD

---

## Modifikasi Model: Rule

- Validasi input melalui method rule di Model

- `boolean` : yii\validators\BooleanValidator
- `captcha` : yii\captcha\CaptchaValidator
- `compare` : yii\validators\CompareValidator
- `date` : yii\validators\DateValidator
- `default` : yii\validators\DefaultValueValidator
- `double` : yii\validators\NumberValidator
- `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

<http://www.yiiframework.com/doc-2.0/guide-tutorial-core-validators.html>

<http://www.yiiframework.com/doc-2.0/yii-validators-validator.html>



# Modifikasi CRUD

---

```
// harus diisi
[["username"],"required"],

// harus numeric / integer
[["gender", "status"],"integer"],

// harus boolean true / false (1 / 0)
[["gender", "status"],"boolean"],

// harus string minimal 6 karakter maksimal 15
[["password"],"string","length"=>[6,15]],
[["nip"],"string","min"=>6,"max"=>15],

// harus uniq
[["username"],"unique"],

// harus berformat email
[["email"],"email"],

// harus berformat tanggal (yyyy-mm-dd)
[["birth_day"],"date"],

// harus numeric minimal 2juta maksimal 4juta
```

```
[["gaji"],"double", "min"=>2000000, "max"=>4000000],

// harus diantara nilai dalam array
[["religion"],"in", "range"=>["Islam","Kristen","Katolik"]],

// validasi file harus jpg, png dan gif
['file_name'], 'file', 'extensions' => 'jpg, png, gif',
'mimeTypes' => 'image/jpeg, image/png, image/gif'],

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

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



# Modifikasi CRUD

---

## 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()
    ];
}
```



# Modifikasi CRUD

---

## 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 CRUD

---

## Modifikasi Controller: File Upload

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

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



# Modifikasi CRUD

---

## Modifikasi Controller: Set Initial Gridview Filter & Sorter

```
$searchModel = new EmployeeSearch();  
$queryParams = [];  
  
if ($unit_id !== null) {  
    $queryParams['EmployeeSearch']['unit_id'] = $unit_id;  
}  
  
$queryParams = \yii\helpers\ArrayHelper::merge(  
    Yii::$app->request->getQueryParams(),  
    $queryParams  
);  
  
$dataProvider = $searchModel->search($queryParams);  
  
$dataProvider->getSort()->defaultOrder = [  
    'name' => SORT_ASC  
];
```





# Bekerja dengan Extension

---

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



# Bekerja dengan Extension

---

## 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:** "kartik-v/yii2-widgets": "\*"
- **Penggunaan:** (tergantung widgetnya), contoh

```
<?php
use kartik\widgets\SwitchInput;
echo $form->field($model, 'status')->widget(
    SwitchInput::classname(), [
        'type' => SwitchInput::RADIO
    ]
);
```



# Bekerja dengan Extension

## Yii2 Widget (Krajee Extension): Select2

```
<?php
use \yii\helpers\ArrayHelper;
use \app\models\Training;
use \kartik\widgets\Select2;
$trainings = ArrayHelper::map(Training::find()->all(),"id","name");
echo $form->field($model, 'training_id')->widget(
    Select2::classname(), [
        "data" => $trainings,
        "options" => ["placeholder" => "Pilih diklat"],
    ]
);
```

Training ID

Pilih diklat ...

Diklat Mencuci Piring

Diklat Menyapu Halaman



# Bekerja dengan Extension

---

## Yii2 Widget (Krajee Extension): SwitchInput

```
<?php
use \kartik\widgets\SwitchInput;
echo $form->field($model, 'gender')->widget(
    SwitchInput::classname(), [
        "pluginsOptions" => [
            "onText" => "Pria",
            "offText" => "Wanita",
        ],
    ]
);
```



# Bekerja dengan Extension

## Yii2 Widget (Krajee Extension): DatePicker

```
<?php
use \kartik\widgets\DatePicker;
echo $form->field($model, 'birth_day')->widget(
    DatePicker::classname(), [
        "options" => ["placeholder" => "Pilih tanggal .."],
        "language" => "id",
        "pluginsOptions" => [
            "format" => "yyyy-mm-dd",
            "autoclose" => true,
            "todayHighlight" => true,
            "todayBtn" => true,
        ],
    ]
);
```

The screenshot shows two instances of the DatePicker widget. The top instance is a simple text input with a calendar icon and the date 2015-02-15. The bottom instance is the same widget with a calendar popup open, showing the month of February 2015. The calendar has a grid of days with the 15th highlighted in blue. The popup also shows the current day (15) and the current month (Februari 2015).



# Impor & Ekspor Data

---

## Impor Data

- Mengimpor data yang dimaksudkan di sini adalah memasukkan data teks yang berasal dari *file* berformat *spreadsheet*/tabel (xls/xlsx/ods) ke tabel di basis data. Adakalanya pengguna telah memiliki data berformat *spreadsheets*, sehingga jika pada aplikasi tersedia fitur untuk mengimpor data tentu akan memudahkan pengguna untuk memasukkan data.
- Oleh karena itu, kita membutuhkan *library* yang *powerfull* untuk membaca *file* berformat *spreadsheet*. Pilihan penulis jatuh pada php-excel (<https://phpexcel.codeplex.com>), sebuah *library* yang cukup populer untuk menangani *spreadsheet* menggunakan PHP.
- Ada sebuah *extension* untuk Yii yang telah membungkus *library* tersebut yaitu hscstudio/yii2-export (<https://github.com/hscstudio/yii2-export>).

```
composer require --prefer-dist hscstudio/yii2-export "1.0.0"
```



# Impor & Ekspor Data

---

## Impor Data

- Siapkan data excel yang akan diimport

	A	B	
1	<b>NO</b>	<b>NAMA</b>	
2	1	DJP	
3	2	BPPK	
4	3	DJPBN	
5	4	SETJEN	
6	5	ITEN	
7			
8			



# Impor & Ekspor Data

## Impor Data

### CONTROLLER

```
public function actionImport()
{
    $model = new \yii\base\DynamicModel([
        'fileImport' => 'File Import',
    ]);
    $model->addRule(['fileImport'], 'required');
    $model->addRule(['fileImport'], 'file', ['extensions'=>'xls,xlsx'],['maxSize'=>1024*1024]);

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





# Impor & Ekspor Data

## Impor Data

VIEW

```
<?php
use yii\widgets\ActiveForm;
use yii\helpers\Html;
?>
<h1>Import Data</h1>
<?php
$form = ActiveForm::begin([
    'options' => ['enctype' => 'multipart/form-data'],
]) ?>
<?= $form->field($model,'fileImport')->fileInput() ?>
<?= Html::submitButton('Import',['class'=>'btn btn-primary']) ?>
<?php ActiveForm::end() ?>
```



# Impor & Ekspor Data

## Impor Data

### CONTROLLER

```
public function actionImport()
{
    $model = new \yii\base\DynamicModel([
        'fileImport' => 'File Import',
    ]);
    $model->addRule(['fileImport'], 'required');
    $model->addRule(['fileImport'], 'file',
        ['extensions'=>'xls,xlsx'],['maxSize'=>1024*1024]);
    if (Yii::$app->request->post()) {
        $model->fileImport =
        \yii\web\UploadedFile::getInstance($modelImport,
        'fileImport');
        if ($model->fileImport && $model->validate()) {
            $inputFileType = \PHPExcel_IOFactory::identify($model-
            >fileImport->tempName );
            $objReader =
            \PHPExcel_IOFactory::createReader($inputFileType);
            $objPHPExcel = $objReader->load($model->fileImport-
            >tempName);
            $sheetData = $objPHPExcel->getActiveSheet()-
            >toArray(null,true,true,true);
            $baseRow = 2;
            while(!empty($sheetData[$baseRow]['A'])) {
                $model = new \app\models\Unit();
                $model->nama = (string)$sheetData[$baseRow]['B'];
                $model->save();
                $baseRow++;
            }
        }
        return $this->render('import',[
            'model' => $model,
        ]);
    }
}
```



# Impor & Ekspor Data

---

## Ekspor Data

- Ekspor data yang dimaksud di sini adalah menggenerate data dari basis data menjadi format dokumen misalnya *spreadsheet*, PDF, Word, dsb. Penerapan fitur ini dalam dunia nyata memiliki beragam jenis, antara lain: cetak data pendukung, pelaporan, cetak bukti pembayaran, dsb.
- Pada panduan ini kita akan menggunakan *extension* yii2-export dimana *extension* ini mendukung banyak format dokumen yaitu pdf, docx, doc, odt, odf, xls, xlsx, ppt, pptx. Ada tiga teknologi dibalik *extension* ini sehingga mendukung banyak format *file* yaitu TinyButStrong, PHPExcel, dan MPDF.



# Impor & Ekspor Data

## Ekspor Data : Ke Excel

- Dengan menggunakan yii2-export, ada dua cara expor data ke excel yaitu melalui TinyButStrong, dan PHPExcel
- Misal kita ingin menggenerate data Unit, maka kita harus siapkan dulu template excelnya

TinyButStrong

	A	B
1	<b>NO</b>	<b>NAMA</b>
2	[data.id;noerr;block=tbs:row]	[data.nama;noerr]
3		
4		
5		

PHPExcel

	A	B
1	<b>NO</b>	<b>NAMA</b>
2		
3		
4		
5		

- data merupakan *variabel* yang nanti akan dikirimkan dari PHP.
- data.no, data.nama adalah properti yang dimiliki variabel data (yang sebagiannya merupakan atribut dari tabel unit)
- noerr untuk mencegah tampilnya error yang mungkin terjadi
- tbs:row menunjukkan baris baru



# Impor & Ekspor Data

## Ekspor Data : Ke Excel

CONTROLLER : TINYBUTSTRONG

```
public function actionExportExcel()
{
    $searchModel = new UnitSearch();
    $dataProvider = $searchModel->search(Yii::$app->request-
    >queryParams);
    // Initialize the TBS instance
    $OpenTBS = new \hscstudio\export\OpenTBS;
    // set template
    $template = Yii::getAlias('@app/views/unit').'_export.xlsx';
    $OpenTBS->LoadTemplate($template);

    $data = [];
    $no=1;
    foreach($dataProvider->getModels() as $unit){
        $data[] = [
            'no'=>$no++,
            'nama'=>$unit->nama,
```



# Impor & Ekspor Data

## Ekspor Data : Ke Excel

### CONTROLLER : PHPEXCEL

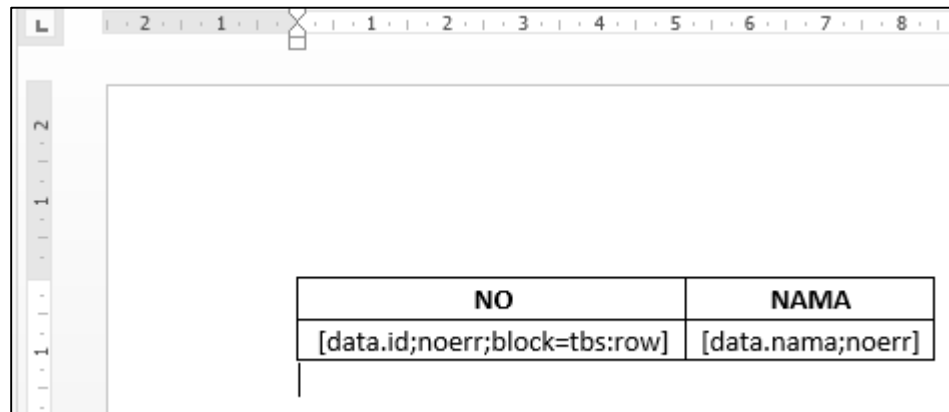
```
public function actionExportExcel2()
{
    $searchModel = new UnitSearch();
    $dataProvider = $searchModel->search(Yii::$app->request-
>queryParams);
    $objReader =
\PHPExcel_IOFactory::createReader('Excel2007');
    // set template
    $template = Yii::getAlias('@app/views/unit').'_export2.xlsx';
    $objPHPExcel = $objReader->load($template);
    $activeSheet = $objPHPExcel->getActiveSheet();
    // set orientasi & ukuran kertas
    $activeSheet->getPageSetup()
        -
>setOrientation(\PHPExcel_Worksheet_PageSetup::ORIENTATIO
N_LANDSCAPE)
        -
>setPaperSize(\PHPExcel_Worksheet_PageSetup::PAPERSIZE_FO
LIO);
    $baseRow=2;
    foreach($dataProvider->getModels() as $category){
        $activeSheet->setCellValue('A'.$baseRow, $baseRow-1)
            ->setCellValue('B'.$baseRow, $unit->nama)
        $baseRow++;
    }
    header('Content-Type: application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet');
    header('Content-Disposition:
attachment;filename="_export2.xlsx");
    header('Cache-Control: max-age=0');
    $objWriter =
\PHPExcel_IOFactory::createWriter($objPHPExcel, "Excel2007");
    $objWriter->save('php://output');
    exit;
}
```



# Impor & Ekspor Data

## Ekspor Data : Ke Word

- Misal kita ingin menggenerate data Unit, maka kita harus siapkan dulu template dokumen Ms Wordnya.
- Karena library yang digunakan adalah TinyButStrong maka aturan templatanya sama dengan bagian sebelumnya tapi menggunakan Ms Word



The screenshot shows a TinyButStrong template editor window. It features a horizontal ruler at the top with markings from 1 to 8. On the left side, there is a vertical ruler with markings from 1 to 2. The main editing area contains a table with two columns and two rows. The first row contains the headers 'NO' and 'NAMA'. The second row contains the data placeholders '[data.id;noerr;block=tbs:row]' and '[data.nama;noerr]'. The table is positioned in the lower right quadrant of the editor window.

NO	NAMA
[data.id;noerr;block=tbs:row]	[data.nama;noerr]



# Impor & Ekspor Data

## Ekspor Data : Ke Word

CONTROLLER : TINYBUTSTRONG

```
public function actionExportWord()
{
    $searchModel = new UnitSearch();
    $dataProvider = $searchModel->search(Yii::$app->request-
>queryParams);
    // Initialize the TBS instance
    $OpenTBS = new \hscstudio\export\OpenTBS;
    // set template
    $template = Yii::getAlias('@app/views/unit').'_export.docx';
    $OpenTBS->LoadTemplate($template);

    $data = [];
    $no=1;
    foreach($dataProvider->getModels() as $unit){
        $data[] = [
            'no'=>$no++,
            'nama'=>$unit->nama,
```





# Impor & Ekspor Data

---

## Ekspor Data : Ke PDF

- Library yang digunakan adalah MPDF. Fungsinya menggenerate dokumen HTML ke PDF
- Misal kita akan menggenerate data Unit, maka kita perlu siapkan tempalatenya namun dalam bentuk kode PHP & HTML



# Impor & Ekspor Data

## Ekspor Data : Ke PDF

### VIEW : TEMPLATE

```
<!DOCTYPE html>
<html>
<head>
  <title>Data Unit</title>
  <style type="text/css">
    .page{padding:2cm;}
    table{border-spacing:0;border-collapse:
collapse;width:100%;}
    table td, table th{border: 1px solid #ccc;}
  </style>
</head>
<body>
  <div class="page">
    <table border="0">
      <tr>
        <th>No</th>
        <th>Nama</th>
```

```
</tr>
<?php
$no = 1;
foreach($dataProvider->getModels() as $unit){
  ?>
  <tr>
    <td><?= $no++ ?></td>
    <td><?= $unit->nama ?></td>
  </tr>
  <?php
  }
  ?>
</table>
</div>
</body>
</html>
```



# Impor & Ekspor Data

## Ekspor Data : Ke PDF

### CONTROLLER

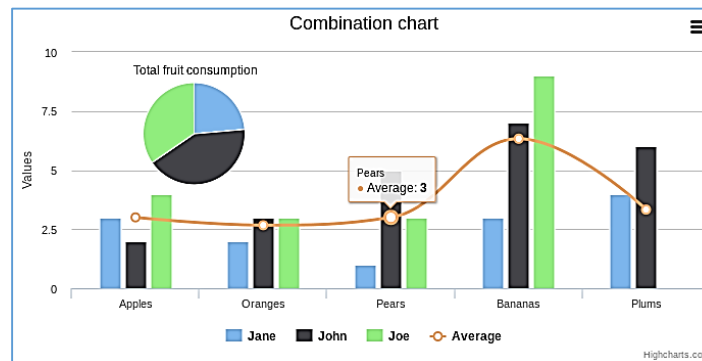
```
public function actionExportPdf()
{
    $searchModel = new UnitSearch();
    $dataProvider = $searchModel->search(Yii::$app->request-
>queryParams);
    $html = $this-
>renderPartial('_export',['dataProvider'=>$dataProvider]);
    $mpdf=new \mPDF('c','A4','','', 0 , 0 , 0 , 0 , 0 , 0);
    $mpdf->SetDisplayMode('fullpage');
    // 1 or 0 - whether to indent the first level of a list
    $mpdf->list_indent_first_level = 0;
    $mpdf->WriteHTML($html);
    $mpdf->Output();
    exit;
}
```



# Visualisasi Data

## Gambaran Umum

- 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.





# Visualisasi Data

---

## Instalasi

- 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 --prefer-dist miloschuman/yii2-highcharts-widget "*"
```



# Visualisasi Data

## Penggunaan : Statis

### VIEW

```
use miloschuman\highcharts\Highcharts;
echo Highcharts::widget([
    'options' => [
        'chart' => ['type' => 'column'],
        'title' => ['text' => 'PHP Framework Survey Result'],
        'xAxis' => [
            'categories' => ['2014', '2015']
        ],
        'yAxis' => [
            'title' => ['text' => 'Usage (thousand)']
        ],
        'series' => [
            ['name' => 'Code Igniter', 'data' => [90,85]],
            ['name' => 'Laravel', 'data' => [75,95]],
            ['name' => 'Symfony', 'data' => [40,70]],
            ['name' => 'Yii', 'data' => [60,80]],
        ],
        'plotOptions' => [
            'column' => [
                'dataLabels' => [
                    'enabled' => true,
                ],
            ],
        ],
    ],
]);
```



# Visualisasi Data

## Penggunaan : Dinamis

### CONTROLLER

```
public function actionChart ()
{
    $db = \Yii::$app->db;
    $datas = $db->createCommand('
        SELECT unit.name as nama, count(employee.user_id) as jumlah FROM unit, employee
        WHERE unit.id = employee.unit_id
        GROUP BY unit.id')
        ->queryAll();
    foreach($datas as $data){
        $categories[] = $data["nama"];
        $series[] = [
            'name' => $data["nama"],
            'data' => [(int)$data["jumlah"]],
        ];
    }
    return $this->render('chart',["categories"=>$categories,"series"=>$series]);
}
```



# Visualisasi Data

## Penggunaan : Dinamis

### VIEW

```
<?php
use miloschuman\highcharts\Highcharts;
echo Highcharts::widget([
    'options' => [
        'chart' => ['type' => 'column'],
        'title' => ['text' => 'Jumlah Pegawai Per Unit'],
        'xAxis' => [
            'categories' => $categories,
        ],
        'yAxis' => [
            'title' => ['text' => 'Jumlah Pegawai']
        ],
        'series' => $series,
        'plotOptions' => [
            'column' => [
                'dataLabels' => [
                    'enabled' => true,
```





# Otorisasi

---

## Instalasi

- 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*,



# Otorisasi

---

## ACF

- 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()`



# Otorisasi

## Contoh ACF

### CONTROLLER

```
public function behaviors()
{
    return [
        'access' => [
            'class' => AccessControl::className(),
            'only' => ['login', 'error', 'logout', 'index'], // opsional
            'rules' => [
                [
                    'actions' => ['login', 'error'],
                    'allow' => true,
                ],
                [
                    'actions' => ['logout', 'index'],
                    'allow' => true,
                    'roles' => ['@'],
                ],
            ],
        ],
        'verbs' => [
            'class' => VerbFilter::className(),
            'actions' => [
                'logout' => ['post'],
            ],
        ],
    ];
}
```

use yii\filters\VerbFilter; use yii\filters\AccessControl;



# Otorisasi

---

## Penjelasan Contoh ACF

- *Only* berisi daftar fungsi *action* yang dikenai *filter access* ini, yaitu login, error, logout dan index. Jika parameter ini tidak didefinisikan artinya *filter access* akan diimplementasikan pada semua fungsi *action*.
- *Rules* berisi daftar aturan *filter action* tentang boleh atau tidaknya *pengguna* tertentu mengakses suatu *action*.
  - *Actions* berisi daftar fungsi *action* yang dikenai suatu aturan
  - *Allow* bernilai *boolean* yang menunjukkan boleh tidaknya suatu *access*
  - *Roles* berkaitan dengan siapa yang dikenai aturan tersebut. Tanda ? menunjukkan *pengguna* yang belum login atau *guest*, tanda @ menunjukkan *pengguna* yang sudah login / *authenticated user*. Di samping itu kita bisa menggunakan teks lain (nama *role: admin, staff*) yang akan men-trigger pemanggilan fungsi `yii\web\User::can()`, dimana masuk ke ranah RBAC.
- *Rules* pertama bermakna *action login dan eror* diperbolehkan untuk diakses oleh semua jenis *pengguna baik yang sudah login maupun belum*. Sedangkan *rules* kedua artinya fungsi *logout & index* hanya diperbolehkan untuk diakses oleh *pengguna* yang sudah login.
- Dari contoh di atas kita bisa menarik kesimpulan sementara bahwa *pengguna* yang telah login atau *authenticated user* tidak lantas bisa melakukan semua *action*. Kita bisa menentukan mana saja *action* yang boleh dilakukan oleh *pengguna* dan mana yang tidak.



# Otorisasi

---

## 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]]`.

Forbidden (#403)

You are not allowed to perform this action.



# Otorisasi

---

## Multi Level User menggunakan ACF

- Untuk kasus yang sederhana misalnya kita mempunyai tiga level *user* yaitu *guest*, *employee* dan *admin*, maka kita bisa memaksimalkan ACF yaitu dengan menggunakan parameter `matchCallback` pada *rules*
- Misalnya, untuk membedakan level user, kita menggunakan field `role` (varchar 25) yang isinya "employee" atau "admin"

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



# Finishing Proyek

---

- Setting Pretty URL
- Setting Aplikasi Sebagai Production
- Setting @app/web Sebagai Mount Point di Apache



# Finishing Proyek

---

## Setting Pretty URL

- Aktifkan Pretty URL di @app/config/web.php bagian components

@app/config/web.php

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

- Tambahkan .htaccess di @app/web/

@app/web/.htaccess

```
RewriteEngine on  
# RewriteBase /  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteRule . index.php
```





# Finishing Proyek

---

## Setting Aplikasi Sebagai Production

- Matikan mode debug atau ubah nilainya dari true menjadi false, kemudian matikan const environment atau ubah nilainya dari dev ke prod

@app/config/web.php

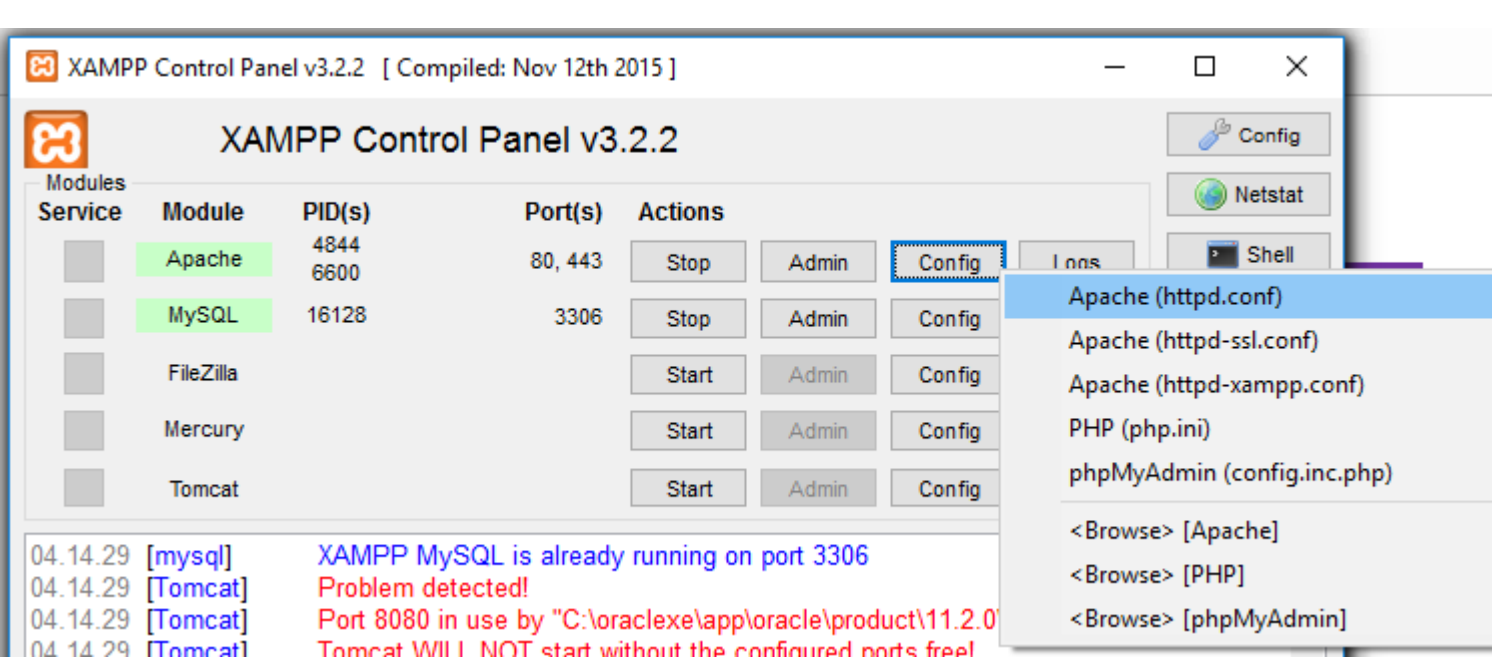
```
// comment out the following two lines when deployed to production  
// defined('YII_DEBUG') or define('YII_DEBUG', true);  
// defined('YII_ENV') or define('YII_ENV', 'dev');
```



# Finishing Proyek

## Setting @app/web Sebagai Mount Point di Apache

- Ubah document root di apache.conf dari htdocs ke @app/web, sehingga kita bisa mengakses aplikasi kita `http://localhost`



apache.conf

```
DocumentRoot "C:/xampp/htdocs/simpeg/web"  
<Directory "C:/xampp/htdocs/simpeg/web">
```



# Terima Kasih

---

Contact us:  
[hafidmukhlasin@gmail.com](mailto:hafidmukhlasin@gmail.com)