



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

**Mengetahui dan memahami Dasar-Dasar Web Framework**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>01</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

## 1.1 Pengenalan Codeigniter (CI)

Menurut Hakim (2010:8) “Codeigniter adalah sebuah framework PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibanding jika menulis kode program dari awal”. Ada banyak library dan helper yang berguna di dalamnya dan tentunya akan mempermudah proses pengembangan perangkat lunak.



**Gambar 0.1 Logo CodeIgniter**

Sumber: <https://codeigniter.com/> tanggal 22 Februari 2018

CI menggunakan konsep MVC (Model – View – Controller) yang memungkinkan pemisahan layer application logic dan presentation. MVC merupakan konsep yang memisahkan komponen utama menjadi tiga komponen yaitu Model, View dan Controller.

- a. Model merupakan bagian penanganan yang berhubungan dengan manipulasi database termasuk melakukan proses CRUD (Create Read Update Delete) yang dengan bantuan controller yang merujuk ke model yang sesuai dan function yang sesuai maka sebuah proses yang berkaitan dengan database sudah bisa dibuat.
- b. Views merupakan bagian yang menangani masalah user interface. Tampilan dari userinterface ini dipisahkan dari controller dan model untuk memudahkan pengembang dalam membuat disain tampilan. Anda juga dapat membuat folder di dalam folder views untuk mempermudah pengelompokan desain antarmuka yang anda buat.
- c. Controller berfungsi untuk menyimpan file-file controller yang digunakan untuk mengatur alur proses aliran data atau fungsi yang dibuat dan merupakan kumpulan instruksi yang menghubungkan antara View ke Model. Controller merupakan salah satu bagian yang sangat berperan dalam website berbasiskan MVC. Tanpa controller, view atau model tidak akan dapat berjalan dengan semestinya. Pada controller terdapat satu

class yang disesuaikan dengan nama file yang dibuat dan beberapa fungsi yang disesuaikan dengan kebutuhan yang ada.

Keuntungan yang didapat ketika membuat aplikasi dengan menggunakan framework yaitu sebagai berikut:

a. Menghemat waktu pengembangan

Dengan menggunakan struktur dan library yang telah disediakan oleh framework maka kita tidak perlu lagi memikirkan hal tersebut, hanya tinggal fokus pada bisnis proses yang akan kita kembangkan.

b. *Re-use of Code*

Dengan menggunakan *framework* maka struktur pemrograman kita akan memiliki struktur yang baku, sehingga kode program yang sudah dibuat akan dapat digunakan untuk aplikasi yang lain.

c. Bantuan komunitas

Banyak komunitas yang siap membantu dan juga berbagi ilmu sehingga dapat meningkatkan kemampuan pemrograman kita

d. Kumpulan *best practice*

*Framework* merupakan kumpulan *best practice* sehingga kita dapat menguatkan pemrograman kita.

Untuk melakukan pembuatan aplikasi bisnis ini maka kita harus mempersiapkan:

1. *Framework CI*, kita dapat mengunjungi <https://codeigniter.com/> untuk mengunduh versi terbarunya. Untuk diperhatikan ketika menggunakan CI versi 3+ maka PHP yang digunakan haruslah PHP versi 5.3.+.
2. Persiapkan editor untuk melakukan pengkodean, kita dapat menggunakan Notepad++, SublimeText, dll.
3. Persiapkan database yang akan digunakan, sebagai contoh kita dapat menggunakan XAMPP atau WAMP
4. Persiapkan template Jquery EasyUI

### 1.1.1 Struktur Framework CI

Pada saat kali pertama kita menginstal *framework codeigniter* secara default terdapat 3 (tiga) folder utama yaitu *application*, *system*, dan *user\_guide*.

### a. Application

Merupakan tempat di mana seluruh *syntax project* aplikasi kita simpan. Folder *application* memiliki beberapa folder yang dapat membantu kita dalam men-develop sebuah aplikasi. Diantaranya adalah sebagai berikut:

1. Folder *Cache*, merupakan folder yang digunakan untuk menyimpan semua cache yang dibuat oleh cache library
2. Folder *Config*, merupakan folder yang berisi file konfigurasi dari aplikasi yang dibuat mulai dari database, router dan autoload aplikasi.
3. Folder *Controllers*, merupakan folder yang berisi file controller dari aplikasi yang dibuat, digunakan sebagai penghubung antara models dan views.
4. Folder *Core*, merupakan folder yang digunakan untuk memperluas class inti codeigniter.
5. Folder *Helpers*, merupakan folder yang digunakan untuk menyimpan helpers core ci.
6. Folder *Hooks*, merupakan folder untuk menyimpan hooks yang digunakan untuk mengubah alur fungsi dari core codeigniter.
7. Folder *Language*, merupakan folder yang digunakan untuk menyimpan seluruh bahasa-bahasa yang akan digunakan.
8. Folder *Libraries* merupakan folder yang digunakan untuk menyimpan seluruh library.
9. Folder *Logs*, merupakan folder yang digunakan untuk menyimpan seluruh error log apabila error log diaktifkan.
10. Folder *Models*, merupakan folder yang digunakan untuk menyimpan models yang akan mendefinisikan tabel dari database yang dapat kita gunakan oleh controller yang kita buat untuk mengakses database.
11. Folder *Third\_party*, merupakan folder yang digunakan untuk menyimpan fungsi-fungsi tambahan di codeigniter.
12. Folder *Views*, merupakan folder yang digunakan untuk menyimpan seluruh tampilan-tampilan dari aplikasi yang dibuat atau sering disebut dengan user interface.

### b. System

Folder *system* biasanya berisi file, baik itu file aplikasi yang sedang dibuat maupun *core-framework* nya. Diantaranya adalah sebagai berikut :

1. Folder *Core*, merupakan folder yang digunakan untuk memperluas *class* inti *codeigniter*.
  2. Folder *Database*, merupakan folder yang digunakan untuk menyimpan semua *database driver* dan *class* yang akan digunakan.
  3. Folder *Fonts*, merupakan folder yang digunakan untuk menyimpan *font* yang digunakan *image manipulation library*.
  4. Folder *Helpers* merupakan folder yang digunakan untuk menyimpan *helper core ci*.
  5. Folder *Language*, merupakan folder yang digunakan untuk menyimpan semua bahasa *core ci*.
  6. Folder *Libraries*, merupakan folder yang digunakan untuk menyimpan semua *library core ci*
- c. *User\_guide*

Folder ini biasanya berisi *userguide/* manual penggunaan ci. berikut folder-folder yang ada didalam folder *user\_guide*

### 1.1.2 Konfigurasi Framework CI

Setelah mengetahui fungsi ketiga *directory* tadi, maka selanjutnya kita mengetahui file-file di *codeigniter* yang dapat gunakan untuk mengonfigurasi project aplikasi. File-file konfigurasi tersebut disimpan secara rapi pada folder *config* tepatnya pada folder *application/ config*. Berikut file-file konfigurasi yang ada di dalam folder tersebut:

1. *config.php*, digunakan untuk membuat pengaturan dasar pada *project* aplikas, seperti *base\_url*, *index\_page*, *cookie*, *proxy* dan lain-lain
2. *constant.php*, digunakan untuk membuat sebuah *constant* baru.
3. *database.php*, digunakan untuk mengatur koneksi database pada *project* aplikasi.
4. *doctypes.php*, digunakan untuk tempat penyimpanan deklarasi dokumen *Doctype*.
5. *foreign\_chars.php*, digunakan untuk tempat penyimpanan karakter-karakter asing
6. *hooks.php*, digunakan untuk mendefine "*hooks*" untuk meng *extends* CI.
7. *memcached.php*, digunakan untuk kita agar bisa memcache database, driver dan lain-lain sehingga lebih efektif.
8. *migration.php*, digunakan untuk kita agar bisa melakukan *database migration*. Secara *default* nya *config* ini dijadikan *false*.

9. *mimes.php*, digunakan untuk menyimpan array yang berisi tipe file untuk fungsi *upload*.
10. *profiler.php* digunakan untuk mengatur *profiler* yang berguna pada saat *debugging*.
11. *routes.php*, digunakan untuk mengatur *default controller* dan *override 404*.
12. *smileys.php*, digunakan untuk menyimpan array yang berisi *smiley (emoticon)* untuk membantu *helperemoticon*.
13. *user\_agents.php*, digunakan untuk menyimpan data *useragent*, yang membantu *classuser* agen untuk mengidentifikasi *browser*, *platform*, robot dan *datamobiledevice*.

Pada konfigurasi sederhana biasanya *developer* atau *programmer* mengatur pengaturan *config*, *autoload*, *database* dan *routes*.

## 1.2 Pengenalan EasyUI

EasyUI merupakan plugin yang memudahkan kita untuk membuat aplikasi berbasis web, sehingga developer tidak perlu membuat kode program dari nol. EasyUI dapat kita gabungkan penggunaannya dengan framework CodeIgniter. Kita dapat menggunakan EasyUI dengan langsung mengunjungi laman <https://www.jeasyui.com/>.



Gambar 0.2 Logo EasyUI

Sumber: <https://www.jeasyui.com/>. Tanggal 22 Pebruari 2018

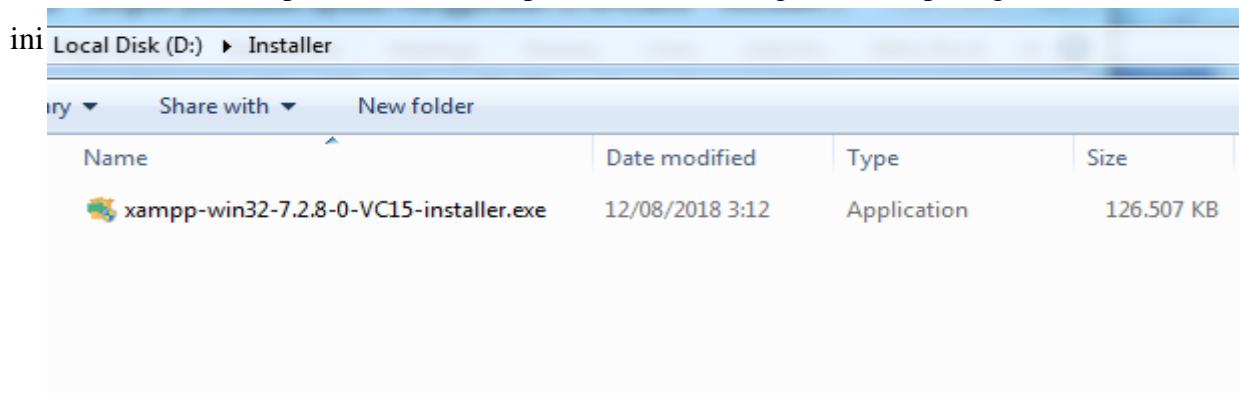
Web Demos	Mobile Demos	Angular Demos	Theme Builder				
Application	Panel	Accordion	Tabs	Layout	DataGridView	DataTable	
PropertyGrid	Tree	TreeGrid	LinkButton	Menu	MenuButton	SplitButton	
SwitchButton	Pagination	ProgressBar	SearchBox	Form	TextBox	PasswordBox	
FileBox	TagBox	Combo	ComboBox	ComboBoxGrid	ComboTree	ComboTreeGrid	
NumberBox	NumberSpinner	Calendar	DateBox	DateTimeBox	TimeSpinner	DateTimeSpinner	
Slider	ValidateBox	Window	Dialog	Messenger	Draggable	Droppable	
Resizable	Tooltip						

Gambar 0.3 Plugin JqueryEasyUI

Sumber: <https://www.jeasyui.com/> tanggal 22 Pebruari 2018

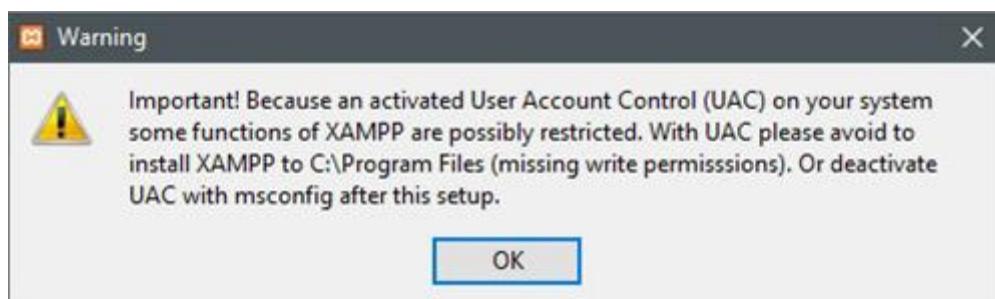
### 1.3 Instalasi XAMPP

Sebelum kita melakukan proses instalasi XAMPP, maka terlebih dahulu kita dapat mengunduh software Bitnami for XAMPP pada laman web <https://bitnami.com/download>. Link tersebut akan secara otomatis mengunduh file installer, tunggu sesaat sampai proses unduh selesai dan siap untuk dilakukan proses instalasi dengan hasil seperti gambar di bawah ini



Gambar 0.4 Gambar Installer xampp

Klik dua kali pada aplikasi xampp yang telah diunduh dan akan tampil *splash screen* bitnami, setelah itu akan ada peringatan *error* seperti pada gambar di bawah ini



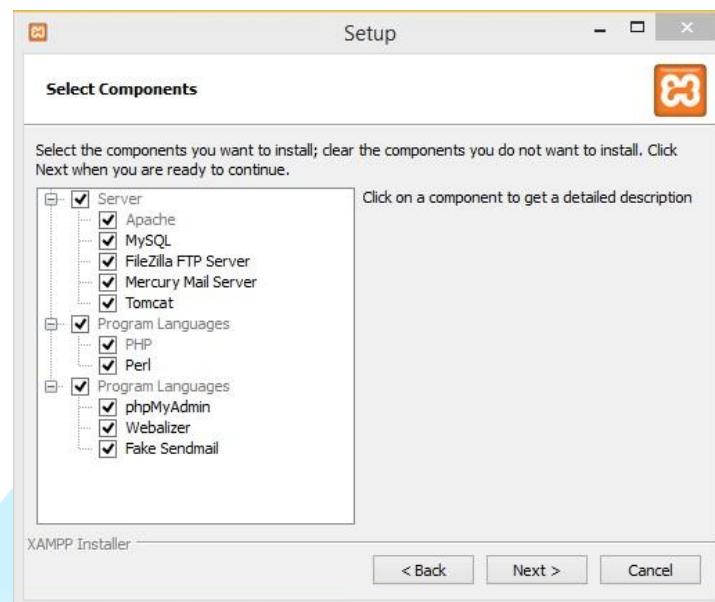
Gambar 0.5 Tampilan warning User Account Control

*Error* tersebut terkait dengan masalah hak akses, klik OK maka akan melakukan proses awal instalasi *Setup – XAMPP* dengan tampilan seperti di bawah ini



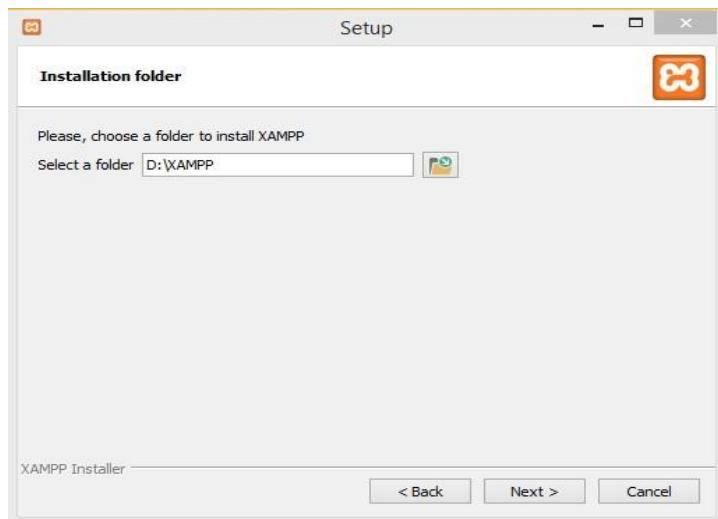
Gambar 0.6 Tampilan welcome installer xampp

Setelah itu maka akan tampil halaman pemilihan item apa saja yang akan kita install di dalam web server xampp. Silahkn disesuaikan dengan kebutuhan kita. Pada kasus ini kita pilih seluruhnya item-item yang akan diinstall dengan tampilan seperti gambar di bawah ini



Gambar 0.7 Tampilan pilih komponen installer

Selanjutnya pilih tombol Next untuk melanjutkan, dan kita akan diarahkan untuk menentukan drive dan folder instalasi xampp. Jika secara default installer aplikasi maka akan diarahkan ke drive C, namun karena permasalahan hak akses seperti pembahasan di atas maka pada penulisan ini kita arahkan pada drive lain. Pada kasus ini kita akan mengarahkan ke drive selain drive C, misal ke drive D dengan folder XAMPP sesuai dengan gambar dibawah ini



Gambar 0.8 Tampilan memilih folder installer

Selanjutnya pilih Next dan akan tampil promosi bitnami menawarkan cara menginstall CMS seperti wordpress dan lain-lain. Pada langkah ini terdapat sebuah checkbox pada bagian learn more about Bitnami for XAMPP dan kita hilangkan ceklis pada checkbox tersebut seperti pada gambar di bawah ini



Gambar 0.9 Tampilan bitnami setup

Selanjutnya pilih Next dan proses instalasi akan berjalan untuk melakukan ekstrak file-file xampp ke dalam hard drive folder yang telah kita tentukan pada pembahasan sebelumnya. Selanjutnya tunggu sesaat proses tersebut hingga selesai



Gambar 0.10 Tampilan progress instalasi



Gambar 0.11 Tampilan proses finishing instalasi

Selanjutnya langkah terakhir dari proses instalasi Xampp adalah akan tampil form Completing the XAMPP Setup Wizard. Lalu terdapat checkbox Do you want to start the Control Panel now?, seperti pada gambar di bawah ini



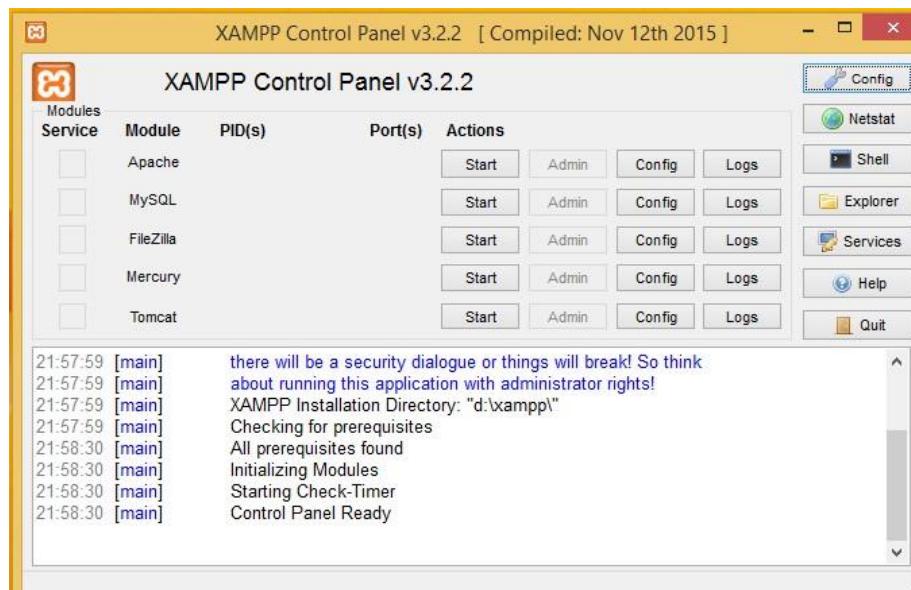
**Gambar 0.12 Tampilan selesai proses instalasi**

biarkan ceklis tersebut, atau jika tidak ada ceklis maka berikan ceklis tersebut maka akan secara otomatis XAMPP Control Panel akan berjalan lalu klik Finish. Sampai proses ini instalasi XAMPP telah selesai. Lalu kita coba cek pada folder yang telah kita siapkan sebelumnya pada proses instalasi. Karena kita memberikan ceklis pada proses akhir, maka Xampp Control panel akan ditampilkan, namun untuk kali pertama dijalankan akan diminta untuk menentukan bahasa yang akan kita gunakan, seperti pada gambar di bawah ini



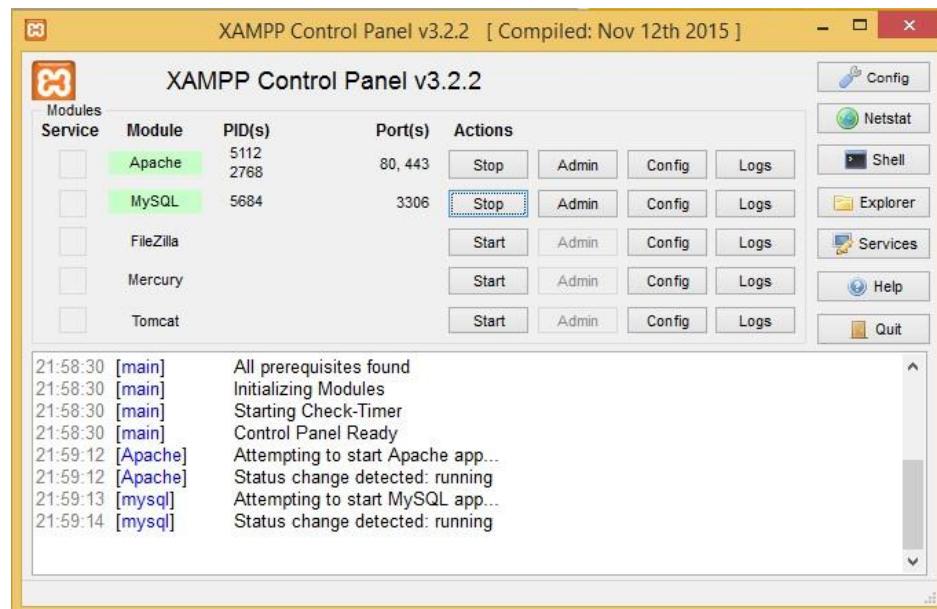
**Gambar 0.13 Tampilan pemilihan bahasa**

Setelah kita pilih OK maka selanjutnya akan tampil Xampp Control Panel seperti pada gambar di bawah ini



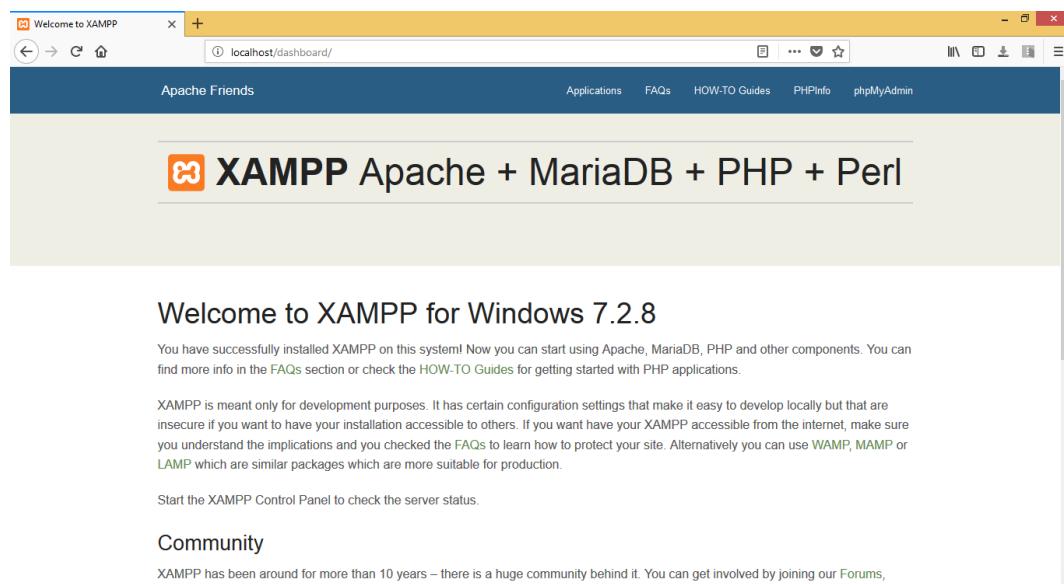
Gambar 0.14 Tampilan control panel xampp

Lalu kita pilih tombol Start pada Apache dan MySQL untuk menjalankan web server dan database kita, dan akan tampil seperti pada gambar di bawah ini



Gambar 0.15 Tampilan running module xampp

Untuk menguji jika semua sudah berjalan dengan baik, mari kita buka browser dan arahkan ke alamat URL <http://localhost> atau <http://127.0.0.1>, jika berhasil maka akan tampil seperti pada gambar di bawah ini



**Gambar 0.16 Tampilan running localhost**

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

**Mengetahui dan memahami  
Konsep OOP dalam Pemrograman  
Web**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>02</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Pada bab ini diharapkan mahasiswa dapat memahami secara detail tentang pembuatan dan penggunaan komponen model, view, controller yang merupakan kunci utama dalam menggunakan framework codeigniter. Pada pengertian codeigniter di atas tadi di jelaskan bahwa codeigniter menggunakan metode MVC. Apa itu MVC? kita juga harus mengetahui apa itu MVC sebelum masuk dan lebih jauh dalam belajar codeigniter.

MVC adalah teknik atau konsep yang memisahkan komponen utama menjadi tiga komponen yaitu model, view dan controller.

## a. Model

Model adalah kelas yang merepresentasikan atau memodelkan tipe data yang akan digunakan oleh aplikasi. Model juga dapat didefinisikan sebagai bagian penanganan yang berhubungan dengan pengolahan atau manipulasi database. seperti misalnya mengambil data dari database, menginput dan pengolahan database lainnya. semua intruksi atau fungsi yang berhubung dengan pengolahan database di letakkan di dalam model. Sebagai contoh, jika ingin membuat aplikasi untuk menghitung luas dan keliling lingkaran, maka dapat memodelkan objek lingkaran sebagai kelas model.

## b. View

View merupakan bagian yang menangani halaman user interface atau halaman yang muncul pada user(pada browser). tampilan dari user interface di kumpulkan pada view untuk memisahkannya dengan controller dan model sehingga memudahkan web designer dalam melakukan pengembangan tampilan halaman website.

## c. Controller

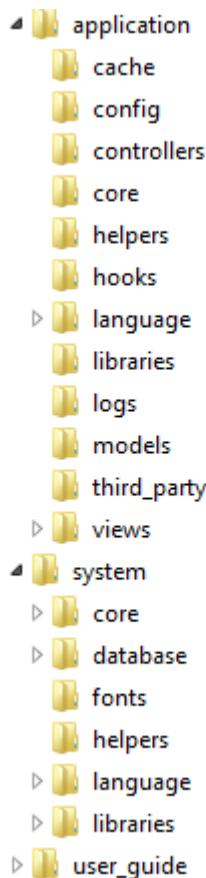
Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) di ambil oleh controller dan kemudian controller pula yang menampilkan nya ke view. Jadi controller lah yang mengolah intruksi.

Dari penjelasan tentang model view dan controller di atas dapat di simpulkan bahwa controller sebagai penghubung view dan model. misalnya pada aplikasi yang menampilkan data dengan menggunakan metode konsep mvc, controller memanggil intruksi pada model yang mengambil data pada database, kemudian controller yang meneruskannya pada view untuk di tampilkan. jadi jelas sudah dan sangat mudah dalam pengembangan aplikasi dengan cara mvc ini karena web designer atau front-end

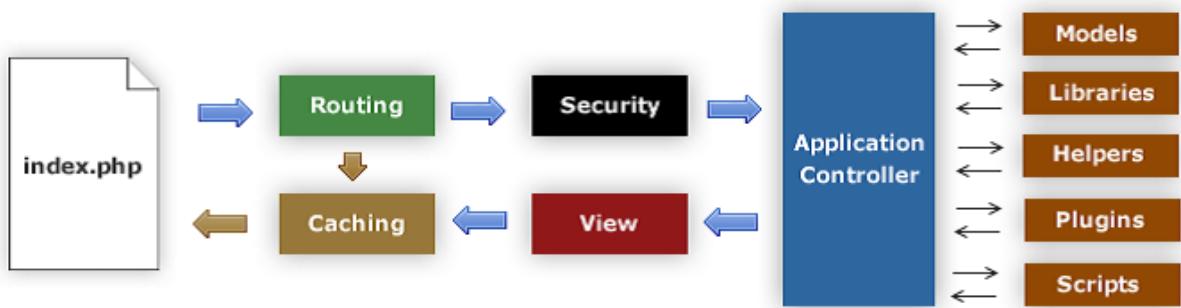
developer tidak perlu lagi berhubungan dengan controller, dia hanya perlu berhubungan dengan view untuk mendesign tampilan aplikasi, karena back-end developer yang menangani bagian controller dan modelnya. Jadi pembagian tugas pun menjadi mudah dan pengembangan aplikasi dapat dilakukan dengan cepat dan terstruktur.

Pada pembahasan ini kita akan coba secara bertahap memulai membuat aplikasi bisnis menggunakan framework CI. Untuk memulai membuat aplikasi menggunakan CodeIgniter dan Jquery EasyUI langkah yang harus kita lakukan adalah sebagai berikut:

1. Buat folder baru dan beri nama PW2 pada folder D:\XAMPP\htdocs. Perlu diperhatikan bahwa **drive D** ini sesuai dengan proses instalasi xampp pada pembahasan di atas.
2. Ekstrak file CI dari hasil download <https://codeigniter.com/download> dan simpan pada *under root web server* kita. Jika kita menggunakan XAMPP, maka secara default root tersebut terletak pada D:\xampp\htdocs\PW2\ dan jika kita menggunakan WAMP maka secara default root tersebut terdapat pada D:\wamp\www\PW2\. Hasil ekstrak akan membentuk struktur file seperti di bawah ini



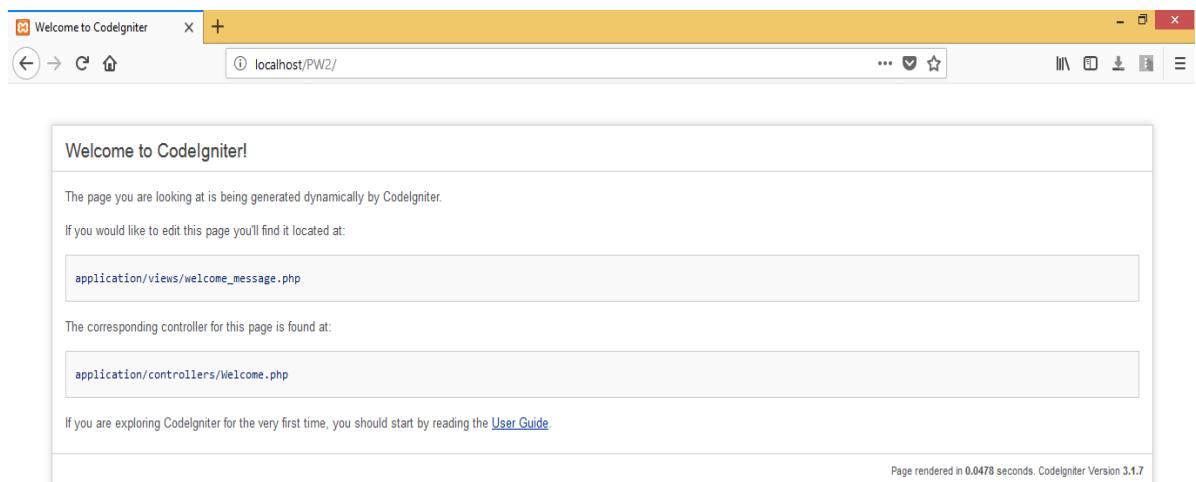
Gambar 0-1 Struktur File CI



**Gambar 0-2 Application Flowchart**

Sumber: Hakim (2010:12) Membangun Web berbasis PHP dengan *framework CodeIgniter*

3. selanjutnya kita jalankan CI yang telah kita download pada browser dengan mengetikkan [http://localhost/PW2/\[nama-file\]/](http://localhost/PW2/[nama-file]/) dan akan tampil seperti gambar di bawah

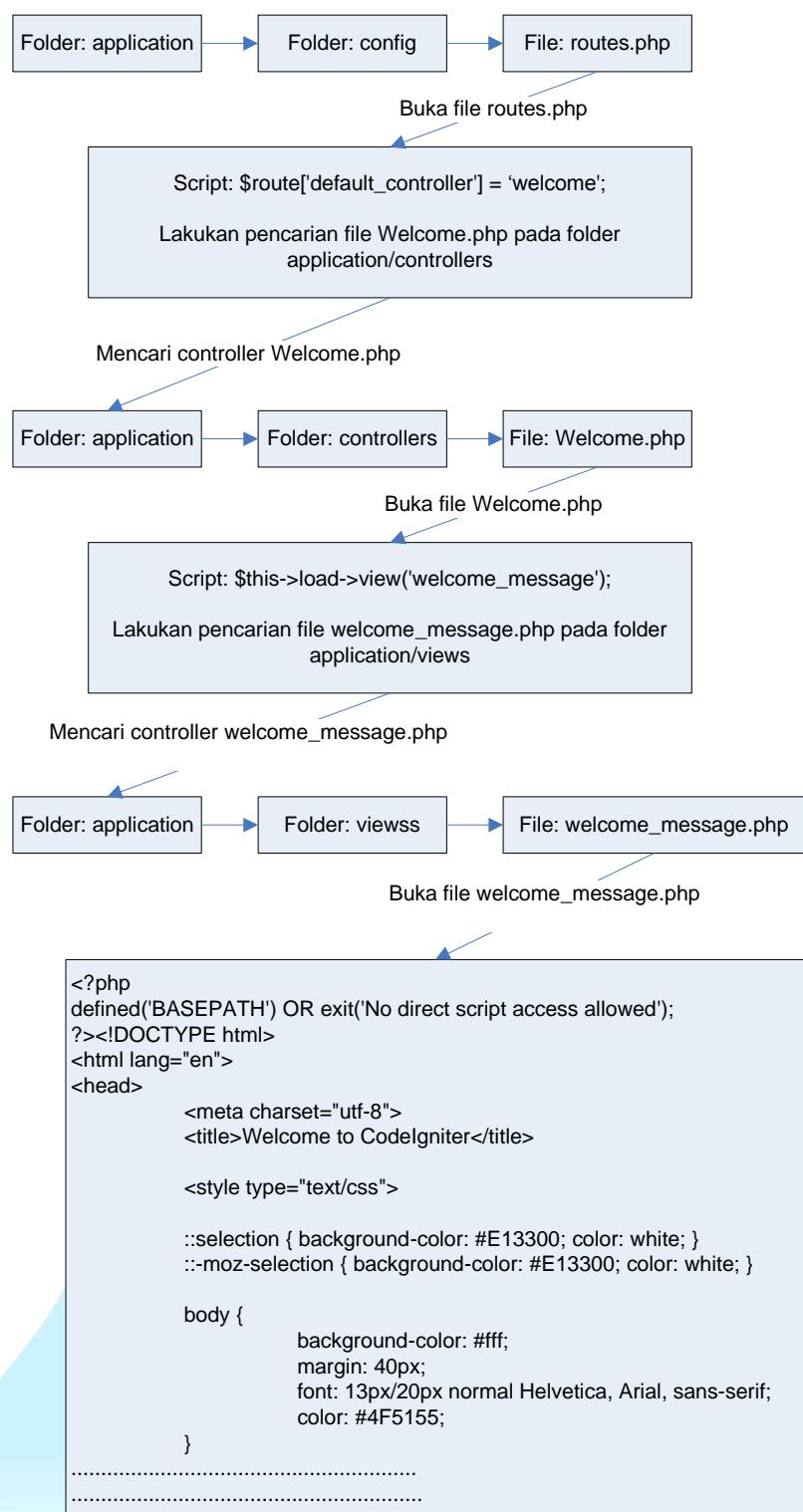


**Gambar 0-3 Tampilan Welcome to CodeIgniter**

Untuk diketahui sebelumnya bahwa proses menampilkan aplikasi pada bahasa program PHP Native akan berbeda dengan PHP Framework, khususnya pada Framework CodeIgniter. Jika dalam PHP Native yang akan ditampilkan adalah file dengan nama index.php atau index.html, namun pada framework CodeIgniter yang akan ditampilkan

adalah sesuai dengan apa yang kita setting pada file routes.php di dalam folder Application/Config/.

Proses untuk menampilkan “Welcome to CodeIgniter!” dapat dilihat dari proses seperti pada gambar di bawah ini

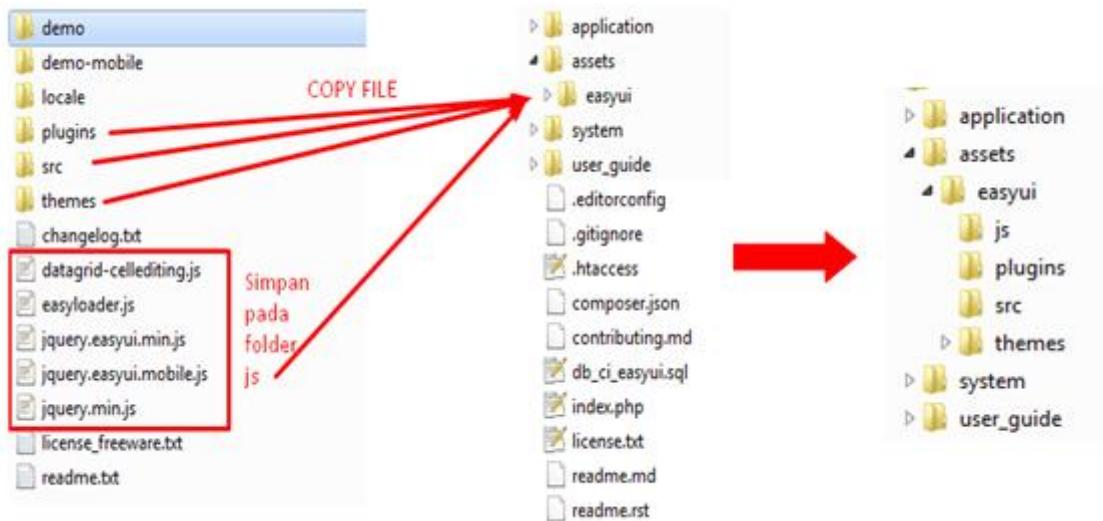


Gambar 0-4 Alur Baca Program pada Framework CI

4. Ubah settingan `$config['base_url']` pada file config yang terletak pada `application\config\config.php` menjadi lokasi server kita, misalkan menjadi <http://localhost/PW2/>, seperti terlihat pada script di bawah ini

```
<?php if(! defined('BASEPATH')) exit('No direct script access allowed');  
/*  
|-----  
| Base Site URL  
|-----  
|  
| URL to your CodeIgniter root. Typically this will be your base URL,  
| WITH a trailing slash:  
|  
|     http://example.com/  
|  
| If this is not set then CodeIgniter will guess the protocol, domain and  
| path to your installation.  
*/  
$config['base_url'] ='http://localhost/PW2/';  
  
/*  
|-----  
| Index File  
|-----  
| Typically this will be your index.php file, unless you've renamed it to  
| something else. If you are using mod_rewrite to remove the page set this  
| variable so that it is blank.  
*/  
$config['index_page']='index.php';
```

5. Siapkan template yang akan kita gunakan untuk membuat aplikasi, pada pembahasan ini kita akan menggunakan template Easyui yang sebelumnya sudah kita download. Simpan folder themes dan js yang terdapat pada JqueryEasyUI di folder assets, seperti terlihat pada gambar



**Gambar 0-5 Tampilan Struktur File**

6. Lalu untuk melakukan setting file aplikasi mana yang akan kita *load* saat aplikasi kali pertama kita jalankan terdapat pada `$route['default_controller'] = "[nama file]";` di lokasi `application\config\routes.php`. [nama file] di sini kita buat dengan nama `layout.php`

```

| -----
| RESERVED ROUTES
| -----
| There area two reserved routes:
|     $route['default_controller'] = 'welcome';
|
| This route indicates which controller class should be loaded if the
| URI contains no data. In the above example, the "welcome" class
| would be loaded.
|
|     $route['404_override'] = 'errors/page_missing';
|
| This route will tell the Router what URI segments to use if those provided
| in the URL cannot be matched to a valid route.
*/

```

```

$route['default_controller']="layout";
$route['404_override']='';

/* End of file routes.php */

```

7. [nama file] tersebut akan dieksekusi pada folder application/controllers/[nama file]
8. Tambahkan CSS file dengan nama file tambah.css pada folder \assets\easyui\themes seperti script di bawah ini

```
#fm{  
    margin:0;  
    padding:10px 30px;  
}  
.ftitle{  
    font-size:14px;  
    font-weight:bold;  
    color:#666;  
    padding:5px 0;  
    margin-bottom:10px;  
    border-bottom:1px solid #ccc;  
}  
.fitem{  
    margin-bottom:5px;  
}  
.fitem label{  
    display:inline-block;  
    width:180px;  
}
```

Pada saat kali pertama kita menginstal *framework codeigniter* secara default terdapat 3 (tiga) folder utama yaitu *application*, *system*, dan *user\_guide*.

a. *Application*

Merupakan tempat di mana seluruh *syntax project* aplikasi kita simpan. Folder *application* memiliki beberapa folder yang dapat membantu kita dalam men-develop sebuah aplikasi. Diantaranya adalah sebagai berikut:

1. Folder *Cache*, merupakan folder yang digunakan untuk menyimpan semua cache yang dibuat oleh cache library
2. Folder *Config*, merupakan folder yang berisi file konfigurasi dari aplikasi yang dibuat mulai dari database, router dan autoload aplikasi.
3. Folder *Controllers*, merupakan folder yang berisi file controller dari aplikasi yang dibuat, digunakan sebagai penghubung antara models dan views.
4. Folder *Core*, merupakan folder yang digunakan untuk memperluas class inti codeigniter.
5. Folder *Helpers*, merupakan folder yang digunakan untuk menyimpan helpers core ci.
6. Folder *Hooks*, merupakan folder untuk menyimpan hooks yang digunakan untuk mengubah alur fungsi dari core codeigniter.
7. Folder *Language*, merupakan folder yang digunakan untuk menyimpan seluruh bahasa-bahasa yang akan digunakan.

8. Folder *Libraries* merupakan folder yang digunakan untuk menyimpan seluruh library.
9. Folder *Logs*, merupakan folder yang digunakan untuk menyimpan seluruh error log apabila error log diaktifkan.
10. Folder *Models*, merupakan folder yang digunakan untuk menyimpan models yang akan mendefinisikan tabel dari database yang dapat kita gunakan oleh controller yang kita buat untuk mengakses database.
11. Folder *Third\_party*, merupakan folder yang digunakan untuk menyimpan fungsi-fungsi tambahan di codeigniter.
12. Folder *Views*, merupakan folder yang digunakan untuk menyimpan seluruh tampilan-tampilan dari aplikasi yang dibuat atau sering disebut dengan user interface.

*b. System*

Folder *system* biasanya berisi file, baik itu file aplikasi yang sedang dibuat maupun *core-framework* nya. Diantaranya adalah sebagai berikut :

1. Folder *Core*, merupakan folder yang digunakan untuk memperluas *class* inti codeigniter.
2. Folder *Database*, merupakan folder yang digunakan untuk menyimpan semua *database driver* dan *class* yang akan digunakan.
3. Folder *Fonts*, merupakan folder yang digunakan untuk menyimpan *font* yang digunakan *image manipulation library*.
4. Folder *Helpers* merupakan folder yang digunakan untuk menyimpan *helper core ci*.
5. Folder *Language*, merupakan folder yang digunakan untuk menyimpan semua bahasa *core ci*.
6. Folder *Libraries*, merupakan folder yang digunakan untuk menyimpan semua *library core ci*

*c. User\_guide*

Folder ini biasanya berisi *userguide/* manual penggunaan ci. berikut folder-folder yang ada didalam folder *user\_guide*

### 1.1.1 Konfigurasi Framework CI

Setelah mengetahui fungsi ketiga *directory* tadi, maka selanjutnya kita mengetahui file-file di *codeigniter* yang dapat gunakan untuk mengonfigurasi project aplikasi. File-file konfigurasi tersebut disimpan secara rapi pada folder *config* tepatnya pada folder *application/config*.

Berikut file-file konfigurasi yang ada di dalam folder tersebut:

1. *config.php*, digunakan untuk membuat pengaturan dasar pada *project* aplikasi, seperti *base\_url*, *index\_page*, *cookie*, *proxy* dan lain-lain

```
<?php if(! defined('BASEPATH')) exit('No direct script access allowed');  
/*  
|-----  
| Base Site URL  
|-----  
  
| URL to your CodeIgniter root. Typically this will be your base URL,  
| WITH a trailing slash:  
  
| http://example.com/  
  
| If this is not set then CodeIgniter will guess the protocol, domain and  
| path to your installation.  
*/  
$config['base_url'] ='http://localhost/PW2/';  
  
/*  
|-----  
| Index File  
|-----  
  
| Typically this will be your index.php file, unless you've renamed it to  
| something else. If you are using mod_rewrite to remove the page set this  
| variable so that it is blank.  
*/  
$config['index_page']='index.php';
```

2. *constant.php*, digunakan untuk membuat sebuah *constant* baru.
3. *database.php*, digunakan untuk mengatur koneksi database pada *project* aplikasi.

```
| the query builder class.  
*/  
$active_group = 'default';  
$query_builder = TRUE;  
  
$db['default'] = array(  
    'dsn' => '',  
    'hostname' => 'localhost',  
    'username' => 'root',  
    'password' => '',  
    'database' => 'fasilkom',  
    'dbdriver' => 'mysqli',  
    'dbprefix' => '',  
    'pconnect' => FALSE,  
    'db_debug' => (ENVIRONMENT !== 'production'),  
    'cache_on' => FALSE,  
    'cachedir' => '',  
    'char_set' => 'utf8',  
    'dbcollat' => 'utf8_general_ci',  
    'swap_pre' => '',  
    'encrypt' => FALSE,  
    'compress' => FALSE,
```

```

'strictOn' => FALSE,
'failover' => array(),
'save_queries' => TRUE
);

```

4. *doctypes.php*, digunakan untuk tempat penyimpanan deklarasi dokumen *Dotype*.
5. *foreign\_chars.php*, digunakan untuk tempat penyimpanan karakter-karakter asing
6. *hooks.php*, digunakan untuk mendefine "*hooks*" untuk meng *extends CI*.
7. *memcached.php*, digunakan untuk kita agar bisa mencache database, driver dan lain-lain sehingga lebih efektif.
8. *migration.php*, digunakan untuk kita agar bisa melakukan *database migration*. Secara *default* nya *config* ini dijadikan *false*.
9. *mimes.php*, digunakan untuk menyimpan array yang berisi tipe file untuk fungsi *upload*.
10. *profiler.php* digunakan untuk mengatur *profiler* yang berguna pada saat *debugging*.
11. *routes.php*, digunakan untuk mengatur *default controller* dan *override 404*.

```

| -----
| RESERVED ROUTES
| -----
| There area two reserved routes:
|     $route['default_controller'] = 'welcome';
|
| This route indicates which controller class should be loaded if the
| URI contains no data. In the above example, the "welcome" class
| would be loaded.
|
|     $route['404_override'] = 'errors/page_missing';
|
| This route will tell the Router what URI segments to use if those provided
| in the URL cannot be matched to a valid route.
*/

```

```

$route['default_controller']="welcome";
$route['404_override']='';

/* End of file routes.php */

```

12. *smileys.php*, digunakan untuk menyimpan array yang berisi *smiley (emoticon)* untuk membantu *helperemoticon*.
13. *user\_agents.php*, digunakan untuk menyimpan data *useragent*, yang membantu *classuser* agen untuk mengidentifikasi *browser*, *platform*, robot dan *datamobiledevice*.

Pada konfigurasi sederhana biasanya *developer* atau *programmer* mengatur pengaturan *config*, *autoload*, *database* dan *routes*.

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

**Mengetahui dan memahami  
Konsep OOP dalam Database Web**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>03</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

## 3.1. Konsep Database

Untuk mengambil data yang telah kita simpan pada database kita harus melakukan pemanggilan dengan menggunakan query. Setelah query itu lakukan, selanjutnya baru kita akan mendapatkan hasil dari query tersebut dalam bentuk object maupun dalam bentuk array.

### Contoh Query:

```
$query=$this->db->query ("SELECT nama,alamat,no_kontak FROM petugas ");
```

Perlu kita ketahui bahwa fungsi query di atas belum dapat menghasilkan data apapun. Keluarannya hanya berupa Object(*True*) atau *False*. Jika keluarannya berupa *False* maka bisa dipastikan query yang kita lakukan gagal. Tetapi jika bernilai *True* maka query yang dilakukan berarti berhasil. Kita dapat mengambil data yang diinginkan dengan contoh seperti di bawah ini

```
$query=$this->db->query ("SELECT nama,alamat,no_kontak FROM petugas");

foreach ($query->result () as $row) {
    echo $row->nama;
    echo $row->alamat;
    echo $row->no_kontak;
}

echo 'Total Result: ' . $query->num_rows();
```

Contoh tersebut di atas merupakan salah satu penggunaan query. Penggunaan `$query->result()` merupakan syntax untuk mendapatkan hasil query dalam bentuk object, sedangkan untuk mendapatkan hasil query dalam bentuk array asosiatif kita dapat menggunakan `$query->result_array()`.

Untuk menampilkan hasil dari query dapat dilakukan dengan menggunakan script `echo $row->nama, echo $row->alamat;, dan $row->no_kontak.` Sedangkan script `echo 'Total Result: ' . $query->num_rows();` digunakan untuk menampilkan berapa banyak record yang terdapat pada table `petugas`.

Seperti yang telah disebutkan di atas, setelah query dilakukan kita perlu memanggil fungsi tertentu untuk mendapatkan hasilnya. Secara umum ada dua jenis tipe data yang dapat dihasilkan yaitu array dan object.

### A. Mengembalikan Hasil Query Sebagai Kumpulan Array

Ketika kita ingin mengambil query alangkah baiknya jika kita cek terlebih dahulu apakah query tersebut memiliki hasil atau tidak. Contoh script dapat dilakukan seperti di bawah ini

```
$sql = "SELECT nama,alamat,no_kontak FROM petugas";
$result = $this->db->query($sql);

if ($result->num_rows() > 0){
    foreach($result->result_array() as $row) {
        echo $row['nama'];
        echo $row['alamat'];
        echo $row['no_kontak'];
    }
}
```

Keterangan script:

Fungsi `$result->result_array()` merupakan method yang digunakan untuk mengambil hasil query dalam bentuk array.

Fungsi `$result->num_rows()` merupakan method yang digunakan untuk mengetahui berapa banyak jumlah record yang didapat.

Fungsi `$row['nama'];` merupakan method yang digunakan untuk mengambil nilai dalam bentuk array

## B. Mengembalikan Hasil Query Sebagai Kumpulan Object

Fungsi yang digunakan hampir sama dengan mengembalikan nilai query sebagai array, cukup memanggil dengan method `$result->result()`. Contoh script dapat dilakukan seperti di bawah ini

```
$sql = "SELECT nama,alamat,no_kontak FROM petugas";
$result = $this->db->query($sql);
if ($result->num_rows() > 0) {
    foreach($result->result() as $row) {
        echo $row->nama;
        echo $row->alamat;
        echo $row->no_kontak;
    }
}
```

Keterangan script:

Fungsi `$result->result()` merupakan method yang digunakan untuk mengambil hasil query dalam bentuk object.

Fungsi `$result->num_rows()` merupakan method yang digunakan untuk mengetahui berapa banyak jumlah record yang didapat.

Fungsi `$row->nama;` `$row->alamat;` `$row->no_kontak;` merupakan method yang digunakan untuk mengambil nilai dalam bentuk object

## C. Mengembalikan Hasil Query Sebagai Row Array

Ketika kita ingin mengambil query alangkah baiknya jika kita cek terlebih dahulu apakah query tersebut memiliki hasil atau tidak. Contoh script seperti di bawah ini

```
$sql = "SELECT nama,alamat,no_kontak FROM petugas";
$result = $this->db->query($sql);

if ($result->num_rows() > 0) {
    $row = $result->row();
    echo $row['nama'];
    echo $row['alamat'];
    echo $row['no_kontak'];
}
```

Keterangan script:

Fungsi `$result->row()` merupakan method yang digunakan untuk mengambil hasil query

Fungsi `$result->num_rows()` merupakan method yang digunakan untuk mengetahui berapa banyak jumlah record yang didapat.

Fungsi `$row['nama']`; merupakan method yang digunakan untuk mengambil nilai dalam bentuk array

#### D. Mengembalikan Hasil Query Sebagai Row Object

Fungsi digunakan hampir sama dengan mengembalikan nilai query sebagai array, cukup memanggil dengan method `($result->result())`. Contoh script seperti di bawah ini

```
$sql = "SELECT nama,alamat,no_kontak FROM petugas";
$result = $this->db->query($sql);

if ($result->num_rows() > 0) {
    $row = $result->row()
    echo $row->nama;
    echo $row->alamat;
    echo $row->no_kontak;
}
```

Keterangan script:

Fungsi `$result->result()` merupakan method yang digunakan untuk mengambil hasil query

Fungsi `$result->num_rows()` merupakan method yang digunakan untuk mengetahui berapa banyak jumlah record yang didapat.

Fungsi `$row->nama;` merupakan method yang digunakan untuk mengambil nilai dalam bentuk object

### 3.1.1 Data Manipulation Language (DML)

*Active record* menyediakan fungsionalitas untuk melakukan *insert data*. Untuk melakukan *insert data* kita memerlukan dua parameter yaitu nama table dan data yang akan kita insert. Data yang akan kita masukkan haruslah memiliki format berupa *array asosiatif* ataupun *object*. Adapun keuntungan menggunakan *active record* untuk *insert data* adalah sederhana untuk digunakan, dan juga query yang kita gunakan sudah bisa dikatakan aman dari kesalahan karenan metode ini dilakukan secara *build-in* yang disertai fungsi untuk penanganan *escaping*.

#### A. Insert Data

Untuk melakukan proses memasukan data ke dalam tabel yang dibuat, maka kita dapat menggunakan script `this->db->insert()` dengan contoh seperti di bawah ini

```
$data = array(  
    'GroupProduct_Name'=>$this->input->post('txtGroupProduk', true)  
)  
  
$query=$this->db->insert('m_groupproduct', $data);
```

Selain dengan cara di atas, kita dapat juga melakukan dengan perintah di bawah ini

```
$namaGroup=$this->input->post('txtGroupProduk');  
  
$query=$this->db->query("INSERT INTO m_groupproduct(GroupProduct_Name)  
VALUES ('$namaGroup')");
```

Contoh di atas akan menghasilkan query yang sama.

#### B. Update Data

Untuk melakukan proses perubahan data ke dalam tabel yang dibuat, maka kita dapat menggunakan script `$this->db->update()` dengan contoh seperti di bawah ini

```
$this->db->where('GroupProduct_auto', $txtId);  
  
$query=$this->db->update('m_groupproduct', array(  
    'GroupProduct_Name'=>$this->input->post('txtGroupProduk', true)  
) );
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$data=array('GroupProduct_Name'=>$this->input->post('txtGroupProduk',true));
$where="GroupProduct_auto = '$txtId'";
$str=$this->db->update('m_groupproduct',$data,$where);
```

atau dapat juga dilakukan dengan perintah di bawah ini

```
$namaGroup=$this->input->post('txtGroupProduk');
$query=$this->db->query("UPDATE m_groupproduct
    SET GroupProduct_Name='$namaGroup'
    WHERE GroupProduct_auto='$txtId'");
```

### C. Delete Data

Untuk melakukan proses penghapusan data dari tabel yang dibuat, maka kita dapat menggunakan script `$this->db->delete()` dengan contoh seperti di bawah ini

```
$query=$this->db->delete('m_groupproduct',array('GroupProduct_auto'=>$id));
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$this->db->where('GroupProduct_auto',$id);
$this->db->delete('m_groupproduct');
```

atau dapat juga dilakukan dengan perintah di bawah ini

```
$query=$this->db->query("DELETE from m_groupproduct
    WHERE GroupProduct_auto='$id'");
```

## D. Select Data

Untuk melakukan proses pemilihan data dari tabel yang dibuat, maka kita dapat menggunakan script `$this->db->get()` dengan contoh seperti di bawah ini

### 1. SELECT secara umum

```
$groupproduct=$this->db->get('m_groupproduct');
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query('SELECT * FROM m_groupproduct');
```

### 2. SELECT dengan LIMIT

```
$groupproduct=$this->db->get('m_groupproduct',3,0);
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query('SELECT * FROM m_groupproduct limit 0,3');
```

### 3. SELECT dengan WHERE

```
$this->db->where('GroupProduct_auto',34);
$groupproduct=$this->db->get('m_groupproduct');
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query('SELECT * FROM m_groupproduct WHERE
GroupProduct_auto="34"');
```

#### 4. SELECT dengan FIELD tertentu

```
$this->db->select ('GroupProduct_auto, GroupProduct_Name') ;  
$groupproduct=$this->db->get ('m_groupproduct') ;
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query ('SELECT GroupProduct_auto, GroupProduct_Name  
FROM m_groupproduct') ;
```

#### 5. SELECT MAX

```
$this->db->select_max ('GroupProduct_auto') ;  
$groupproduct=$this->db->get ('m_groupproduct') ;
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query ('SELECT MAX(GroupProduct_auto) as  
GroupProduct_autoFROM m_groupproduct') ;
```

#### 6. SELECT MIN

```
$this->db->select_min ('GroupProduct_auto') ;  
$groupproduct=$this->db->get ('m_groupproduct') ;
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query ('SELECT MIN(GroupProduct_auto) as  
GroupProduct_autoFROM m_groupproduct') ;
```

#### 7. SELECT AVG

```
$this->db->select_avg ('GroupProduct_auto') ;  
$groupproduct=$this->db->get ('m_groupproduct') ;
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query('SELECT AVG(GroupProduct_auto) as  
GroupProduct_autoFROM m_groupproduct');
```

## 8. SELECT SUM

```
$this->db->select_sum('GroupProduct_auto');  
$groupproduct=$this->db->get('m_groupproduct');
```

Script di atas dapat juga dilakukan dengan perintah di bawah ini

```
$groupproduct=$this->db->query('SELECT SUM(GroupProduct_auto) as  
GroupProduct_autoFROM m_groupproduct');
```

# Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

**Mengetahui dan memahami  
Konsep MVC: Controller**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>04</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

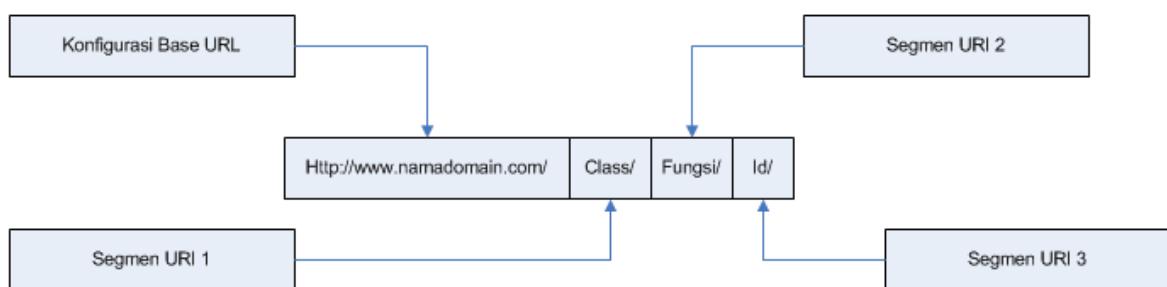
### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Controller merupakan kumpulan intruksi aksi yang menghubungkan model dan view, jadi user tidak akan berhubungan dengan model secara langsung, intinya data yang tersimpan di database (model) di ambil oleh controller dan kemudian controller pula yang menampilkan nya ke view. Jadi controller lah yang mengolah intruksi.

Segmen-segmen pada URL di CodeIgniter mencerminkan Controller yang akan dipanggil. Contoh <http://www.namadomain.com/class/fungsi/id> maka domain tersebut dapat dipecah menjadi bagian-bagian diantaranya:



**Gambar 0.1 Tampilan skema pemanggilan MVC**

Keterangan dari gambar tersebut di atas sebagai berikut:

Keterangan	Fungsi
Konfigurasi Base URL	Merupakan URL yang kita masukan pada konfigurasi base_url yang merupakan URL paling dasar untuk mengakses web kita
Segmen URI 1	Merupakan Class yang terdapat pada controller yang akan dipanggil. Apabila segmen ini kosong maka akan digantikan dengan default controller yang telah kita lakukan proses setting di konfigurasi router.php
Segmen URI 2	Fungsi dari Class Controller yang telah kita panggil pada URI 1. Jika Segemen 2 ini kosong maka yang akan dilakukan proses pemanggilan adalah fungsi index pada class controller tersebut
Segmen URI 3	Biasanya berisikan parameter dari fungsi. Jika fungsi dari controller yang dipanggil mempunyai parameter, maka parameternya harus dimasukan sebagai segmen URI sesuai urutan

## Controllers

*Controller* merupakan jantung dari aplikasi yang menentukan bagaimana permintaan HTTP harus ditangani. Sebuah *controller* hanyalah sebuah nama *class* yang dikaitkan dengan URI.

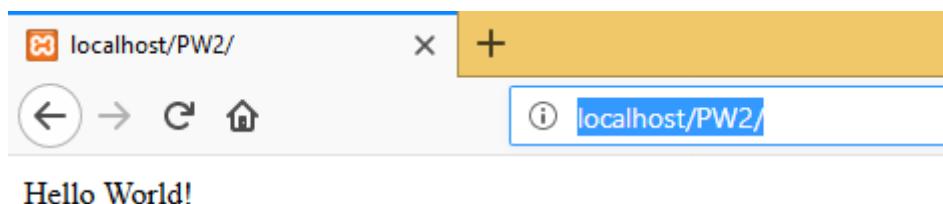
### Contoh 1: Penggunaan hanya controller

Selanjutnya mari kita buat *controller* sederhana yang akan dijalankan pada URL <http://localhost/PW2/index.php/hello>. Langkah yang harus kita lakukan adalah dengan mengetikan *script* di bawah ini pada *text editor* dan simpan pada folder application\controllers\ dengan nama hello.php

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Hello extends CI_Controller {
    public function index()
    {
        echo "Hello World!";
    }
}
```

Setelah selesai lalu kunjungi web yang kita buat dengan mengetikan URL <http://localhost/PW2/index.php/hello> dan akan tampil seperti pada gambar di bawah ini



Gambar 0.2 Tampilan Hello World!

Catatan: Nama kelas harus dimulai dengan huruf besar

Contoh yang benar

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Hello extends CI_Controller {
    public function index()
    {
        echo "Hello World!";
    }
}
```

Contoh yang salah

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class hello extends CI_Controller {
    public function index()
    {
        echo "Hello World!";
    }
}
```

## Contoh 2: Penggunaan controller dan model

Selanjutnya mari kita buat file hello\_model.php pada folder application/models dengan mengetikan *script* di bawah ini pada *text editor*

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Hello_model extends CI_Model{
    //mendefinisikan properti dengan nama $str
    Public $str = "Hello World!";
}

?>
```

Selanjutnya buat file hello.php pada folder application/controllers dengan mengetikan *script* di bawah ini pada *text editor*

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Hello extends CI_Controller {
    public function index( ) {
        //memuat model dengan nama Hello_model
```

```

$this->load->model ('Hello_model');

//menangkap objek dari kelas Hello_model yang telah dimuat ke
variabel $model
$model = $this->Hello_model;

//memanggil properti $str milik objek $model
echo $model->str;

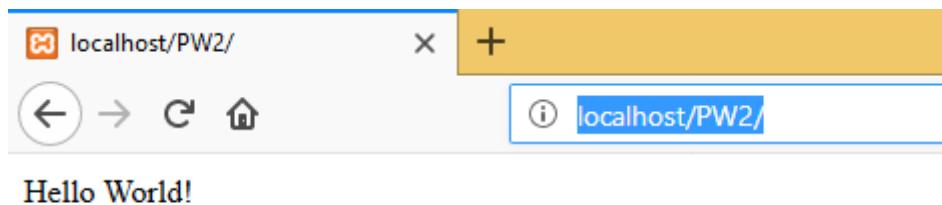
echo "</body>";
echo "</html>";
}

?

?

```

Setelah selesai lalu kunjungi web yang kita buat dengan mengetikan URL <http://localhost/PW2/index.php/hello> dan akan tampil seperti pada gambar di bawah ini



**Gambar 0.3 Tampilan Hello World!**

### Contoh 3: Penggunaan controller dan Views

Selanjutnya mari kita buat file hello\_view.php pada folder application/views dengan mengetikan *script* di bawah ini pada *text editor*

```

<html>
<head><title>Controller dan View</title></head>
<body>
    <h2>Hello World!</h2>
</body>
</html>

```

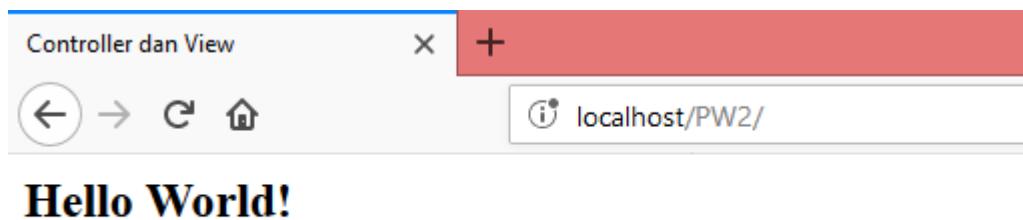
Selanjutnya buat file hello.php pada folder application/controllers dengan mengetikan *script* di bawah ini pada *text editor*

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Hello extends CI_Controller {
    public function index() {
        //memuat model dengan nama Hello2_model
        $this->load->view('Hello_view');
    }
}

?>
```

Setelah selesai lalu kunjungi web yang kita buat dengan mengetikan URL <http://localhost/PW2/index.php/hello> dan akan tampil seperti pada gambar di bawah ini



Gambar 0.4 Tampilan Hello World!

#### Contoh 4: Penggunaan controller, model dan view

Selanjutnya mari kita buat *file* lingkaran\_model.php pada folder application/models dengan mengetikan *script* di bawah ini pada *text editor*

```
<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Lingkaran_model extends CI_Model{
    //mendefinisikan konstanta untuk nilai PI
    const PI = 3.14;

    //atribut model private $jarijari;
    //metode untuk menentukan nilai $jarijari
    public function set_jarijari($r){
        $this->jarijari = $r;
    }

    //metode untuk mengambil nilai $jarijari
    public function get_jarijari(){
        return $this->jarijari;
    }
}
```

```

}

//metode untuk menghitung luas lingkaran
public function hitung_luas( ){
    return self::PI * $this->jarijari * $this->jarijari;
}

//metode untuk menghitung keliling lingkaran
public function hitung_keliling( ){
    return 2 * self::PI * $this->jarijari;
}
}

?>

```

Selanjutnya buat file lingkaran.php pada folder application/controllers dengan mengetikan *script* di bawah ini pada *text editor*

```

<?php if (!defined('BASEPATH')) exit('No direct script access allowed');

class Lingkaran extends CI_Controller {
    public function index() {
        //meload model Lingkaran_model
        $this->load->model('Lingkaran_model');

        //menangkap model yang telah dimuat oleh controller
        $model = $this->Lingkaran_model;

        //menentukan nilai jari jari
        $model->set_jarijari(3);

        //memuat view dan mengirimkan $model ke view
        $this->load->view ('lingkaran_view', array('model' => $model));
    }
}

?>

```

Selanjutnya buat file lingkaran\_view.php pada folder application/views dengan mengetikan *script* di bawah ini pada *text editor*

```

<html>
<head>
    <title> Demo Model, View, Controller </title>
</head>

<body>
<h2> Model Lingkaran </h2>

```

```

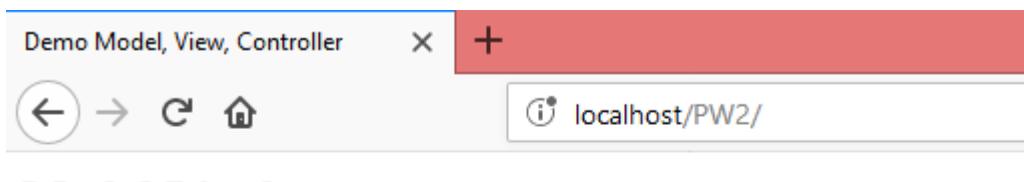
<!-- memanggil metode get_jarijari( ) dari objel $model -->
Nilai Jari Jari : <?php echo $model->get_jarijari( ); ?> <br>

<!-- memanggil metode hitung_luas( ) dari objek $model -->
Luas Lingkaran : <?php echo $model->hitung_luas( ); ?> <br>

<!-- memanggil metode hitung_keliling( ) dari objek $model -->
Keliling Lingkaran : <?php echo $model->hitung_keliling( ); ?>
</body>
</html>

```

Setelah selesai lalu kunjungi web yang kita buat dengan mengetikan URL <http://localhost/PW2/index.php/lingkaran> dan akan tampil seperti pada gambar di bawah ini



## Model Lingkaran

Nilai Jari Jari : 3  
 Luas Lingkaran : 28.26  
 Keliling Lingkaran : 18.84

Gambar 0.5 Tampilan Hello World!

## Fungsi

Pada contoh di atas nama fungsi yang kita buat adalah `index()`. Fungsi “`index()`” akan selalu dimuat secara default jika segmen URI 2 kosong. Cara lain untuk menunjukan pesan kalimat “Hello World!” dapat dilakukan dengan menggunakan URI 2 seperti di bawah ini

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

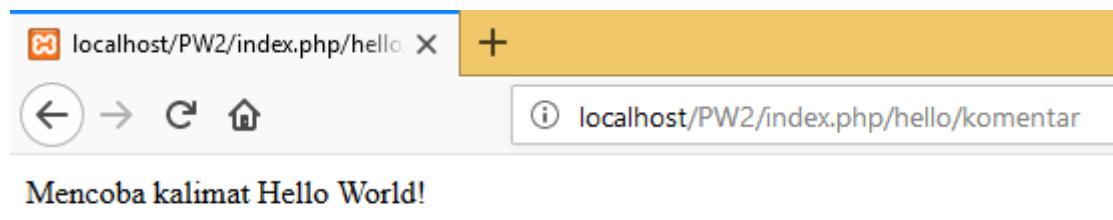
class Hello extends CI_Controller {

    public function index()
    {
        echo "Hello World!";
    }

    public function komentar()
    {
        echo "Mencoba kalimat Hello World!";
    }
}

```

Setelah selesai selanjutnya kita jalankan pada browser dengan alamat URL <http://localhost/PW2/index.php/hello/komentar> dan akan tampil seperti pada gambar di bawah ini



Gambar 0.6 Tampilan contoh fungsi Hello → komentar

### Passing Segmen URI ke Fungsi

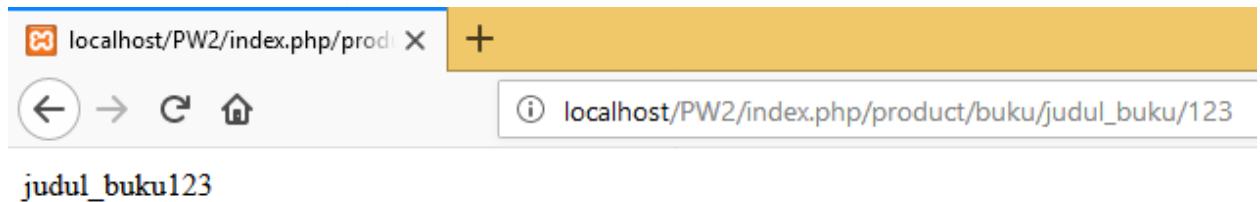
Jika URI kita berisi lebih dari 2 (dua) segmen, maka program akan membaca segmen yang ke-3 sebagai parameter pada fungsi. Sebagai contoh katakanlah kita memiliki URI [http://localhost/PW2/index.php/product/buku/judul\\_buku/123](http://localhost/PW2/index.php/product/buku/judul_buku/123) dengan script yang disimpan pada application/controllers/product.php seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Product extends CI_Controller {

    public function buku($judul, $id)
    {
        echo $judul;
        echo $id;
    }
}
```

Maka akan dihasilkan seperti pada gambar di bawah ini



Gambar 0.7 Tampilan Passing segmen URI ke Fungsi

## Pengorganisasian Controller ke Sub-Folder

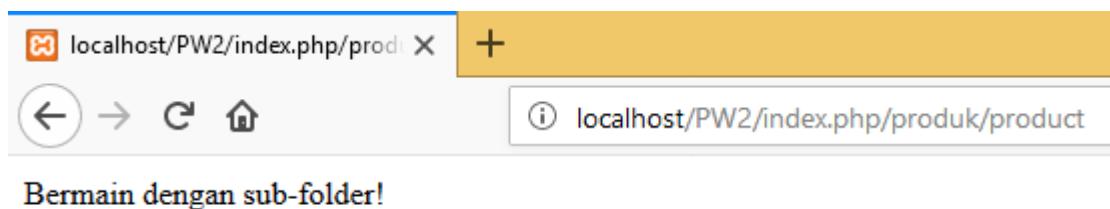
Jika kita akan membangun suatu aplikasi yang besar, maka akan merasa nyaman jika kita mengatur controller pada sub-sub folder. CodeIgniter memungkinkan agar kita melakukan hal tersebut, dengan catatan bahwa URI pada segmen pertama harus menentukan sebuah folder. Sebagai contoh dapat kita lakukan dengan membuat script di bawah ini dan disimpan dengan nama file product.php dalam folder application/controllers/produk/

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Product extends CI_Controller {

    public function index()
    {
        echo "Bermain dengan sub-folder!";
    }
}
```

Untuk memanggil controller tersebut dapat kita lakukan dengan menuliskan URL seperti berikut <http://localhost/PW2/produk/product> dan jika berhasil maka akan tampil seperti gambar di bawah ini



Gambar 0.8 Tampilan controller ke sub-folder

## Class Constructors

Jika kita berminat untuk menggunakan construct dalam salah satu Controller, maka kita harus menempatkan baris kode berikut di dalamnya `parent::__construct();`. Alasan baris ini diperlukan karena konstruktor lokal akan menimpa class induk controller, sehingga kita perlu memanggil secara manual seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Product extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
    }
}
```

Konstruktor akan berguna jika kita akan mengatur beberapa nilai default atau menjalankan default. Konstruktor tidak dapat mengembalikan nilai tetapi dapat melakukan pekerjaan yang menjadi default.

Berikut ini aturan yang dapat digunakan untuk merancang controller.

- 1) Controller boleh mengakses \$\_GET, \$\_POST, dan variabel variabel PHP lainnya yang berkaitan dengan permintaan user.
- 2) Controller boleh membuat objek (instance) dari kelas kelas model dan mengarahkan ke view (jika model bersangkutan menghasilkan data yang perlu ditampilkan ke user)
- 3) Controller seharusnya tidak berisi kode kode yang mengakses data secara langsung dari database. Tugas seperti ini sebaiknya dilakukan oleh model. Dan controller tinggal menggunakan saja.
- 4) Controller seharusnya tidak berisi kode kode HTML yang diperlukan untuk tujuan presentasi. Tugas ini sebaiknya dilakukan oleh view.

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

**Mengetahui dan memahami  
Konsep MVC: View**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>05</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

View digunakan hanya untuk menampilkan halaman web atau fragmen halaman, seperti header, footer, sidebar, dan lain-lain. View tidak pernah ditampilkan secara langsung namun harus melalui controller. Ingat bahwa dalam kerangka MVC, controller berperan sebagai polisi sehingga bertanggung jawab untuk menampilkan view tertentu.

## Membuat View

Buatlah program dengan script di bawah ini untuk menampilkan view bernama hello.php

```
<!DOCTYPE html>
<html>
<head>
    <title>Hello World</title>
</head>
<body>

<p>View pertama saya</p>

</body>
</html>
```

Kemudian simpan file tersebut dalam folder application/views/

## Loading View

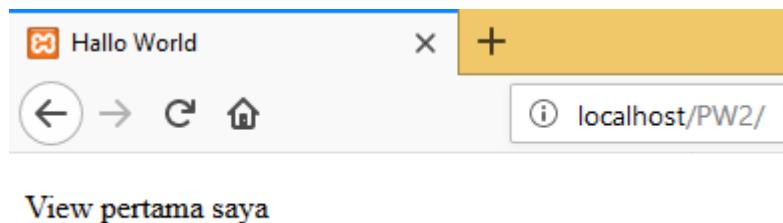
Untuk memuat file view tertentu dapat menggunakan fungsi sebagai berikut `$this->load->view('hello');`

Selanjutnya kita buka file controller yang kita buat sebelumnya pada file hello.php lalu ubah script menjadi seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Hello extends CI_Controller {
    public function index()
    {
        $this->load->view('hello');
    }
}
```

Lalu jalankan URL seperti <http://localhost/PW2/> sehingga akan tampil seperti gambar di bawah ini



**Gambar 0.1 Tampilan view pertama**

### Menambahkan Dynamic Data ke View

Data yang akan ditampilkan ke views melalui controllers dengan cara array dalam parameter dapat dilakukan seperti script di bawah ini. Selanjutnya kita ketikkan script di bawah ini pada text editor dan simpan file tersebut dengan nama file layout.php pada folder application\controllers\

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Layout extends CI_Controller {

    public function index()
    {
        $data = array(
            'title'          => "Hallo World",
            'pos'           => "Posting",
            'message'       => "Pesan saya adalah"
        );
        $this->load->view('layout', $data);
    }
}
```

atau

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Layout extends CI_Controller {

    public function index()
    {
        $data['title'] = "Hallo World";
        $data['pos'] = "Posting";
        $data['message'] = "Pesan saya adalah";

        $this->load->view('layout', $data);
    }
}
```

Selanjutnya kita tambahkan file view dengan nama layout.php pada application/views/ untuk memanggil data dari controller seperti script di bawah ini

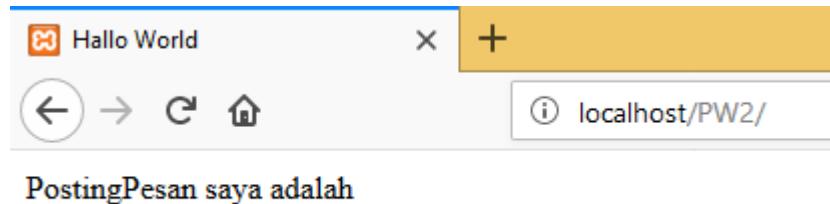
```
<!DOCTYPE html>
<html>
<head>
    <title><?php echo $title; ?></title>
</head>
<body>
<?php
    echo $pos;
    echo $message;
?>

</body>
</html>
```

Sebelum kita jalankan hasil script tersebut, maka kita harus melakukan perubahan data pada application/config/routes.php seperti berikut ini

```
| Examples: my-controller/index      -> my_controller/index
|           my-controller/my-method -> my_controller/my_method
*/
$route['default_controller'] = 'layout';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

Kemudian buka halaman tersebut dengan menjalankan URL seperti berikut <http://localhost/PW2/> dan akan tampil hasil seperti gambar di bawah ini



**Gambar 0.2 Tampilan view mengambil data dari controller**

## Membuat Loops

Kita dapat melakukan array multi-dimensi yang dapat diulang (loops) beberapa baris. Sebagai contoh jika kita menggunakan database, maka untuk menampilkan datanya dapat menggunakan array multi-dimensi. Berikut adalah contoh sederhana untuk array multi-dimensi.

Buat controllers dengan nama looping.php dan simpan pada application\controllers\ dengan script seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class Looping extends CI_Controller {

    public function index()
    {
        $data['title'] = "Hallo World";
        $data['nama_buah'] = array("Semangka", "Mangga", "Jeruk", "Melon");
        $data['message'] = "Pesanan saya adalah";

        $this->load->view('looping', $data);
    }
}
```

Berikutnya kita buat file untuk views dengan nama file looping.php simpan pada application/views/ dengan script seperti di bawah ini

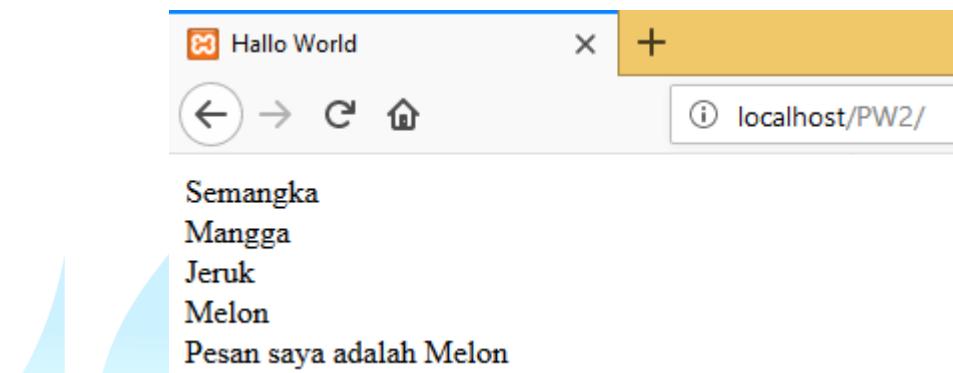
```
<!DOCTYPE html>
<html>
<head>
    <title><?php echo $title; ?></title>
</head>
<body>
<?php
    foreach ($nama_buah as $buah) {
        echo $buah."<br>";
    }
    echo $message." ".$buah;
?>

</body>
</html>
```

Sebelum kita jalankan hasil script tersebut, maka kita harus melakukan perubahan data pada application/config/routes.php seperti berikut ini

```
| Examples: my-controller/index      -> my_controller/index
|           my-controller/my-method -> my_controller/my_method
*/
$route['default_controller'] = 'looping';
$route['404_override'] = '';
$route['translate_uri_dashes'] = FALSE;
```

Kemudian buka halaman tersebut dengan menjalankan URL seperti berikut <http://localhost/PW2/> dan akan tampil hasil seperti gambar di bawah ini



Gambar 0.3 Tampilan data dari controller menggunakan loops array

Berikut ini aturan yang dapat digunakan untuk merancang view.

- 1) View harus berisi kode-kode yang bersifat presentasional, biasanya berupa kode HTML. View juga dapat berisi perintah-perintah PHP sederhana yang masih berkaitan dengan proses pembuatan tampilan, misalnya untuk menampilkan data yang dihasilkan oleh model tertentu.
- 2) View seharusnya tidak berisi kode PHP yang melakukan akses data secara langsung ke database. Pekerjaan semacam ini sebaiknya dilakukan oleh model.
- 3) View seharusnya menghindari kode PHP yang mengakses secara langsung permintaan dari user (tidak berisi \$\_GET maupun \$\_POST). Tugas ini sebaiknya didelegasikan ke controller

### Membuat Layout untuk aplikasi

Implementasi yang akan kita gunakan menggunakan template ace-master yang dapat diunduh pada laman <https://github.com/bopoda/ace> dan template tersebut telah menggunakan bootstrap.

Pembahasan ini kita akan mencoba membuat tampilan utama dalam aplikasi. Tampilan ini berupa bagaimana mempersiapkan lokasi untuk menampilkan header, footer, sidebar, dan juga content dari aplikasi. Rancangan pada layout ini dapat dijabarkan sebagai berikut:

1. Header digunakan untuk menampilkan logo perusahaan atau identitas perusahaan, dan juga untuk menampilkan siapa yang sedang aktif pada aplikasi bisnis tersebut dan juga tautan untuk keluar dari aplikasi (*logout*).
2. Footer digunakan untuk menampilkan siapa pemilik dari aplikasi bisnis yang akan kita kembangkan termasuk kepemilikan dari aplikasi bisnis tersebut.
3. *Sidebar* untuk sisi kiri digunakan untuk mempersiapkan menu-menu apa saja yang ada pada aplikasi bisnis yang akan kita kembangkan, termasuk akan kita persiapkan untuk hak akses dari aplikasi bisnis tersebut.
4. *Sidebar* untuk sisi kanan kita siapkan untuk tampilan form dari menu yang kita pilih

#### 4.1. Controller Layout

Setelah pembahasan di atas, selanjutnya kita akan persiapkan pembuatan program dengan nama file layout.php pada lokasi application/controller/ sesuai dengan `$route['default_controller']` yang sebelumnya sudah kita persiapkan. File layout.php tersebut yang akan dieksekusi kali pertama ketika aplikasi dijalankan. Pastikan nama file

yang kita buat sama dengan nama class yang terdapat di dalam file tersebut, seperti yang terlihat dalam gambar.

```
<?php if(! defined('BASEPATH')) exit('No direct script access allowed');

class layout extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
    }

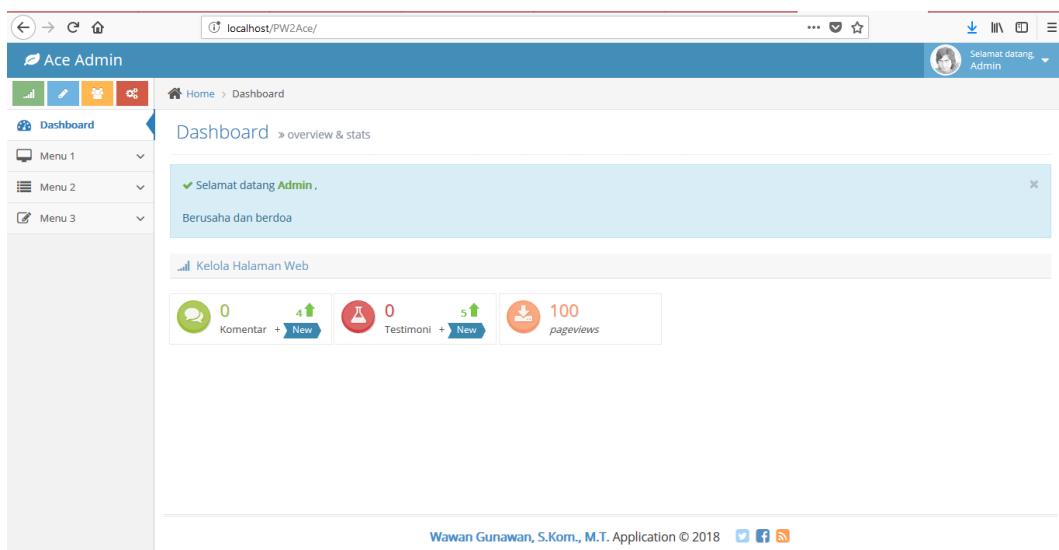
    public function index()
    {
        $this->load->view('layout');
    }
}
```

Keterangan script:

- a. Script `class layout extends CI_Controller {` merupakan nama class layout yang diambil dari nama file layout.php
- b. Pada function `index()` terdapat `$this->load->view('layout');` yang akan melakukan eksekusi file layout.php pada folder application\views yang merupakan tampilan awal dari aplikasi.

## 4.2. Views Layout

Setelah file layout.php pada controller selesai kita buat, selanjutnya kita buat untuk tampilan program pada folder application\views\layout.php dengan tampilan yang akan kita buat seperti pada gambar di bawah ini



Gambar 0.4 Tampilan layout yang diharapkan

Untuk membuat tampilan seperti gambar di atas, maka kita akan membuat script seperti di bawah ini dan simpan pada folder views dengan nama layout.php

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Pemrograman Web 2</title>

    <link rel="stylesheet" type="text/css" href="php echo base_url('assets/css/bootstrap.min.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo base_url('assets/font-awesome/4.5.0/css/font-awesome.min.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo base_url('assets/css/fonts.googleapis.com.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo base_url('assets/css/ace.min.css');?&gt;" class="ace-main-stylesheet" id="main-ace-style" /&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo base_url('assets/css/ace-skins.min.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo base_url('assets/css/ace-rtl.min.css');?&gt;"&gt;
    &lt;script type="text/javascript" src="<?php echo base_url('assets/js/ace-extra.min.js');?&gt;"&gt;&lt;/script&gt;

&lt;/head&gt;
&lt;body class="no-skin"&gt;
    &lt;div id="navbar" class="navbar navbar-default ace-save-state"&gt;
        &lt;div class="navbar-container ace-save-state" id="navbar-container"&gt;
            &lt;button type="button" class="navbar-toggle menu-toggler pull-left" id="menu-toggler" data-target="#sidebar"&gt;
                &lt;span class="sr-only"&gt;Toggle sidebar&lt;/span&gt;
                &lt;span class="icon-bar"&gt;&lt;/span&gt;
                &lt;span class="icon-bar"&gt;&lt;/span&gt;
                &lt;span class="icon-bar"&gt;&lt;/span&gt;
            &lt;/button&gt;

            &lt;div class="navbar-header pull-left"&gt;
                &lt;a href="" class="navbar-brand"&gt;
                    &lt;small&gt;
                        &lt;i class="fa fa-leaf"&gt;&lt;/i&gt;
                        Ace Admin
                    &lt;/small&gt;
                &lt;/a&gt;
            &lt;/div&gt;

            &lt;div class="navbar-buttons navbar-header pull-right" role="navigation"&gt;
                &lt;ul class="nav ace-nav"&gt;
                    &lt;li class="light-blue dropdown-modal"&gt;
                        &lt;a data-toggle="dropdown" href="#" class="dropdown-toggle"&gt;
                            &lt;img class="nav-user-photo" src="<?php echo base_url('assets/images/avatars/user.jpg'); ?&gt;" alt="Jason's Photo" /&gt;
                            &lt;span class="user-info"&gt;
                                &lt;small&gt;Selamat datang,&lt;/small&gt;
                                Admin
                            &lt;/span&gt;
                            &lt;i class="ace-icon fa fa-caret-down"&gt;&lt;/i&gt;
                        &lt;/a&gt;
                    &lt;/li&gt;
                &lt;/ul&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/body&gt;</pre
```

```

        <ul class="user-menu dropdown-menu-right dropdown-menu dropdown-yellow dropdown-caret dropdown-close">
            <li>
                <a href="#">
                    <i class="ace-icon fa fa-cog"></i>
                    Ubah Password
                </a>
            </li>

            <li class="divider"></li>

            <li>
                <a href="#">
                    <i class="ace-icon fa fa-power-off"></i>
                    Logout
                </a>
            </li>
        </ul>
    </li>
</ul>
</div><!-- /.navbar-container -->
</div>

<div class="main-container ace-save-state" id="main-container">
    <script type="text/javascript">
        try{ace.settings.loadState('main-container')}catch(e){}
    </script>

    <div id="sidebar" class="sidebar" responsive ace-save-state">
        <script type="text/javascript">
            try{ace.settings.loadState('sidebar')}catch(e){}
        </script>

        <div class="sidebar-shortcuts" id="sidebar-shortcuts">
            <div class="sidebar-shortcuts-large" id="sidebar-shortcuts-large">
                <button class="btn btn-success">
                    <i class="ace-icon fa fa-signal"></i>
                </button>

                <button class="btn btn-info">
                    <i class="ace-icon fa fa-pencil"></i>
                </button>

                <button class="btn btn-warning">
                    <i class="ace-icon fa fa-users"></i>
                </button>

                <button class="btn btn-danger">
                    <i class="ace-icon fa fa-cogs"></i>
                </button>
            </div>
            <div class="sidebar-shortcuts-mini" id="sidebar-shortcuts-mini">
                <span class="btn btn-success"></span>

                <span class="btn btn-info"></span>

                <span class="btn btn-warning"></span>

                <span class="btn btn-danger"></span>
            </div>
        </div><!-- /.sidebar-shortcuts -->

        <ul class="nav nav-list">
            <li class="active">

```

```

        <a href="#">
            <i class="menu-icon fa fa-tachometer"></i>
            <span class="menu-text"> Dashboard </span>
        </a>

        <b class="arrow"></b>
    </li>

    <li class="">
        <a href="#" class="dropdown-toggle">
            <i class="menu-icon fa fa-desktop"></i>
            <span class="menu-text">
                Menu 1
            </span>
            <b class="arrow fa fa-angle-down"></b>
        </a>

        <b class="arrow"></b>
        <ul class="submenu">
            <li class="">
                <a href="#">
                    <i class="menu-icon fa fa-caret-right"></i>
                    Sub Menu 1.1
                </a>
                <b class="arrow"></b>
            </li>
            <li class="">
                <a href="#">
                    <i class="menu-icon fa fa-caret-right"></i>
                    Sub Menu 1.2
                </a>
                <b class="arrow"></b>
            </li>
        </ul>
    </li>

    <li class="">
        <a href="#" class="dropdown-toggle">
            <i class="menu-icon fa fa-list"></i>
            <span class="menu-text"> Menu 2 </span>
            <b class="arrow fa fa-angle-down"></b>
        </a>
        <b class="arrow"></b>
        <ul class="submenu">
            <li class="">
                <a href="#">
                    <i class="menu-icon fa fa-caret-right"></i>
                    Sub Menu 2.1
                </a>
                <b class="arrow"></b>
            </li>
            <li class="">
                <a href="#">
                    <i class="menu-icon fa fa-caret-right"></i>
                    Sub Menu 2.2
                </a>
                <b class="arrow"></b>
            </li>
        </ul>
    </li>

    <li class="">
        <a href="#" class="dropdown-toggle">
            <i class="menu-icon fa fa-pencil-square-o"></i>
            <span class="menu-text"> Menu 3 </span>
            <b class="arrow fa fa-angle-down"></b>
        </a>
        <b class="arrow"></b>
    </li>

```

```

        <ul class="submenu">
            <li class="">
                <a href="#">
                    <i class="menu-icon fa fa-caret-right"></i>
                    Sub Menu 3.1
                </a>
                <b class="arrow"></b>
            </li>
            <li class="">
                <a href="#">
                    <i class="menu-icon fa fa-caret-right"></i>
                    Sub Menu 3.2
                </a>
                <b class="arrow"></b>
            </li>
        </ul>
    </li>
</ul><!-- /.nav-list -->
</div>

<div class="main-content">
    <div class="main-content-inner">
        <div class="breadcrumbs ace-save-state" id="breadcrumbs">
            <ul class="breadcrumb">
                <li>
                    <i class="ace-icon fa fa-home home-icon"></i>
                    <a href="#">Home</a>
                </li>
                <li class="active">Dashboard</li>
            </ul><!-- /.breadcrumb -->
        </div>

        <div class="page-content">
            <div class="page-header">
                <h1>
                    Dashboard
                    <small>
                        <i class="ace-icon fa fa-angle-double-right"></i>
                        overview &amp; stats
                    </small>
                </h1>
            </div><!-- /.page-header -->
            <div class="row">
                <div class="col-xs-12">
                    <div class="alert alert-block alert-info">
                        <button type="button" class="close" data-
dismiss="alert">
                            <i class="ace-icon fa fa-times"></i>
                        </button>
                        <i class="ace-icon fa fa-check green"></i>
                        Selamat datang
                        <strong class="green">
                            <b>Admin</b>
                        </strong>, <br><br>
                        Berusaha dan berdoa
                    </div>
                    <div class="widget-header widget-header-flat
widget-header-small">
                        <h5 class="widget-title">
                            <i class="ace-icon fa fa-signal"></i>
                            Kelola Halaman Web
                        </h5>
                    </div>
                    <br>
                    <div class="widget-body">
                        <div class="infobox infobox-green ">

```

```

        <div class="infobox-icon">
            <i class="ace-icon fa fa-
comments"></i>
        </div>
        <div class="infobox-data">
            <span class="infobox-data-
number">0</span>
            <div class="infobox-
content">Komentar &nbsp;
+<br/>
arrowed-right arrowed-in">New</span>
        </div>
        <div class="stat stat-
success">4</div>
        </div>
        <div class="infobox infobox-red">
            <div class="infobox-icon">
                <i class="ace-icon fa fa-flask"></i>
            </div>
            <div class="infobox-data">
                <span class="infobox-data-
number">0</span>
                <div class="infobox-
content">Testimoni &nbsp;
+<br/>
arrowed-right arrowed-in">New</span>
            </div>
            <div class="stat stat-
success">5</div>
            </div>
            <div class="infobox infobox-orange2">
                <div class="infobox-icon">
                    <i class="ace-icon fa fa-
download"></i>
                </div>
                <div class="infobox-data">
                    <span class="infobox-data-
number">100</span>
                    <div class="infobox-
content"><i>pageviews</i></div>
                    </div>
                </div>
            <!-- /.widget-body -->
            <!-- PAGE CONTENT ENDS -->
        </div><!-- /.col -->
        </div><!-- /.row -->
    </div><!-- /.page-content -->
</div><!-- /.main-content -->
<div class="footer">
    <div class="footer-inner">
        <div class="footer-content">
            <span class="bigger-120">
                <span class="blue bolder">Wawan Gunawan, S.Kom.,<br/>
M.T.</span>
                Application &copy; 2018
            </span>
            &nbsp; &nbsp;
            <span class="action-buttons">
                <a href="#">
                    <i class="ace-icon fa fa-twitter-square light-blue
bigger-150"></i>
            </span>
        </div>
    </div>
</div>

```

```

        </a>

        <a href="#">
            <i class="ace-icon fa fa-facebook-square text-primary bigger-150"></i>
        </a>

        <a href="#">
            <i class="ace-icon fa fa-rss-square orange bigger-150"></i>
        </a>
    </span>
</div>
</div>
</div>

<a href="#" id="btn-scroll-up" class="btn-scroll-up btn btn-sm btn-inverse">
    <i class="ace-icon fa fa-angle-double-up icon-only bigger-110"></i>
</a>
</div><!-- /.main-container -->

<script type="text/javascript" src="php echo base_url('assets/js/jquery-2.1.4.min.js'); ?&gt;"></script>
<script type="text/javascript">
    if('ontouchstart' in document.documentElement) document.write("<script
src='assets/js/jquery.mobile.custom.min.js'>"+"<"+"/script>");
</script>
<script type="text/javascript" src="php echo base_url('assets/js/bootstrap.min.js'); ?&gt;"></script>

<script type="text/javascript" src="php echo base_url('assets/js/jquery-ui.custom.min.js'); ?&gt;"></script>
<script type="text/javascript" src="php echo base_url('assets/js/jquery.ui.touch-punch.min.js'); ?&gt;"></script>
<script type="text/javascript" src="php echo base_url('assets/js/jquery.easypiechart.min.js'); ?&gt;"></script>
<script type="text/javascript" src="php echo base_url('assets/js/jquery.sparkline.index.min.js'); ?&gt;"></script>
<script type="text/javascript" src="php echo base_url('assets/js/jquery.flot.min.js'); ?&gt;"></script>
<script type="text/javascript" src="php echo base_url('assets/js/jquery.flot.pie.min.js'); ?&gt;"></script>
<script type="text/javascript" src="php echo base_url('assets/js/jquery.flot.resize.min.js'); ?&gt;"></script>

<!-- ace scripts -->
<script type="text/javascript" src="php echo base_url('assets/js/ace-elements.min.js'); ?&gt;"></script>
<script type="text/javascript" src="php echo base_url('assets/js/ace.min.js'); ?&gt;"></script>

</body>
</html>

```

Pada script di atas, pembuatan header, footer, sidebar, dan content dilakukan pada file yang sama. Kondisi tersebut akan sangat memberatkan programmer jika aplikasi bisnis yang dikembangkan memiliki banyak menu. Oleh karena itu untuk mempermudah dalam pembuatan program, maka kita lakukan pembuatan header, footer, sidebar, dan content

dengan metode MVC dengan melakukan perubahan script yang telah dibuat di atas seperti pada ilustrasi di bawah ini.

### File header.php

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Pemrograman Web 2</title>

    <script src="<?php echo base_url('assets/js/jquery-
2.1.4.min.js');?>"></script>
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/bootstrap.min.css');?>">
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/font-awesome/4.5.0/css/font-awesome.min.css');?>">

    <!-- dataTables -->
    <link rel="stylesheet" href="<?php echo
base_url('assets/css/dataTables.bootstrap.min.css');?>" />
    <link rel="stylesheet" href="<?php echo
base_url('assets/css/responsive.bootstrap.min.css');?>" />

    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/fonts.googleapis.com.css');?>">
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/ace.min.css');?>" class="ace-main-stylesheet" id="main-
ace-style" />
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/ace-skins.min.css');?>">
    <link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/ace-rtl.min.css');?>">
    <script type="text/javascript" src="<?php echo
base_url('assets/js/ace-extra.min.jss');?>"></script>

</head>
<body class="no-skin">
    <div id="navbar" class="navbar navbar-default ace-save-
state">
        <div class="navbar-container ace-save-state" id="navbar-
container">
            <button type="button" class="navbar-toggle menu-
toggler pull-left" id="menu-toggler" data-target="#sidebar">
                <span class="sr-only">Toggle sidebar</span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
                <span class="icon-bar"></span>
            </button>
            <div class="navbar-header pull-left">
                <a href="" class="navbar-brand">
                    <small>
```

```

                <i class="fa fa-leaf"></i>
                Ace Admin
            </small>
        </a>
    </div>

    <div class="navbar-buttons navbar-header pull-right"
role="navigation">
        <ul class="nav ace-nav">
            <li class="light-blue dropdown-modal">
                <a data-toggle="dropdown" href="#">
                    
                    <span class="user-info">
                        <small>Selamat
                datang,</small>
                        Admin
                    </span>
                <i class="ace-icon fa fa-
caret-down"></i>
            </a>
            <ul class="user-menu dropdown-
menu-right dropdown-menu dropdown-yellow dropdown-caret dropdown-close">
                <li>
                    <a href="#">
                        <i class="ace-
icon fa fa-cog"></i>
                    <Ubah Password
                </a>
                </li>
                <li class="divider"></li>
                <li>
                    <a href="#">
                        <i class="ace-
icon fa fa-power-off"></i>
                    <Logout
                </a>
                </li>
            </ul>
        </li>
    </ul>
</div><!-- /.navbar-container -->
</div>

```

File footer.php

```

<div class="footer">
    <div class="footer-inner">
        <div class="footer-content">
            <span class="bigger-120">

```

```

        <span class="blue bolder">Wawan
Gunawan, S.Kom., M.T.</span>
        Application &copy; 2018
</span>

        &nbsp; &nbsp;
<span class="action-buttons">
<a href="#">
    <i class="ace-icon fa fa-
twitter-square light-blue bigger-150"></i>
</a>

<a href="#">
    <i class="ace-icon fa fa-
facebook-square text-primary bigger-150"></i>
</a>

<a href="#">
    <i class="ace-icon fa fa-
rss-square orange bigger-150"></i>
</a>
</span>
</div>
</div>

<a href="#" id="btn-scroll-up" class="btn-scroll-up btn
btn-sm btn-inverse">
    <i class="ace-icon fa fa-angle-double-up icon-only
bigger-110"></i>
</a>
</div>!!-- /.main-container -->

<script type="text/javascript" src="php echo
base_url('assets/js/jquery-2.1.4.min.js');?&gt;"&gt;&lt;/script&gt;
&lt;script type="text/javascript"&gt;
    if('ontouchstart' in document.documentElement)
document.write("&lt;script
src='assets/js/jquery.mobile.custom.min.js'"+"+"+ "/script&gt;");
&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo
base_url('assets/js/bootstrap.min.js');?&gt;"&gt;&lt;/script&gt;

&lt!-- page specific plugin scripts --&gt;
&lt;script src="<?php echo
base_url('assets/js/jquery.dataTables.min.js');?&gt;"&gt;&lt;/script&gt;
&lt;script src="<?php echo
base_url('assets/js/jquery.dataTables.bootstrap.min.js');?&gt;"&gt;&lt;/script&gt;
&lt;script src="<?php echo
base_url('assets/js/dataTables.buttons.min.js');?&gt;"&gt;&lt;/script&gt;

&lt;script type="text/javascript" src="<?php echo
base_url('assets/js/jquery-ui.custom.min.js');?&gt;"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo
base_url('assets/js/jquery.ui.touch-punch.min.js');?&gt;"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo
base_url('assets/js/jquery.easypiechart.min.js');?&gt;"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo
base_url('assets/js/jquery.sparkline.index.min.js');?&gt;"&gt;&lt;/script&gt;
&lt;script type="text/javascript" src="<?php echo
</pre

```

```

base_url('assets/js/jquery.flot.min.js');?>"></script>
    <script type="text/javascript" src="<?php echo
base_url('assets/js/jquery.flot.pie.min.js');?>"></script>
    <script type="text/javascript" src="<?php echo
base_url('assets/js/jquery.flot.resize.min.js');?>"></script>

    <!-- ace scripts -->
    <script type="text/javascript" src="<?php echo
base_url('assets/js/ace-elements.min.js');?>"></script>
    <script type="text/javascript" src="<?php echo
base_url('assets/js/ace.min.js');?>"></script>

</body>
</html>

```

## File menu.php

```

<div class="main-container ace-save-state" id="main-container">
    <script type="text/javascript">
        try{ace.settings.loadState('main-
container')}catch(e){}
    </script>

    <div id="sidebar" class="sidebar responsive ace-save-
state">
        <script type="text/javascript">
            try{ace.settings.loadState('sidebar')}catch(e){}
        </script>

        <div class="sidebar-shortcuts" id="sidebar-
shortcuts">
            <div class="sidebar-shortcuts-large" id="sidebar-
shortcuts-large">
                <button class="btn btn-success">
                    <i class="ace-icon fa fa-
signal"></i>
                </button>

                <button class="btn btn-info">
                    <i class="ace-icon fa fa-
pencil"></i>
                </button>

                <button class="btn btn-warning">
                    <i class="ace-icon fa fa-
users"></i>
                </button>

                <button class="btn btn-danger">
                    <i class="ace-icon fa fa-
cogs"></i>
                </button>
            </div>
            <div class="sidebar-shortcuts-mini" id="sidebar-
shortcuts-mini">

```

```

</span>
</span>
</span>
</span>

```

</div><!-- /.sidebar-shortcuts -->

- Dashboard
**</b>**
- Menu 1
**</b>**
  - Sub Menu 1.1
**</b>**
  - Sub Menu 1.2
**</b>**
- Menu 2
**</b>**

```

down"></b>
      </a>
      <b class="arrow"></b>
      <ul class="submenu">
        <li class="">
          <a href="#">
            <i class="menu-icon fa fa-caret-right"></i>
          </a>
          <b class="arrow"></b>
        </li>
        <li class="">
          <a href="#">
            <i class="menu-icon fa fa-caret-right"></i>
          </a>
          <b class="arrow"></b>
        </li>
      </ul>
    </li>

    <li class="">
      <a href="#" class="dropdown-toggle">
        <i class="menu-icon fa fa-pencil-square-o"></i>
      </a>
      <span class="menu-text"> Menu 3
        <b class="arrow fa fa-angle-down"></b>
      </span>
      <b class="arrow"></b>
      <ul class="submenu">
        <li class="">
          <a href="#">
            <i class="menu-icon fa fa-caret-right"></i>
          </a>
          <b class="arrow"></b>
        </li>
        <li class="">
          <a href="#">
            <i class="menu-icon fa fa-caret-right"></i>
          </a>
          <b class="arrow"></b>
        </li>
      </ul><!-- /.nav-list -->
    </div>

```

File home.php yang akan kita gunakan sebagai content dari web yang akan kita kembangkan.

```

        <div class="main-content">
            <div class="main-content-inner">
                <div class="breadcrumbs ace-save-state"
id="breadcrumbs">
                    <ul class="breadcrumb">
                        <li>
                            <i class="ace-icon fa fa-
home home-icon"></i>
                            <a href="#">Home</a>
                        </li>
                        <li class="active">Dashboard</li>
                    </ul><!-- /.breadcrumb -->
                </div>

                <div class="page-content">
                    <div class="page-header">
                        <h1>
                            Dashboard
                            <small>
                                <i class="ace-icon fa
fa-angle-double-right"></i>
                                overview & stats
                            </small>
                        </h1>
                    </div><!-- /.page-header -->
                    <div class="row">
                        <div class="col-xs-12">
                            <div class="alert
alert-block alert-info">
                                <button
type="button" class="close" data-dismiss="alert">
                                    <i
class="ace-icon fa fa-times"></i>
                                <i class="ace-
icon fa fa-check green"></i>
                                Selamat datang
                                <strong
class="green">
                                    <b>Admin</b>
                                </strong>, <br><br>
                                Berusaha dan
                                berdoa
                            </div>
                            <div class="widget-
header widget-header-flat widget-header-small">
                                <h5
class="widget-title">
                                    <i
class="ace-icon fa fa-signal"></i>
                                Kelola
                                Halaman Web
                            </h5>
                            </div>
                            <br>
                            <div class="widget-
body">

```

```
class="infobox infobox-green">>                                <div
class="infobox-icon">                            <i
class="ace-icon fa fa-comments"></i>          </div>
class="infobox-data">
0
class="infobox-content">Komentar  +>
New
</div>
</div>
class="stat stat-success">4</div>
class="infobox infobox-red">>                                <div
class="infobox-icon">                            <i
class="ace-icon fa fa-flask"></i>          </div>
class="infobox-data">
0
class="infobox-content">Testimoni  +>
New
</div>
</div>
class="stat stat-success">5</div>
class="infobox infobox-orange2">>                                <div
class="infobox-icon">                            <i
class="ace-icon fa fa-download"></i>          </div>
class="infobox-data">
100
class="infobox-content"><i>pageviews</i></div>
</div>
</div>
<!-- /.widget-->
body -->
```

```

        </div><!-- /.col -->
    </div><!-- /.row -->
</div><!-- /.page-content -->
</div>
</div><!-- /.main-content -->

```

setelah pembuatan script selesai kita lakukan, selanjutnya kita harus melakukan perubahan pada file layout.php yang ada pada folder controller seperti script di bawah ini.

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');

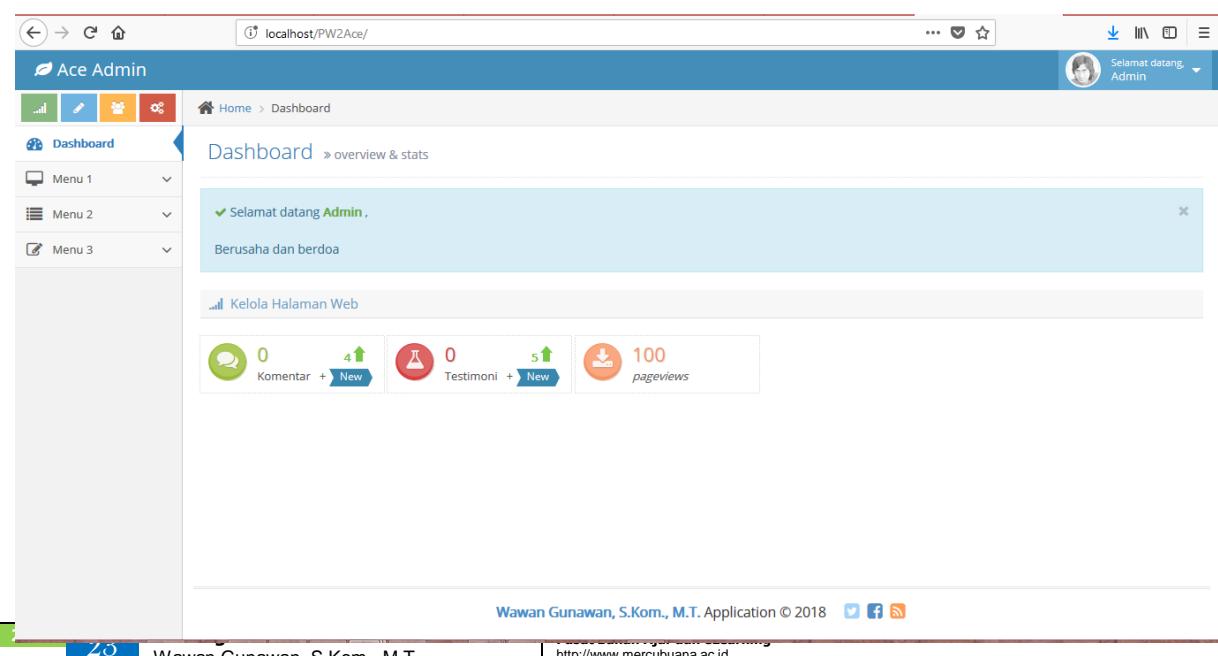
class layout extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
    }

    public function index()
    {
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('home');
        $this->load->view('footer');
    }
}

```

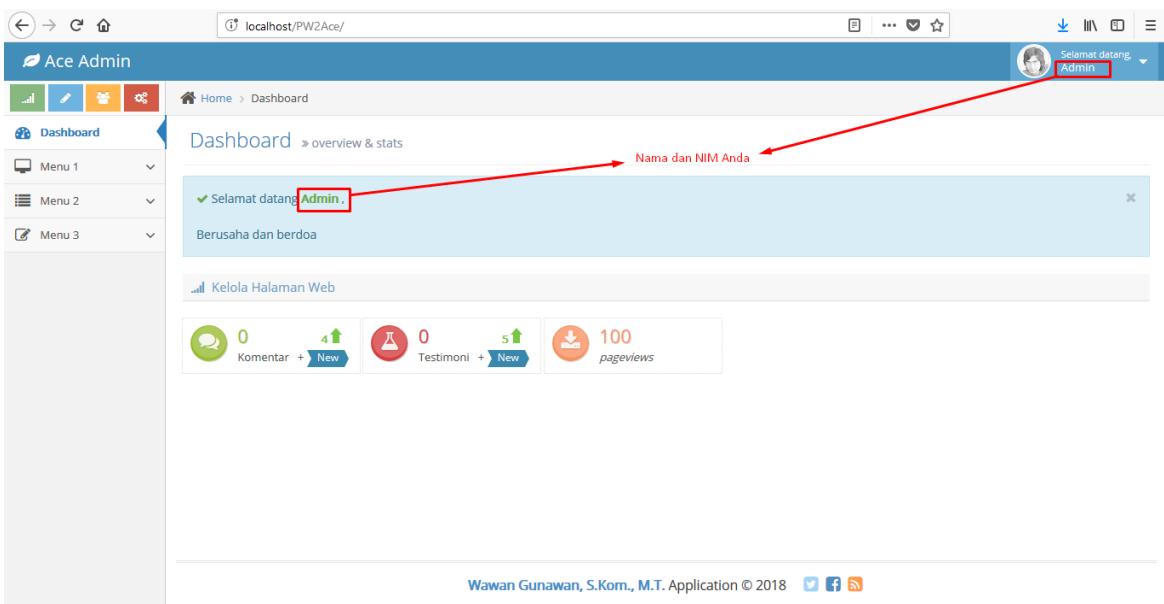
**Gambar 0.5 Tampilan gambar layout**

Selanjutnya jalankan aplikasi sederhana kita sehingga akan dihasilkan tampilan awal seperti gambar di bawah ini



# TUGAS INDIVIDU

1. Buatlah pecahan file layout.php menjadi file header.php, footer.php, menu.php dan home.php
2. Ubahlah nama Admin menjadi nama dan NIM anda, seperti pada tampilan di bawah ini



3. Buatlah langkah-langkah pembuatan file-file tersebut (header.php, footer.php, menu.php, dan home.php) beserta cara baca dari layout.php yang terdapat pada folder controller.
4. Screenshot hasil kerja Anda lalu kirimkan melalui email [wawan.gunawan@mercubuana.ac.id](mailto:wawan.gunawan@mercubuana.ac.id) paling lambat 5 hari setelah materi ini disampaikan.

**Catatan:** Untuk kelengkapan aplikasi yang akan dilakukan pemecahan dapat diunduh pada URL: <http://www.wawangunawan.web.id/download-22-Download> dengan keterangan **File ini merupakan kelengkapan dari modul pertemuan 5**

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

**Mengetahui dan memahami  
Konsep MVC: Models**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	06	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Pada model MVC, models dirancang untuk bekerja dalam memberikan informasi dari database. Sebagai contoh ketika kita menggunakan CodeIgniter untuk mengelola data kita, kita menggunakan model yang berfungsi untuk memasukan data, mengubah data, mengambil data, dan juga menghapus data dari dalam database. Berikut adalah contoh mengenai *class models* seperti pada script di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class groupProduct_model extends CI_Model {

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function error_db()
    {
        $error = $this->db->_error_message();
        if (isset($error['message'])) {
            echo json_encode(array("success"=>$error));
        } else {
            echo json_encode(array("success"=>true));
        }
    }

    public function add()
    {
        /* DAPAT MENGGUNAKAN METODE INI */
        $query = $this->db->insert('m_groupproduct',array(
            'GroupProduct_Name'=>$this->input->post('txtGroupProduk',true)
        ));

        /* ATAU MENGGUNAKAN METODE INI */
        $namaGroup = $this->input->post('txtGroupProduk');
        $query = $this->db->query("INSERT INTO m_groupproduct(GroupProduct_Name)
            VALUES ('$namaGroup')");

        $this->error_db();
    }

    public function edit($txtId)
    {
        /* DAPAT MENGGUNAKAN METODE INI */
        $this->db->where('GroupProduct_auto', $txtId);

        $query = $this->db->update('m_groupproduct',array(
            'GroupProduct_Name'=>$this->input->post('txtGroupProduk',true)
        ));

        /* ATAU MENGGUNAKAN METODE INI */
        $data =array('GroupProduct_Name'=>$this->input->post('txtGroupProduk',true));
        $where = "GroupProduct_auto = '$txtId'";
        $str = $this->db->update('m_groupproduct', $data, $where);

        /* ATAU MENGGUNAKAN METODE INI */
    }
}
```

```

$namaGroup = $this->input->post('txtGroupProduk');
$query = $this->db->query("UPDATE m_groupproduct
                            SET GroupProduct_Name='".$namaGroup'
                            WHERE GroupProduct_auto='".$txtId"'");

        $this->error_db();
    }

public function delete($id)
{
    /* DAPAT MENGGUNAKAN METODE INI */
    $query = $this->db->delete('m_groupproduct',array('GroupProduct_auto'=>$id));

    /* ATAU MENGGUNAKAN METODE INI */
    $this->db->where('GroupProduct_auto', $id);
    $this->db->delete('m_groupproduct');

    /* ATAU MENGGUNAKAN METODE INI */
    $query = $this->db->query("DELETE from m_groupproduct
                                WHERE GroupProduct_auto='".$id"'");

    $this->error_db();
}

public function getJson($offset,$limit,$field,$q='', $sort,$order){
    $sql = "SELECT * FROM m_groupproduct WHERE 1=1 ";
    if($q!=""){
        if($field=='all'){
            $sql .=" AND GroupProduct_Name LIKE '%{$q}%' ";
        } else {
            $sql .=" AND {$field} LIKE '%{$q}%' ";
        }
    }
    $result['count'] = $this->db->query($sql)->num_rows();
    $sql .= " order by {$sort} {$order} ";
    $sql .= " LIMIT {$offset},{$limit} ";
    $result['data'] = $this->db->query($sql)->result();

    return $result;
}
}

```

Keterangan script:

`$this->load->database();` digunakan untuk loading database

`public function error_db()` digunakan untuk parsing jika terjadi error pada perintah query

`public function add()` digunakan untuk melakukan proses penambahan (insert) database

`public function edit($txtId)` digunakan untuk melakukan proses perubahan (update) database

`public function` delete (\$id) digunakan untuk melakukan proses penghapusan (delete) database

`public function`  
getJson (\$offset, \$limit, \$field, \$q= '', \$sort, \$order) digunakan untuk melakukan proses menampilkan (select) database

## Memuat Model

Model biasanya akan dipanggil melalui fungsi controller, untuk memuat fungsi model dapat dilakukan dengan menggunakan script `$this->load->model ('model_name');` dan jika model terletak pada sub-folder dapat dilakukan pemanggilan dengan menggunakan script berikut ini `$this->load->model ('nama_folder/model_name');`

Setelah dimuat, kita akan mengakses fungsi model menggunakan objek dengan nama yang sama dengan class yang sudah kita buat dengan script berikut `$this->load->model ('model_name');`

## Auto-Loading Model

Jika model yang kita buat akan digunakan pada seluruh aplikasi, maka kita harus melakukan pengaturan pada application/config/autoload.php untuk melakukan inisialisasi system dan menambahkan model untuk array autoload

## Meghubungkan ke Database

Ketika kita melakukan proses load model, maka tidak akan secara otomatis tersambung ke database. Kita dapat meghubungkan menggunakan metode database standar, baik dari class Controller ataupun dari Model. Kita dapat melakukan dengan fungsi Model terhubung dengan nilai boolean (TRUE) melalui parameter ketiga dan pengaturan koneksi sebagaimana didefinisikan dalam file konfigurasi database yang kita gunakan. Contoh script dapat kita lakukan seperti berikut `$this->load->model ('Model_name', '', TRUE);`

Kita juga dapat melakukan secara manual melalui pengaturan database melalui parameter

```
$active_group = 'default';
$query_builder = TRUE;
```

```
$db['default'] = array(
    'dsn' => '',
    'hostname' => 'localhost',
    'username' => 'root',
    'password' => '',
    'database' => 'fasilkom',
    'dbdriver' => 'mysqli',
    'dbprefix' => '',
    'pconnect' => FALSE,
    'db_debug' => (ENVIRONMENT !== 'production'),
    'cache_on' => FALSE,
    'cachedir' => '',
    'char_set' => 'utf8',
    'dbcollat' => 'utf8_general_ci',
    'swap_pre' => '',
    'encrypt' => FALSE,
    'compress' => FALSE,
    'stricton' => FALSE,
    'failover' => array(),
    'save_queries' => TRUE
);
```

## FUNGSI HELPER

Helper berguna untuk membantu kita dengan tugas-tugas. Setiap file helper hanyalah kumpulan fungsi pada kategori tertentu. URL Helpers yang membantu dalam menciptakan link, elemen bentuk, format teks, mengatur dan membaca cookie, membantu menangani file, dan lain-lain.

Tidak seperti kebanyakan sistem lain dalam CodeIgniter, Helper tidak ditulis dalam format Object Oriented melainkan sangat sederhana yaitu menggunakan fungsi prosedural. Setiap fungsi Helper melakukan satu tugas tertentu tanpa ketergantungan pada fungsi lainnya.

CodeIgniter tidak memuat File Helper secara default, sehingga langkah pertama untuk menggunakan file Helper adalah dengan me-*load*-nya. Setelah di-*load* maka secara global akan dapat digunakan oleh file Controllers dan juga file Views.

### Loading a Helper

Untuk load file helper sangat sederhana dengan menggunakan fungsi berikut `$this->load->helper('url');`

Secara umum perancangan model dapat dibuat menggunakan aturan berikut:

- 1) Model harus memiliki atribut yang dapat mewakili element tertentu.

- 2) Model seharusnya berisi kode-kode yang menangani proses bisnis untuk data bersangkutan, misalnya untuk melakukan proses validasi data.
- 3) Model boleh berisi kode-kode yang digunakan untuk memanipulasi data, misalnya untuk menambah baris data baru, mengubah baris data, dan menghapus baris data pada sebuah tabel yang terdapat dalam database.
- 4) Model seharusnya tidak berisi kode-kode yang berkaitan langsung dengan permintaan yang dilakukan oleh user melalui URL (tidak berisi \$\_GET maupun \$\_POST). Pekerjaan seperti ini seharusnya dilakukan oleh controller.
- 5) Model seharusnya tidak berisi kode-kode yang berkaitan dengan presentasi (tampilan) halaman web yang akan disajikan ke hadapan user. Pekerjaan semacam ini seharusnya dilakukan oleh view.

## Membuat Form Master data (Kelompok Produk)

Setelah kita berhasil membuat tampilan pada pembahasan sebelumnya, selanjutnya kita akan membuat contoh 1 (satu) form sedehana untuk pembuatan kelompok produk dengan hasil yang diharapkan dari pembuatan kode program pada bagian ini dapat dilihat seperti pada gambar di bawah ini

No	Nama	Kategori Berita	Aksi
1	Alat Tulis		
2	Buku Tulis		
3	Correction Tape		

**Gambar 0.1 Tampilan gambar kelompok produk yang diharapkan**

Keterangan gambar:

1. Pada gambar di atas yang ditunjukan dengan nomor 1 (satu) merupakan data yang berhasil ditampilkan pada datagrid
2. Pada gambar di atas yang ditunjukan dengan nomor 2 (dua) merupakan tampilan dialog-form untuk melakukan proses pencarian setelah memilih icon luv (search)
3. Pada gambar di atas yang ditunjukan dengan nomor 3 (tiga) merupakan tampilan dialog-form untuk melakukan proses Tambah Data setelah memilih icon Tambah Data dan melakukan proses Ubah Data setelah memilih icon Ubah Data

Untuk mendapatkan hasil yang sesuai dengan gambar di atas, maka kita harus mempersiapkan konfigurasi database, membuat controller, membuat views, dan juga membuat models.

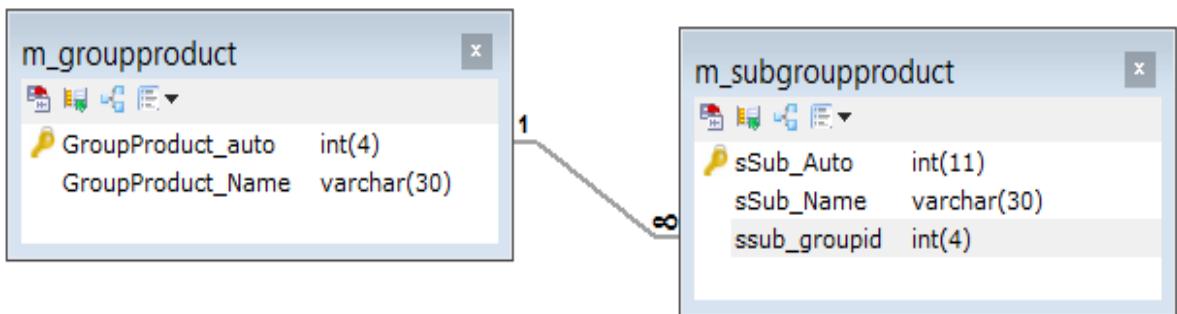
### 5.1. Konfigurasi Database

Agar aplikasi yang dibuat dapat berjalan menggunakan database dan dapat *handling error database*, maka kita harus melakukan konfigurasi database yang terdapat pada file application/config/database.php, seperti terlihat pada script di bawah ini

```
$active_group='default';
$active_record=TRUE;

$db['default']['hostname']='localhost';
$db['default']['username']='root';
$db['default']['password']='';
$db['default']['database']='fasilkom';
$db['default']['dbdriver']='mysql';
$db['default']['dbprefix']='';
$db['default']['pconnect']=TRUE;
$db['default']['db_debug']=FALSE;
$db['default']['cache_on']=FALSE;
$db['default']['cachedir']='';
$db['default']['char_set']='utf8';
$db['default']['dbcollat']='utf8_general_ci';
$db['default']['swap_pre']='';
$db['default']['autoinit']=TRUE;
$db['default']['stricton']=FALSE;
```

Setelah melakukan konfigurasi database, selanjutnya kita harus mempersiapkan database pada DBMS MySQL dengan nama database **fasilkom** serta beberapa tabel yang dibutuhkan untuk pembuatan aplikasi bisnis sederhana seperti pada rancangan LRS di bawah ini



M\_groupproduct

Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	Auto Incr?
* GroupProduct_auto	int	4		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
GroupProduct_Name	varchar	30	''	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```

CREATE TABLE `m_groupproduct` (
`GroupProduct_auto` int(4)NOT NULL auto_increment,
`GroupProduct_Name` varchar(30)default '',
PRIMARY KEY(`GroupProduct_auto`),
UNIQUE KEY `NewIndex1` (`GroupProduct_Name`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

M\_subgroupproduct

Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	Auto Incr?
* sSub_Auto	int	11		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
sSub_Name	varchar	30		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
ssub_groupid	int	4		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

```

CREATE TABLE `m_subgroupproduct` (
`sSub_Auto` int(11)NOT NULL auto_increment,
`sSub_Name` varchar(30)default NULL,
`ssub_groupid` int(4)NOT NULL,
PRIMARY KEY(`sSub_Auto`),
KEY `FK_m_subgroupproduct` (`ssub_groupid`),
CONSTRAINT `FK_m_subgroupproduct` FOREIGN KEY(`ssub_groupid`)
REFERENCES `m_groupproduct` (`GroupProduct_auto`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1

```

## 5.2. Menampilkan DataGrid

Selanjutnya lakukan perubahan pada file menu.php seperti script di bawah ini

```
<li class="">
    <a href="#" class="dropdown-toggle">
        <i class="menu-icon fa fa-desktop"></i>
        <span class="menu-text">
            Menu 1
        </span>
        <b class="arrow fa fa-angle-down"></b>
    </a>

    <b class="arrow"></b>
    <ul class="submenu">
        <li class="">
            <a href="#">
                <i class="menu-icon fa fa-caret-right"></i>
                Sub Menu 1.1
            </a>
            <b class="arrow"></b>
        </li>

        <li class="">
            <a href="#">
                <i class="menu-icon fa fa-caret-right"></i>
                Sub Menu 1.2
            </a>
            <b class="arrow"></b>
        </li>
    </ul>
</li>
```

Menjadi seperti di bawah ini

```
<li class="">
    <a href="#" class="dropdown-toggle">
        <i class="menu-icon fa fa-desktop"></i>
        <span class="menu-text">
            Menu 1
        </span>
        <b class="arrow fa fa-angle-down"></b>
    </a>

    <b class="arrow"></b>
    <ul class="submenu">
        <li class="">
            <a href="php echo site_url('groupProduct'); ?">
                <i class="menu-icon fa fa-caret-right"></i>
                Sub Menu 1.1
            </a>
            <b class="arrow"></b>
        </li>

        <li class="">
            <a href="#">
```

```

        <i class="menu-icon fa fa-caret-right"></i>
        Sub Menu 1.2
    </a>
    <b class="arrow"></b>
</li>
</ul>
</li>

```

Selanjutnya kita akan persiapkan pembuatan program controller dengan nama file `groupProduct.php` pada lokasi `application/controller/` untuk menjalankan instruksi dari script menu dengan script "`<?php echo site_url('groupProduct'); ?>`" yang telah kita buat sebelumnya, kemudian ketikkan script berikut ini

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');

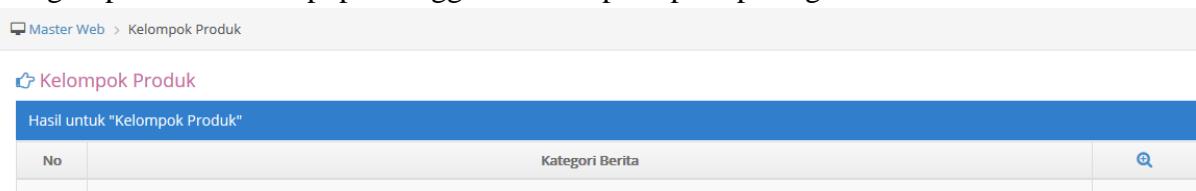
class GroupProduct extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
    }

    public function index()
    {
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('groupProduct_view');
        $this->load->view('footer');
    }
}

```

Keterangan script:

Ketika pengguna memilih menu Kelompok Produk, maka yang akan dijalankan kali pertama adalah controller `groupProduct.php/index`, selanjutnya pada function `index` akan menjalankan perintah `$this->load->view('groupProduct_view');` untuk melakukan eksekusi pada view `groupProduct_view.php` sehingga akan tampil seperti pada gambar



**Gambar 0.3 Tampilan Query Kelompok Produk yang diharapkan**

Selanjutnya buat file groupProduct\_view.php pada folder views untuk tampilan datagrid kita dengan script seperti di bawah ini

```
<!-- HEADER TITLE -->
<div class="main-content">
    <div class="main-content-inner">
        <div class="breadcrumbs ace-save-state" id="breadcrumbs">
            <ul class="breadcrumb">
                <li>
                    <i class="fa fa-desktop"></i>
                    <a href="#">Master Web</a>
                </li>
                <li class="active">Kelompok Produk</li>
            </ul><!-- /.breadcrumb -->
            <div class="nav-search" id="nav-search">
            </div><!-- /.nav-search -->
        </div>
        <div class="page-content">
            <div class="row">
                <div class="col-xs-12">
                    <div class="clearfix">
                        <h4 class="pink">
                            <i class="ace-icon fa fa-hand-o-right icon-animated-hand-pointer blue"></i>
                            Kelompok Produk
                        <div class="pull-right tableTools-container"></div>
                    </h4>
                    <div class="table-header">
                        Hasil untuk "Kelompok Produk"
                    </div>
                    <!-- BATAS HEADER TITLE -->
                    <div class="ln_solid"></div>
                    <!--DATAGRID BERDASARKAN DATA YANG AKAN KITA TAMPILKAN -->
                    <table id="datatable" class="table table-striped table-bordered">
                        <thead>
                            <tr>
                                <th class="center" width="6%">No</th>
                                <th class="center" width="85%">Kategori Produk</th>
                                <th class="center" >
                                    <a href="#" 
                                        class="tooltip-info" title="Tambah">
                                        <span class="blue"><i class="ace-icon fa fa-search-plus bigger-120"></i></span></a>
                                    </th>
                            </tr>
                        </thead>
                        <tr>
                            <td align="center"></td>
                            <td></td>
                            <td align="center"></td>
                        </tr>
                    </table>
                </div>
            </div>
        </div>
    </div>
</div>
```

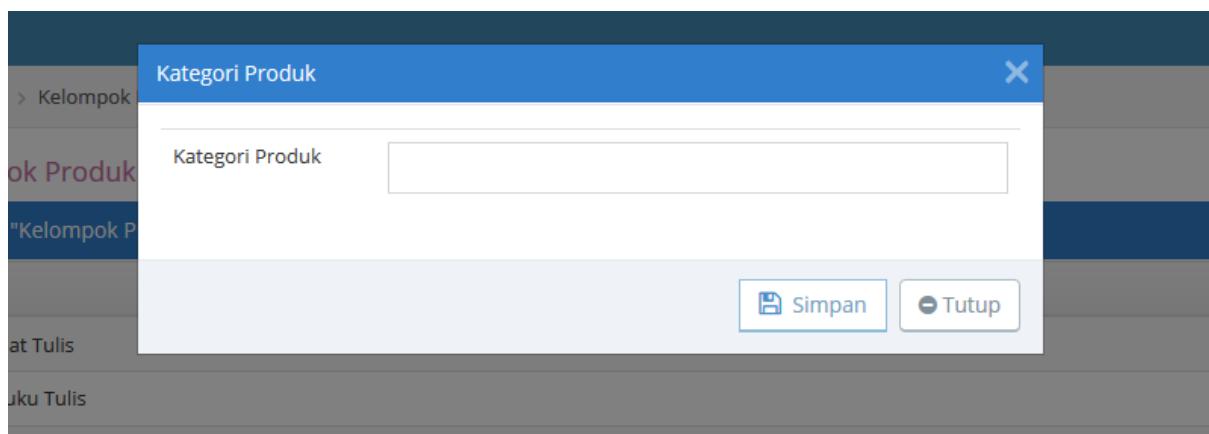
```

        <!-- BATAS DATAGRID BERDASARKAN DATA YANG AKAN
KITA TAMPILKAN -->
            </div>
        </div>
    </div>
</div>

```

### 5.3. Menambahkan Data

Selanjutnya yang akan kita persiapkan adalah bagaimana membuat Form sederhana ketika kita memilih icon Tambah padadatagrid dengan tampilan seperti pada gambar di bawah ini



#### View

Tambahkan pada file groupProduct\_view.php dengan menambahkan script seperti di bawah ini

```

<div class="modal fade" id="m_kategori" tabindex="-1" role="dialog" aria-
hidden="true">
    <div class="modal-dialog">
        <div class="modal-content">
            <div class="modal-header no-padding">
                <div class="table-header">
                    <button type="button" class="close" data-
dismiss="modal" aria-label="Close">
                        <span class="white">&times;</span>
                    </button>
                    Kategori Produk
                </div>
            </div>
            <div class="modal-body">
                <form name="f_kategori" id="f_kategori" >
                    <input type="hidden" name="id" id="id"
value="">
                    <div id="konfirmasi"></div>
                    <table class="table table-form">
                        <tr><td style="width: 25%">Kategori
Produk</td>
                        <td style="width: 75%">

```

```

        <input type="text"
class="form-control" name="txtGroupProduk" id="txtGroupProduk" required
value="">
            </td>
        </tr>
    </table>
</div>
<div class="modal-footer">
    <button class="btn btn-white btn-info btn-bold"
type="button" id="btnSimpan" name="btnSimpan">
        <i class="ace-icon fa fa-floppy-o bigger-120
blue"></i> Simpan</button>
    <button class="btn btn-white btn-default btn-round"
data-dismiss="modal" aria-hidden="true">
        <i class="fa fa-minus-circle"></i>
Tutup</button>
    </div>
</form>
</div>
</div>
</div>

<script>
$(document).ready(function() {

    $('#datatable thead').on( 'click', '.fa-search-plus', function () {
        $('#m_kategori').modal('show');
        $('#txtKodeProduk').val('');
        $('#txtGroupProduk').val('');

    });

    $('#btnSimpan').click(function(){

        $.post( "<?php echo base_url(); ?>groupProduct/add/", {
            type:1,
            txtKodeProduk:$('#txtKodeProduk').val(),
            txtGroupProduk:$('#txtGroupProduk').val()
        }, function( data ) {

            console.log(data);
            obj = $.parseJSON(data);

            if (obj.msg[0]==="ok"){
                $('#konfirmasi').html(
                    '<div class="alert alert-success alert-
dismissible fade in" role="alert">' +
                        ' <button type="button" class="close" data-
dismiss="alert" aria-label="Close"><span aria-
hidden="true">&times;</span></button>' +
                        ' <strong>Sukses</strong><br/>' +obj.msg[1]+
                    '</div>');
            }

            setTimeout(function() {
                $('#konfirmasi').html('');
            },2000);

            $('#txtKodeProduk').val('');
            $('#txtGroupProduk').val('');

        });
    });

});

```

```

        } else {

            $('#konfirmasi').html(
                '<div class="alert alert-danger alert-
dismissible fade in" role="alert">' +
                    '    <button type="button" class="close" data-
dismiss="alert" aria-label="Close"><span aria-
hidden="true">&times;</span></button>' +
                    '    <strong>Error</strong><br/>' + obj.msg[1] +
                '</div>');
        }

        myTable.ajax.reload();
    });

});

});
```

## Controller

kita buat seperti di bawah ini

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');

class groupProduct extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('groupProduct_model');
    }

    public function index()
    {
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('groupProduct_view');
        $this->load->view('footer');
    }

    public function add()
    {
        if (!isset($_POST))
            show_404();

        $this->groupProduct_model->add();
    }
}
```

## Models

kita buat seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class groupProduct_model extends CI_Model {

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

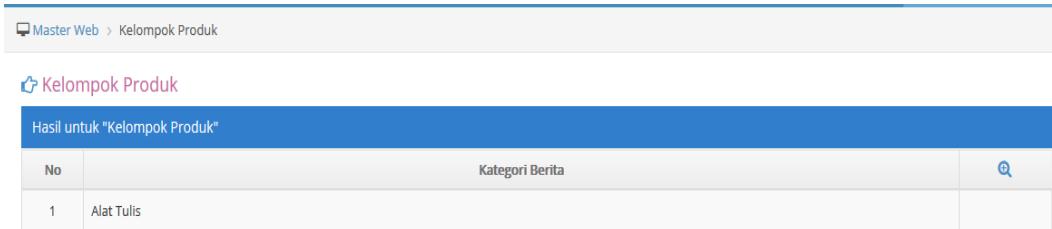
    public function add()
    {
        $namaGroup = $this->input->post('txtGroupProduk');
        $kodeGroup = $this->input->post('txtKodeProduk');

        if($this->db->simple_query("INSERT INTO
m_groupproduct(GroupProduct_Name)
VALUES ('$namaGroup')")) {
            $data['msg'][0] = "ok";
            $data['msg'][1] = "Data berhasil ditambahkan.....";
        } else {
            $error = $this->db->error();
            $myJSON = json_encode($error['code'] . ":" .
".$error['message']");
            $data['msg'][0] = "err";
            $data['msg'][1] = $myJSON;
        }

        echo json_encode($data);
    }
}
```

## 5.4. Menampilkan Data Query

Selanjutnya yang akan kita persiapkan adalah bagaimana menampilkan data pada datagrid dari database yang sudah kita input datanya ke data grid Kelompok Produk agar dapat menghasilkan seperti pada data di bawah ini dengan tampilan seperti pada gambar di bawah ini



No	Kategori Berita	
1	Alat Tulis	

Gambar 0.4 Tampilan query Kelompok Produk yang diharapkan

### Controller

kita buat seperti di bawah ini

```
<?php if(! defined('BASEPATH')) exit('No direct script access allowed');

class groupProduct extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('groupProduct_model');
    }

    public function index()
    {

        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('groupProduct_view');
        $this->load->view('footer');
    }
}
```

```
public function daftar_data()
{
    echo $this->groupProduct_model->getData();
}
```

```
public function add()
{
    if (!isset($_POST))
        show_404();

    $this->groupProduct_model->add();
}

}
```

## View

kita buat seperti di bawah ini

```
<script>
$(document).ready(function() {

    var myTable =$('#datatable').DataTable({
        "ajax": {
            type : "POST",
            url : "<?php echo base_url(); ?>groupProduct/daftar_data/",
            data : function(d) {
                ...
            },
            "columnDefs": [
                { "orderable": false, "targets": 2 },
                { "visible": false, "targets": [3], "searchable": false }
            ],
            select: {
                style: 'multi'
            }
        });
    });

    $('#datatable thead').on( 'click', '.fa-search-plus', function () {
        $('#m_kategori').modal('show');
        $('#txtKodeProduk').val('');
        $('#txtGroupProduk').val('');

    });
}) ;
```

## Models

kita buat seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class groupProduct_model extends CI_Model {

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function getData()
    {
        $i = 1;
        $groupproduct = $this->db->query("SELECT * from m_groupproduct");

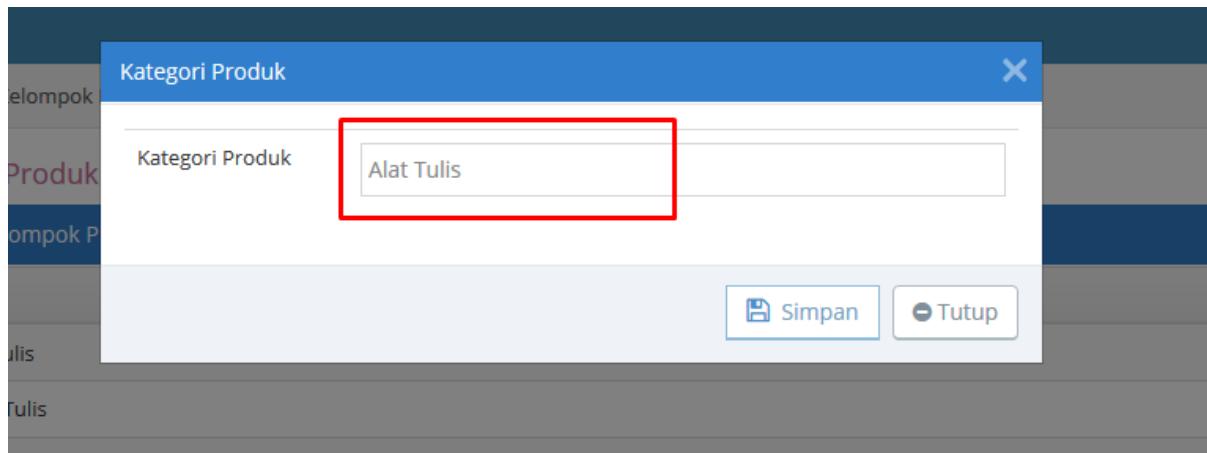
        foreach($groupproduct->result() as $rowa) {
            $akses=<center><a href='#!' class='tooltip-success' data-
rel='tooltip' title='Ubah'><span class='green'><i class='ace-icon fa-
pencil-square-o bigger-120'></i></span></a> <a href='#!' class='tooltip-
error' data-rel='tooltip' title='Hapus'><span class='red'><i class='ace-
icon fa-trash-o bigger-120'></i></span></a></center>;

            $row = array();
            $row[] = $i;
            $row[] = $rowa->GroupProduct_Name;
            $row[] = $akses;
            $row[] = $rowa->GroupProduct_auto;

            $data['data'][] = $row;
            $i++;
        }
        echo json_encode($data);
    }
}
```

## 5.5. Merubah Data

Selanjutnya yang akan kita persiapkan adalah bagaimana membuat Form sederhana ketika kita memilih icon Ubah data pada datagrid dengan tampilan seperti pada gambar di bawah ini



Gambar 0.5 Tampilan query untuk perubahan data (diharapkan)

## Models

kita buat seperti di bawah ini

```
public function __construct()
{
    parent::__construct();
    $this->load->database();
}

public function add()
{
    $namaGroup = $this->input->post('txtGroupProduk');
    $kodeGroup = $this->input->post('txtKodeProduk');

    $sql = "SELECT * FROM m_groupproduct WHERE
GroupProduct_auto='$kodeGroup'";
    $result =$this->db->query($sql);

    if ($result->num_rows() > 0){
        if($this->db->simple_query("UPDATE m_groupproduct
            SET GroupProduct_Name='$namaGroup'
            WHERE GroupProduct_auto='$kodeGroup'")) {
            $data ['msg'] [0] = "ok";
            $data ['msg'] [1] = "Data berhasil diubah.....";
        } else {
            $error = $this->db->error();
            $myJSON = json_encode($error['code'] . ":" .
".$error['message']");
            $data ['msg'] [0] = "err";
            $data ['msg'] [1] = $myJSON;
        }
    }
}
```

```

    } else {

        if($this->db->simple_query("INSERT INTO
m_groupproduct(GroupProduct_Name)
VALUES ('$namaGroup'))){
            $data['msg'][0] = "ok";
            $data['msg'][1] = "Data berhasil ditambahkan.....";
        } else {
            $error = $this->db->error();
            $myJSON = json_encode($error['code']:[
                ".{$error['message']}];
            $data['msg'][0] = "err";
            $data['msg'][1] = $myJSON;
        }
    }

}

```

## View

kita tambahkan groupProduct seperti di bawah ini

```

var myTable =$('#datatable').DataTable({
    "ajax": {
        type : "POST",
        url : "<?php echo base_url(); ?>groupProduct/daftar_data/",
        data : function(d) {

        },
        "columnDefs": [
            { "orderable": false, "targets": 2 },
            { "visible": false, "targets": [3], "searchable": false }
        ],
        select: {
            style: 'multi'
        }
    });

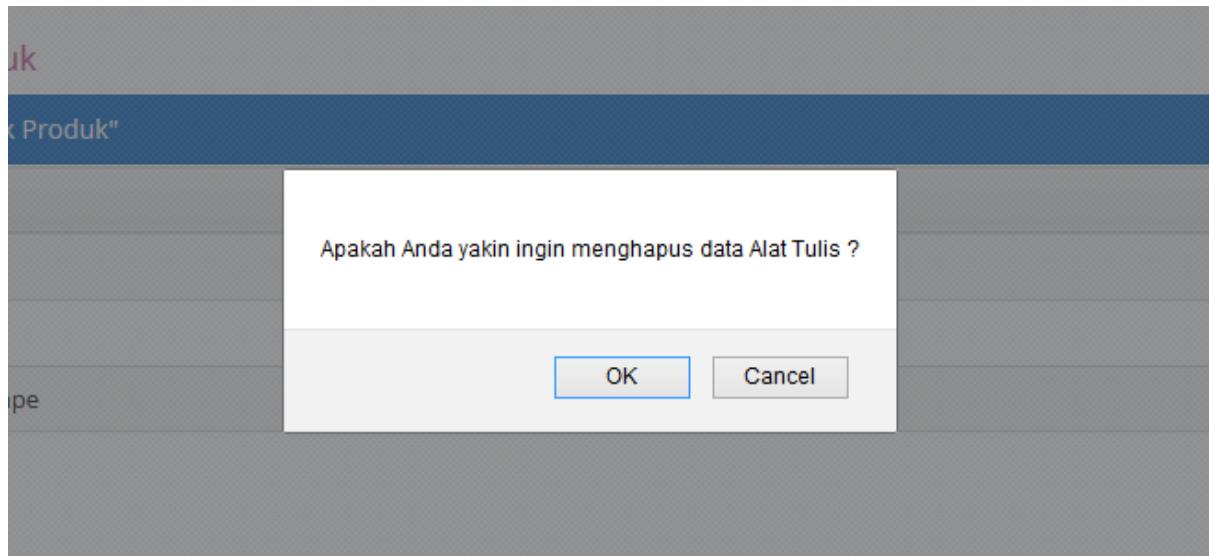
$('#datatable thead').on( 'click', '.fa-search-plus', function () {
    $('#m_kategori').modal('show');
    $('#txtKodeProduk').val('');
    $('#txtGroupProduk').val('');
});

$('#datatable tbody').on( 'click', '.fa-pencil-square-o', function () {
    $('#m_kategori').modal('show');
    nama = myTable.row( $(this).parents('tr') ).data()[1];
    id = myTable.row( $(this).parents('tr') ).data()[3];
    $('#txtGroupProduk').val(nama);
    $('#txtKodeProduk').val(id);
});

```

## 5.6. Menghapus Data

Selanjutnya yang akan kita persiapkan adalah bagaimana membuat Form sederhana ketika kita memilih icon Hapus data pada datagrid dengan tampilan seperti pada gambar di bawah ini



Gambar 0.6 Tampilan Query Hapus data (yang diharapkan)

## View

kita tambahkan groupProduct seperti di bawah ini

```
var myTable =$('#datatable').DataTable({
    "ajax": {
        type : "POST",
        url : "<?php echo base_url(); ?>groupProduct/daftar_data/",
        data : function(d) {
            ...
        }
    },
    "columnDefs": [
        { "orderable": false, "targets": 2 },
        { "visible": false, "targets": [3], "searchable": false }
    ],
    select: {
        style: 'multi'
    }
});
$( '#datatable tbody' ).on( 'click', '.fa-trash-o', function () {
    var jawab;
```

```

        obj= $(this).parents('tr');
        id = myTable.row( $(this).parents('tr') ).data()[3];

        jawab=confirm("Apakah yakin untuk menghapus record
\n"+

                                "Perlengkapan:
"+myTable.row( $(this).parents('tr') ).data()[1]+"?");
        if (jawab==false){
            exit();
        }

        console.log(id);
        $.post( "<?php echo
base_url(); ?>groupProduct/hapus/", { id:id }, function( data ) {
            obj = $.parseJSON(data);
            $("#myModal3").modal('show');
            if (obj.msg[0]=="hapus"){

                $('#konfirmasihapus').html(
                    '<div class="alert alert-success
alert-dismissible fade in" role="alert">' +
                        ' <button type="button"
class="close" data-dismiss="alert" aria-label="Close"><span aria-
hidden="true">&times;</span></button>' +
                            ' <strong>Hapus
Data</strong><br/>' +obj.msg[1]+
                            '</div>');
                setTimeout(function() {
                    $('#konfirmasihapus').html('');
                    $("#myModal3").modal('hide');
                },2000);
                myTable.ajax.reload();
            } else {

                $('#konfirmasihapus').html(
                    '<div class="alert alert-danger alert-
dismissible fade in" role="alert">' +
                        ' <button type="button" class="close"
data-dismiss="alert" aria-label="Close"><span aria-
hidden="true">&times;</span></button>' +
                            '
<strong>Error</strong><br/>' +obj.msg[1]+
                            '</div>');
            }
        });
    });

    $('#datatable tbody').on( 'click', '.fa-pencil-square-o', function () {
        $("#m_kategori").modal('show');
        nama = myTable.row( $(this).parents('tr') ).data()[1];
        id = myTable.row( $(this).parents('tr') ).data()[3];
        $('#txtGroupProduk').val(nama);
        $('#txtKodeProduk').val(id);

    });

```

## Controller

kita tambahkan controller groupProduct.php seperti di bawah ini

```
public function add()
{
    if (!isset($_POST))
        show_404();

    $this->groupProduct_model->add();
    redirect('groupProduct');
}

public function hapus()
{
    if (!isset($_POST))
        show_404();

    $this->groupProduct_model->hapus();
}

}
```

## Models

kita tambahkan model groupProduct\_model.php seperti di bawah ini

```
public function hapus()
{
    $idGroup = $this->input->post('id');

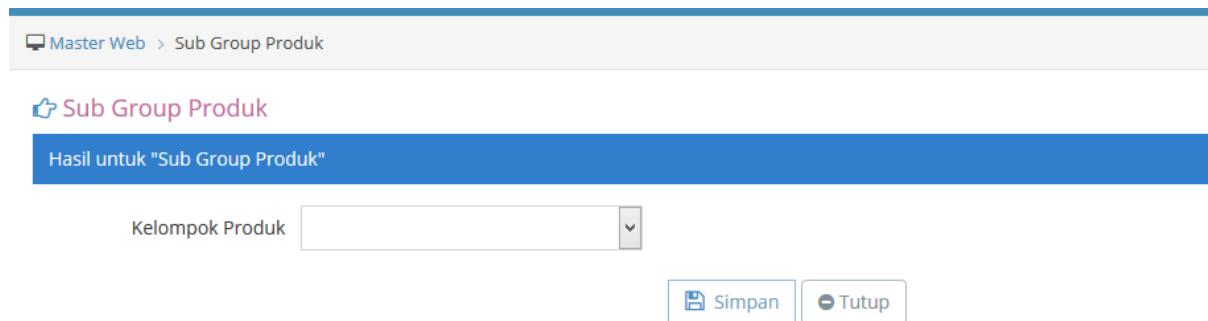
    if($this->db->simple_query("DELETE FROM m_groupproduct WHERE
GroupProduct_auto='".$idGroup."'"))
    {
        $data['msg'][0] = "hapus";
        $data['msg'][1] = "Data berhasil dihapus.....";
    } else {
        $error = $this->db->error();
        $myJSON = json_encode($error['code'].":"
". $error['message']);
        $data['msg'][0] = "err";
        $data['msg'][1] = $myJSON;
    }

    echo json_encode($data);
}
```

## 5.7. Membuat Combo Box

### Tampilan

Tambahkan pada file groupProduct\_view.php dengan menambahkan script seperti di bawah ini



### Menu

Lakukan penambahan pada file menu.php dengan script seperti di bawah ini

```
<li class="">
    <a href="#" class="dropdown-toggle">
        <i class="menu-icon fa fa-desktop"></i>
        <span class="menu-text">
            Menu 1
        </span>
        <b class="arrow fa fa-angle-down"></b>
    </a>

    <b class="arrow"></b>
    <ul class="submenu">
<li class="">
    <a href="php echo site_url('groupProduct'); ?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Group Produk
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;
&lt;li class=""&gt;
    &lt;a href="<?php echo
site_url('subGroupProduct'); ?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Sub Group Produk
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;
&lt;/ul&gt;
&lt;/li&gt;</pre
```

```
        </li>
    </ul>
</li>
```

## View

Tambahkan pada file subGroupProduct\_add.php dengan menambahkan script seperti di bawah ini

```
<!-- HEADER TITLE -->
<div class="main-content">
    <div class="main-content-inner">

        <div class="breadcrumbs ace-save-state" id="breadcrumbs">
            <ul class="breadcrumb">
                <li>
                    <i class="fa fa-desktop"></i>
                    <a href="#">Master Web</a>
                </li>
                <li class="active">Sub Group Produk</li>
            </ul><!-- /.breadcrumb -->
            <div class="nav-search" id="nav-search">
                </div><!-- /.nav-search -->
        </div>
        <div class="page-content">
            <div class="row">
                <div class="col-xs-12">
                    <div class="clearfix">
                        <h4 class="pink">
                            <i class="ace-icon fa fa-hand-o-right icon-animated-hand-pointer blue"></i>
                            Sub Group Produk
                            <div class="pull-right tableTools-container"></div>
                        </h4>
                    </div>
                    <div class="table-header">
                        Hasil untuk "Sub Group Produk"
                    </div>
                    <!-- BATAS HEADER TITLE -->
                    <div class="ln_solid"></div>

                    <!-- PAGE CONTENT BEGINS -->
                    <div class="modal-body">
                        <form class="form-horizontal" role="form" name="f_modal" id="f_modal" action="<?php echo base_url('subGroupProduct/add'); ?>" method="post" enctype="multipart/form-data">
                            <input type="hidden" name="txtKode" id="txtKode" value="">
                            <div class="form-group" control-label no-padding-right">
                                <label class="col-sm-2" for="cmbGroup"> Kelompok Produk </label>
                                <div class="col-sm-3">
                                    <select name="cmbGroup" id="cmbGroup" class="form-control" required data-parsley-error-message="Field ini harus diisi">
                                        <option>

```

```

        value=""></option>
    <?php
    foreach ($data
as $row) {
        echo
"<option value='".$row->GroupProduct_Auto' $cek>$row-
>GroupProduct_Name</option>";
    }
?>
        </select>
    </div>
</div>

        <div class="form-group">
            <label class="col-sm-7
control-label no-padding-right">
                <button class="btn
btn-white btn-info btn-bold" id="btnSimpan" name="btnSimpan"
type="submit"><i class="ace-icon fa fa-floppy-o bigger-120 blue"></i>
Simpan</button>
                <a href=""
target="_self" class="btn btn-white btn-default btn-round" data-
dismiss="modal" aria-hidden="true"><i class="fa fa-minus-circle"></i>
Tutup</a>
            </label>
        </div>
        </form>
    </div>
</div>
</div>
</div>
</div>

```

## Controller

Tambahkan pada file subGroupProduct.php dengan menambahkan script seperti di bawah ini

```

<?php if(! defined('BASEPATH')) exit('No direct script access allowed');

class subGroupProduct extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('subGroupProduct_model');
    }

    public function index()
    {
        $data['data'] = $this->subGroupProduct_model->getGroup();
        json_encode($data);
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('subGroupProduct_add', $data);
        $this->load->view('footer');
    }
}

```

## Model

Tambahkan pada file subGroupProduct\_model.php dengan menambahkan script seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class subGroupProduct_model extends CI_Model {

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function getGroup()
    {
        $groupproduct = $this->db->query("SELECT * from
m_groupproduct");
        return $groupproduct->result();
    }
}
```

## TUGAS

1. Buatlah rancangan menu sesuai dengan tema masing-masing!
2. Lengkapi Menu Sub Group Produk di atas dengan kelengkapan Query, Tambah Data, Ubah Data, dan Hapus Data secara berkelompok dan dikumpulkan 5 hari dari pertemuan hari ini

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

**Mengetahui dan memahami  
Kemanan Aplikasi Web**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	07	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Biasanya pada saat membangun suatu aplikasi kita akan lebih banyak fokus pada cara pembuatannya, bagaimana mengimplementasikan logika kita sehingga menghasilkan suatu sistem yang berjalan sesuai dengan keinginan dan kebutuhan kita. Hal ini tidak selalu salah dan juga tidak selalu benar.

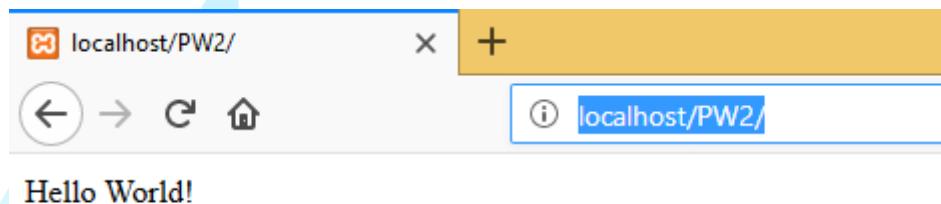
Adakalanya kita perlu membalik pola pikir ini dengan memandang diri kita sebagai pengguna (user) yang mungkin akan memiliki waktu lebih banyak dalam menggunakan aplikasi tersebut. Pada kasus ini mungkin tidak kita perhatikan dari aplikasi yang kita buat dengan pepatah “tidak ada sistem yang sempurna”. Pada pembahasan ini sedikit informasi mengenai keamanan dasar untuk membuat aplikasi berbasis web sehingga kita bisa sedikit melangkah menuju kesempurnaan tersebut.

## Mendefinisikan default Controller

CodeIgniter akan *me-load default* ketika tidak menampilkan URI, sehingga aplikasi akan menampilkan sesuai dengan *setting default* kita pada file router.php. Untuk membuat default dapat kita lakukan pada file application\config\routes.php dan isikan seperti pada gambar di bawah ini

```
| Examples: my-controller/index      -> my_controller/index  
|           my-controller/my-method -> my_controller/my_method  
*/  
$route['default_controller'] = 'hello';  
$route['404_override'] = '';  
$route['translate_uri_dashes'] = FALSE;
```

Di mana `hello` adalah nama Class Controller yang akan kita gunakan. Jika sekarang kita sudah membuat file `hello.php` maka akan tampil hasil seperti di bawah ini



Gambar 0.1 Tampilan mendefinisikan default controller

Beberapa hal yang perlu diperhatikan agar web kita aman dari percobaan cracker antara lain:

1. Penggunaan URL dengan Method GET
2. Melalui media input dengan menggunakan method POST

Seperti kita ketahui bahwa cara umum pengiriman variabel dalam PHP dapat menggunakan `$_GET` dan `$_POST` sehingga kita perlu penanganan khusus dalam penerimaan serta pengiriman variabel tersebut.

### 1. Method `$_POST`

Metode ini merupakan penangkapan variabel dari media input yang ada pada FORM, yang harus kita lakukan pada tahapan ini adalah:

- a. Batasi jumlah karakter (*maxlength*)

Pembatasan jumlah karakter pada media input sangat penting sehingga tidak dimungkinkan penulisan script pada area inputbox.

Contoh implementasi: `<input type="text" name="_txtUsername" id="_txtUsername" maxlength="50">`

Titik rawan: input login, input search

- b. Lapisi dengan penahan kutip (*addslashes()*)

Fungsi ini akan membantu kita menambahkan tanda kutip (‘) pada setiap karakter garis miring (/) sehingga dapat mencegah terjadinya SQL Injection. Penambahan kutip akan membuat garis miring menjadi bertipe string sehingga tidak dianggap sebagai salah satu perintah query SQL

Contoh implementasi: `addslashes($_POST['username']);`

Titik rawan: Form Login

- c. Lapisi dengan penahan tag (*htmlspecialchars()*)

Untuk mencegah terjadinya javascript injection maka ada baiknya kita menambahkan fungsi `htmlspecialchars()` tersebut. Hal ini akan membuat tag html maupun javascript memiliki tipe data string sehingga tidak akan dieksekusi oleh browser.

Contoh implementasi: `: htmlspecialchars($_POST['pesan']);`

Titik rawan: Form Login. Form Kontak

d. Perhatikan nama media input

Sudah saatnya kita berusaha sedikit lebih unik dalam menentukan nama pada media input, sehingga akan mempersulit beberapa tools jahat yang menggunakan library dalam penentuan nama variabel incarannya. Misalnya pada media input kita dapat menambahkan garis bawah ('\_') dalam memberikan identitas pada media input.

Contoh implementasi: `<input type="text" name="_txtUsername" id="_txtUsername" maxlength="50">`

Titik rawan: Input Login

## 2. Method \$\_GET

Metode ini merupakan penangkapan variabel dari URL, yang harus kita lakukan pada tahapan ini adalah:

a. Penggunaan .htaccess

Sebenarnya ini merupakan metode yang menjadikan website kita seakan memiliki struktur direktori yang luas, padahal ini merupakan suatu variabel. Sehingga dapat memcah konsentrasi pengunjung yang berniat iseng. Bila halaman tidak ditemukan maka kita dapat mengarahkannya atau mencirect ke halaman 404 kesayangan kita.

Implementasi:

```
RewriteEngine On
RewriteCond $1
!^(index\.php|assets|images|js|css|uploads|favicon\.png)
RewriteCond %{REQUEST_FILENAME} !-f
RewriteCond %{REQUEST_FILENAME} !-d
RewriteRule ^(.*)$ ./index.php?/$1 [L]
```

Selanjutnya menghilangkan index.php pada url codeigniter. cara nya buka pengaturan “index\_page” codeigniter pada *application/config/config.php*, kemudian cari pengaturan index\_page seperti di bawah ini.

```
$config['index_page'] = 'index.php';
```

Kemudian hapus index.php pada pengaturan sehingga menjadi

```
$config['index_page'] = '';
```

Simpan dan sekarang coba akses pada web untuk mencoba apakah sudah berhasil atau belum menghilangkan index pada codeigniter pada file controller hello.php yang sudah pernah dibuat.

b. Hindari \$\_REQUEST()

Metode ini digunakan untuk menangkap variabel dari methode \$\_POST dan \$\_GET. Ada saatnya kita menggunakan fungsi ini, namun ada baiknya kita melakukan pembatasan penggunaannya. Karena tanpa kita sadari penggunaan fungsi ini akan menjadi masalah di kemudian hari.

### 3. Session dan Cookies Timeout

Gunakan timeout pada session atau cookies terutama untuk penanganan login. Session merupakan penanda dari sisi server dan akan hilang jika browser kita tutup, sedangkan cookies merupakan penanganan dari sisi browser dan jika tidak diset timeout maka status akan terus login yang tentu saja ada kemungkinan-kemungkinan yang akan terjadi selanjutnya.

Implementasi

```
$data = array(
    'admin_id' => $a_data->kd_petugas,
    'admin_user' => $a_data->username,
    'admin_level' => $a_data->level,
    'admin_nama' => $a_data->nm_petugas,
        'admin_add' => $adding,
    'admin_edit' => $edit,
    'admin_delete' => $delete,
        'admin_valid' => true
);
$this->session->set_userdata($data);
```

Sebelum melanjutkan proses login, maka terlebih dahulu kita buat tabel petugas dengan struktur seperti berikut

Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?
* kd_petugas	char	4	''	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
nm_petugas	varchar	100	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
username	varchar	20	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
password	varchar	200	''	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
level	varchar	20	User	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
				<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

```
CREATE TABLE `petugas` (
  `kd_petugas` char(4) NOT NULL default '',
  `nm_petugas` varchar(100) NOT NULL default '',
  `username` varchar(20) NOT NULL,
  `password` varchar(200) NOT NULL,
  `level` varchar(20) NOT NULL default 'User',
PRIMARY KEY(`kd_petugas`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
```

Pada kesempatan ini kita hanya akan memiliki 3 (tiga) level pengguna dengan masing-masing hak akses tergantung dari levelnya. Level yang kita siapkan yaitu Admin, Petugas, dan User. Setelah kita membuat tabel petugas, selanjutnya isikan recordnya dengan script DML seperti di bawah ini

```
INSERT INTO `petugas` (`kd_petugas`, `nm_petugas`, `username`, `password`,
`level`)VALUES
('P001','Administrator','admin',md5('123456'),'Admin'),
('P002','Wawan','wawan',md5('123456'),'Petugas'),
('P003','Gunawan','gunawan',md5('123456'),'User');
```

Untuk memulai penggunaan session maka kita membutuhkan beberapa library yang harus kita aktifkan, yaitu library database dan juga library session serta memastikan bahwa helper url diaktifkan. Berikut adalah cara untuk mengaktifkan library yang akan digunakan.

Aktifkan library database dan library session pada file autoload.php di lokasi application/config/autoload.php

Sebelum dilakukan perubahan

```
$autoload['libraries']=array();
```

Ubah menjadi

```
$autoload['libraries']=array('database','session');
```

Selain itu aktifkan helper url pada file autoload.php di lokasi application/config/autoload.php

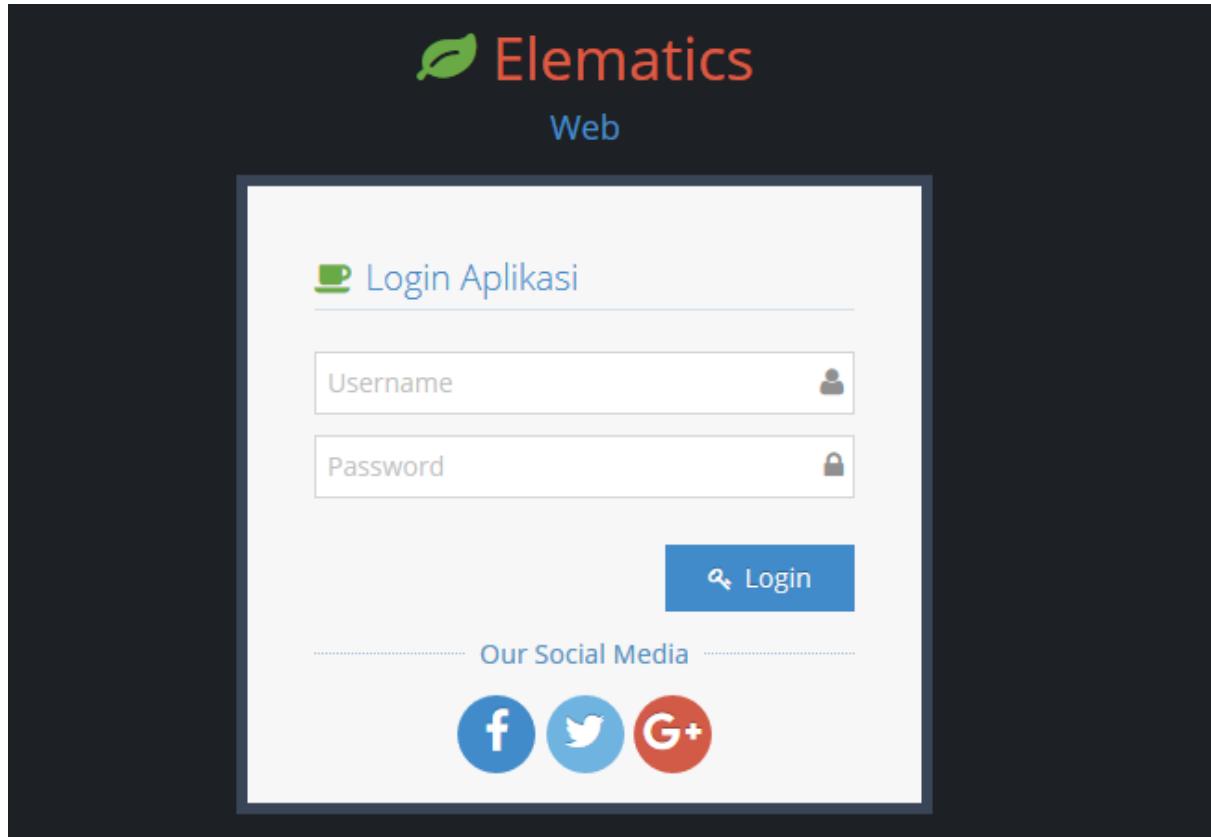
Sebelum dilakukan perubahan

```
$autoload['helper']=array();
```

Ubah menjadi

```
$autoload['helper']=array('url');
```

Membuat Form Login



## Controller

kita buat controller dengan nama login.php dengan script seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class login extends CI_Controller {

    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('login_model');
    }

    public function index()
```

```

{
    $this->load->view('login');
}

public function validasi_login()
{
    if(!isset($_POST)) {
        show_404();
    }

    $this->login_model->validasi_login();
    redirect('layout');
}

```

Kita tambahkan controller layout.php menjadi seperti di bawah ini

```

<?php if(! defined('BASEPATH'))exit('No direct script access allowed');

class layout extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->library('session');
    }

    public function cek_aktif() {
        if ($this->session->userdata('admin_id') == "") {
            redirect('layout/login');
        }
    }

    public function login() {
        $this->load->view('login');
    }

    public function index()
    {
        $this->cek_aktif();
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('home');
        $this->load->view('footer');
    }

    public function logout() {
        $data = array(
            'admin_id'          => '',
            'admin_user'         => '',
            'admin_level'        => '',
            'admin_nama'         => '',
            'admin_add'          => '',
            'admin_edit'         => '',
            'admin_delete'       => '',
            'admin_valid'        => false
        );
        $this->session->set_userdata($data);
    }
}

```

```
        redirect('layout');
    }
}
```

## View

kita buat seperti di bawah ini

```
<!DOCTYPE html>
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
    <title>Pemrograman Web 2</title>

    <link rel="stylesheet" type="text/css" href="php echo
base_url('assets/css/bootstrap.min.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/font-awesome/4.5.0/css/font-awesome.min.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/fonts.googleapis.com.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/ace.min.css');?&gt;" class="ace-main-stylesheet" id="main-
ace-style" /&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/ace-skins.min.css');?&gt;"&gt;
    &lt;link rel="stylesheet" type="text/css" href="<?php echo
base_url('assets/css/ace-rtl.min.css');?&gt;"&gt;
    &lt;script type="text/javascript" src="<?php echo
base_url('assets/js/ace-extra.min.js');?&gt;"&gt;&lt;/script&gt;

&lt;/head&gt;
&lt;body class="login-layout"&gt;
    &lt;div class="main-container"&gt;
        &lt;div class="main-content"&gt;
            &lt;div class="row"&gt;
                &lt;div class="col-sm-10 col-sm-offset-1"&gt;
                    &lt;div class="login-container"&gt;
                        &lt;div class="center"&gt;
                            &lt;h1&gt;
                                &lt;i class="ace-icon fa
fa-leaf green"&gt;&lt;/i&gt;
                                &lt;span
class="red"&gt;Elementics&lt;/span&gt;
                                &lt;span
id="id-text2"&gt;&lt;/span&gt;
                            &lt;?php echo "Web"; ?&gt;&lt;/h1&gt;
                            &lt;h4 class="blue" id="id-
company-text"&gt;&lt;?php echo "Web"; ?&gt;&lt;/h4&gt;
                        &lt;/div&gt;
                        &lt;div class="space-6"&gt;&lt;/div&gt;
                        &lt;div class="position-relative"&gt;
                            &lt;div id="login-box"
class="login-box visible widget-box no-border"&gt;
                                &lt;div class="widget-"</pre
```

```

body">
    <div
        class="widget-main">
            <h4
                class="header blue lighter bigger">
                    <i
                        class="ace-icon fa fa-coffee green"></i>
                    Login Aplikasi
                </h4>
            <div
                class="space-6"></div>
            <form
                name="logForm" id="theform" data-parsley-validate action="<?php echo
                base_url('login/validasi_login'); ?>" method="post">
                <fieldset>
                    <label class="block clearfix">
                        <span class="block input-icon input-icon-right">
                            <input type="text" class="form-control"
                                placeholder="Username" required="required" id="username" name="username"
                                autofocus />
                            <i class="ace-icon fa fa-user"></i>
                        </span>
                    </label>

                    <label class="block clearfix">
                        <span class="block input-icon input-icon-right">
                            <input type="password" class="form-control"
                                placeholder="Password" required="required" id="password" name="password" />
                            <i class="ace-icon fa fa-lock"></i>
                        </span>
                    </label>

                    <div class="space"></div>
                    <div class="clearfix">
                        <button type="submit" name="btnLogin" class="width-35 pull-right
                            btn btn-sm btn-primary">
                            <i class="ace-icon fa fa-key"></i>
                            <span class="bigger-110">Login</span>
                        </button>
                    </div>
                </fieldset>
            </form>
        </div>
    </div>

```

```

        </button>

    </div>

    <div class="space-4"></div>
</fieldset>
</form>

<div
class="social-or-login center">

    <span class="bigger-110">Our Social Media</span>
</div>

<div
class="space-6"></div>
<div
class="social-login center">
    <a
class="btn btn-primary">
        <i class="ace-icon fa fa-facebook"></i>
    </a>
    <a
class="btn btn-info">
        <i class="ace-icon fa fa-twitter"></i>
    </a>
    <a
class="btn btn-danger">
        <i class="ace-icon fa fa-google-plus"></i>
    </a>
</div><!--
/.widget-main -->

</div><!-- /.widget-
body -->
</div><!-- /.login-box -->
</div><!-- /.position-relative -->

</div><!-- /.col -->
</div><!-- /.row -->
</div><!-- /.main-content -->
</div><!-- /.main-container -->

<script src="__/js/jquery-1.11.3.min.js"></script>
<script src="__/js/bootstrap.js"></script>

</body>
</html>

```

## Models

kita buat login\_model.php dengan script seperti di bawah ini

```
<?php if ( ! defined('BASEPATH')) exit('No direct script access allowed');

class login_model extends CI_Model {

    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function validasi_login()
    {
        $username = $this->input->post('username');
        $password = $this->input->post('password');
        $password2 = md5($password);

        $q_data = $this->db->query("SELECT * FROM petugas
WHERE
        username = '". $username ."' AND
password = '$password2' ");
        $j_data = $q_data->num_rows();
        $a_data = $q_data->row();

        $adding = false; $edit = false; $delete = false;

        if ($j_data == 1) {
            if ($a_data->level == "Petugas") {
                $adding = true; $edit = true; $delete = true;
            } else if ($a_data->level == "User") {
                $adding = true;
            }
        }

        $data = array(
            'admin_id' => $a_data->kd_petugas,
            'admin_user' => $a_data->username,
            'admin_level' => $a_data->level,
            'admin_nama' => $a_data->nm_petugas,
            'admin_add' => $adding,
            'admin_edit' => $edit,
            'admin_delete' => $delete,
            'admin_valid' => true
        );
        $this->session->set_userdata($data);
        echo json_encode(array('success'=>true, 'msg'=>'Berhasil
login'));
    } else {
        echo json_encode(array('msg'=>'Username atau Password
salah!!!'));
    }
}
```

Melakukan perbaikan atau perubahan pada file menu.php untuk penentuan hak akses atas aplikasi yang dibuat.

kita buat seperti di bawah ini

```
<b class="arrow"></b>
<ul class="submenu">
    <?php if(($this->userdata('admin_level')=="Petugas") || 
        ($this->userdata('admin_level')=="User")){ ?>
        <li class="">
            <a href="php echo site_url('groupProduct');?&gt;"&gt;
                &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
                Group Produk
            &lt;/a&gt;
            &lt;b class="arrow"&gt;&lt;/b&gt;
        &lt;/li&gt;
    &lt;?php } ?&gt;
    &lt;li class=""&gt;
        &lt;a href="<?php echo site_url('subGroupProduct');?&gt;"&gt;
            &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
            Sub Group Produk
        &lt;/a&gt;
        &lt;b class="arrow"&gt;&lt;/b&gt;
    &lt;/li&gt;
&lt;/ul&gt;</pre
```

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

## **Handling File**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>09</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Sebelum melakukan pembuatan handling file, sebaiknya kita persiapkan terlebih dahulu menu Penanganan File pada folder views di file menu.php dengan menambahkan script seperti di bawah ini

```
<li class="">
    <a href="php echo
site_url('laporan'); ?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Laporan
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;

&lt;li class=""&gt;
    &lt;a href="<?php echo site_url('handlingFile'); ?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Penanganan File
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;

&lt;/ul&gt;
&lt;/li&gt;

&lt;li class=""&gt;
    &lt;a href="#" class="dropdown-toggle"&gt;
        &lt;i class="menu-icon fa fa-pencil-square-o"&gt;&lt;/i&gt;
        &lt;span class="menu-text"&gt; Menu 3 &lt;/span&gt;
        &lt;b class="arrow fa fa-angle-down"&gt;&lt;/b&gt;
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;</pre
```

## 9.1 Membuat Views

Selanjutnya untuk membuat handling file, maka kita harus mempersiapkan beberapa hal.

Pastikan arahan file kita sesuai dengan folder-folder file tersebut. Script yang harus kita persiapkan seperti pada gambar di bawah ini dengan nama file handlingFile.php

```
<!-- HEADER TITLE -->
<div class="main-content">
    <div class="main-content-inner">
        <div class="breadcrumbs ace-save-state"
id="breadcrumbs">
            <ul class="breadcrumb">
                <li>
```

```

        <i class="fa fa-desktop"></i>
        <a href="#">Master Web</a>
    </li>
    <li class="active">Handling File</li>
</ul><!-- /.breadcrumb -->
<div class="nav-search" id="nav-search">
</div><!-- /.nav-search -->
</div>
<div class="page-content">
    <div class="row">
        <div class="col-xs-12">
            <div class="clearfix">
                <h4 class="pink">
                    <i class="ace-icon fa fa-hand-o-right icon-animated-hand-pointer blue"></i>
                    Handling File
                <div class="pull-right
tableTools-container"></div>
                </h4>
            </div>
            <div class="table-header">
                Hasil untuk "Handling
File"
            </div>
            <!-- BATAS HEADER TITLE -->
            <div class="ln_solid"></div>

            <!-- PAGE CONTENT BEGINS -->
            <div class="modal-body">
                <form class="form-horizontal"
role="form" name="f_modal" id="f_modal" action="php echo
base_url('handlingFile/simpan'); ?&gt;" method="post"
enctype="multipart/form-data"&gt;
                    &lt;div class="form-
group"&gt;
                        &lt;label
class="col-sm-4 control-label no-padding-right"&gt; File &lt;/label&gt;
                        &lt;div class="col-
sm-3"&gt;
                            &lt;input
type="file" name="uploadfiles[]" id="uploadfiles" multiple
class="btn btn-info upload"&gt;
                        &lt;/div&gt;
                    &lt;/div&gt;
                    &lt;div class="form-
group"&gt;
                        &lt;label
class="col-sm-7 control-label no-padding-right"&gt;
                            &lt;button
class="btn btn-white btn-info btn-bold" id="btnSimpan"
name="btnSimpan" type="submit"&gt;&lt;i class="ace-icon fa fa-floppy-o
bigger-120 blue"&gt;&lt;/i&gt; Proses&lt;/button&gt;
                        &lt;/label&gt;
                    &lt;/div&gt;
                &lt;/form&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;
</pre

```

```

                </div>
            </div>
        </div>
    </div>
</script>

```

Keterangan:

Pada script `<input type="file" name="uploadfiles[]" id="uploadfiles" multiple class="btn btn-info upload">` kita menggunakan `multiple` dengan tujuan agar kita langsung dapat upload lebih dari satu file secara bersamaan. Sehingga proses upload file menjadi lebih cepat dan efisien.

Selain itu, ketika proses tersebut kita menggunakan `multiple` maka penyimpanan data pada database akan menjadi berbentuk array.

## 9.2 Membuat Controllers

Buat controller handlingFile.php dengan script seperti di bawah ini

```

<?php if(! defined('BASEPATH')) exit('No direct script access
allowed');

class handlingFile extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
    }

    public function index()
    {
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('handlingFile');
        $this->load->view('footer');
    }

    public function simpan()
    {
        $kodeKelompok = "A002";
        $path = 'images';
        if(!file_exists($path)){
            mkdir($path, 0777, true);
        }
        $path=$path.'/'.$kodeKelompok;
    }
}

```

```

if(!file_exists($path)){
    mkdir($path, 0777, true);
}
$files = json_encode($_FILES['uploadfiles']['name']);

for($i=0;$i<count($_FILES['uploadfiles']['name']);$i++) {
    move_uploaded_file($_FILES["uploadfiles"]["tmp_name"][$i],
$path.'/'.$_FILES['uploadfiles']['name'][$i]);
}

$query = $this->db->query("INSERT INTO
tbl_gbr(kode,gambar)
VALUES
('".$kodeKelompok."','".$files"')");
redirect('handlingFile');
}

}

```

### Keterangan:

Pada script `$kodeKelompok = "A002";` ini merupakan asumsi bahwa file yang akan kita upload tersimpan pada folder A002. Anggapannya ketika kita ingin menyimpan gambar berdasarkan kode tertentu pada masing-masing record.

`$path = 'images';` asumsi bahwa kita akan membuat folder baru dengan nama folder images

`if(!file_exists($path)){` untuk melakukan pengecekan apakah file pada variabel `($path)` sudah ada atau belum.

`mkdir` untuk membuat folder

`0777` untuk membuat hak akses pada folder dengan status Read Delete Update Add

`move_uploaded_file(` untuk melakukan proses upload gambar

## 9.3 Membuat Database

Untuk memulai eksekusi script di atas, terlebih dahulu kita tambahkan tabel dengan nama `tbl_gbr` dengan atribut seperti di bawah ini

Alter Table 'tbl_gbr' in 'fasilkom'								
	Field Name	Datatype	Len	Default	PK?	Not Null?	Unsigned?	Auto Incr?
*	kode	varchar	10		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	gambar	text			<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Maka akan tampil script tersebut dengan gambar seperti di bawah ini

The screenshot shows a web application interface. At the top, there's a header with a logo, a search bar containing 'localhost/PW2Ace/handlingFile', and a user profile. The main content area has a title 'Handling File' and a sub-section 'Hasil untuk "Handling File"'. Below this, there's a file input field with a 'Browse...' button and a message 'No files selected.'. At the bottom right, there's a 'Proses' button. On the left, there's a sidebar with a 'Dashboard' section and three menu items: 'Menu 1', 'Menu 2', and 'Menu 3'. At the very bottom, there's a footer with the text 'Wawan Gunawan, S.Kom., M.T. Application © 2018' and social media links.

## 9.4 Menampilkan Gambar

Selanjutnya untuk menampilkan data yang telah kita simpan, maka lakukan penambahan script pada folder views seperti pada script di bawah ini

```
<div class="form-group">
    <label class="col-sm-4 control-label no-padding-right">
file </label>
    <div class="col-sm-3">
        <input type="file" name="uploadfiles[]"
id="uploadfiles" multiple class="btn btn-info upload">
    </div>
</div>
<div class="form-group">
    <label class="col-sm-7 control-label no-padding-right">
```

```

        <button class="btn btn-white btn-info btn-bold"
id="btnSimpan" name="btnSimpan" type="submit"><i class="ace-icon fa fa-floppy-o bigger-120 blue"></i> Proses</button>
    </label>
</div>

</form>
</div>

<?php
foreach ($gambar as $row) {
    $myfiles = ($row->gambar) ? json_decode($row->gambar) : array();
    foreach($myfiles as $afайл) {
        $foto = "images/A002/".$afайл;
    }
}
?> gambar) ? json_decode($row->gambar) : array();` digunakan untuk melakukan proses penyimpanan record menjadi bernilai array

Selanjutnya tambahkan juga controllers pada file handlingFile.php seperti pada script di bawah ini

```
public function index()
{
    $kodeKelompok = "A002";
    $data['gambar'] = $this->db->query("select gambar from tbl_gbr
where kode='$kodeKelompok'")->result();
    json_encode($data);

    $this->load->view('header');
    $this->load->view('menu');
    $this->load->view('handlingFile',$data);
    $this->load->view('footer');
}
```

# LATIHAN

Buatlah tampilan gambar yang Kelompok Anda miliki pada tugas besar dalam website (Front End) kelompok Anda

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

## **Grafik**

| Fakultas               | Program Studi      | Tatap Muka | Kode MK    | Disusun Oleh                |
|------------------------|--------------------|------------|------------|-----------------------------|
| Fakultas Ilmu Komputer | Teknik Informatika | <b>10</b>  | W151700013 | Wawan Gunawan, S.Kom., M.T. |

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Sebelum melakukan pembuatan grafik, sebaiknya kita persiapkan terlebih dahulu menu Contoh Grafik pada folder views di file menu.php dengan menambahkan script seperti di bawah ini

```
<li class="">
    <a href="php echo site_url('handlingFile');?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Penanganan File
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;

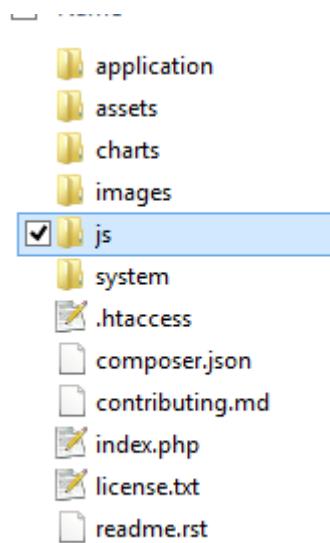
&lt;li class=""&gt;
    &lt;a href="<?php echo site_url('mahasiswa');?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Contoh Grafik
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;

    &lt;/ul&gt;
&lt;/li&gt;

&lt;li class=""&gt;
    &lt;a href="#" class="dropdown-toggle"&gt;
        &lt;i class="menu-icon fa fa-pencil-square-o"&gt;&lt;/i&gt;
        &lt;span class="menu-text"&gt; Menu 3 &lt;/span&gt;
        &lt;b class="arrow fa fa-angle-down"&gt;&lt;/b&gt;
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
</pre
```

Sebelum melanjutkan proses scripting maka persiapkan terlebih dahulu grafik yang akan kita gunakan, pada kesempatan ini kita akan menggunakan Fushion Chart dalam penanganan grafiknya. Simpan file fc\_colors\_helper pada folder application/helpers dan selanjutnya copy FushionCharts.php pada folder application/libraries.

Selanjutnya copy folder charts dan js sehingga akan tampil seperti gambar di bawah ini



Mahasiswa\_m utk models

## 9.1 Membuat Views

Selanjutnya untuk membuat untuk chart file, maka kita harus mempersiapkan beberapa hal.

Pastikan arahan file kita sesuai dengan folder-folder file tersebut. Script yang harus kita persiapkan seperti pada gambar di bawah ini dengan nama file nama mahasiswa\_chart\_v.php

```
<!-- HEADER TITLE -->
<div class="main-content">
    <div class="main-content-inner">

        <div class="breadcrumbs ace-save-state"
id="breadcrumbs">
            <ul class="breadcrumb">
                <li>
                    <i class="fa fa-desktop"></i>
                    <a href="#">Master Web</a>
                </li>
                <li class="active">Grafik</li>
            </ul><!-- /.breadcrumb -->
            <div class="nav-search" id="nav-search">
            </div><!-- /.nav-search -->
        </div>
        <div class="page-content">
            <div class="row">
                <div class="col-xs-12">
                    <div class="clearfix">
                        <h4 class="pink">
                            <i class="ace-icon fa fa-
```

```

fa-hand-o-right icon-animated-hand-pointer blue"></i>
                                Grafik
                <div class="pull-right
tableTools-container"></div>
                        </h4>
                </div>
                <div class="table-header">
                        Hasil untuk "Grafik"
                </div>
<!-- BATAS HEADER TITLE --&gt;
                &lt;div class="ln_solid"&gt;&lt;/div&gt;

<!-- PAGE CONTENT BEGINS --&gt;
                &lt;div class="modal-body"&gt;
                        &lt;form class="form-horizontal"
role="form" name="f_modal" id="f_modal" action="<?php echo
base_url('handlingFile/simpan'); ?&gt;" method="post"
enctype="multipart/form-data"&gt;
                                &lt;div class="form-
group"&gt;
                                    &lt;label
class="col-sm-4 control-label no-padding-right"&gt; Tahun &lt;/label&gt;
                                &lt;div class="col-
sm-3"&gt;
                                    &lt;select
name="cmbTahun" id="cmbTahun"
                                class="form-control" required data-parsley-error-
message="Field ini harus diisi"
                                data-
options="valueField:'id',textField:'text',
                                onChange:function(record){cariTahun()}&gt;
&lt;option value="0"&gt;-- pilih --&lt;/option&gt;
                                &lt;?php
                foreach ($tahun as $data) {
                    echo "&lt;option value='".$data[tahun].'"&gt;$data[tahun]&lt;/option&gt;";
                }
            ?&gt;
                                &lt;/select&gt;
                            &lt;/div&gt;
                            &lt;div class="col-
sm-3"&gt;
                                &lt;select
onChange="chart_onchange();" name="cmbChart" id="cmbChart"&gt;
                                    &lt;option value="Column2D" &lt;?PHP if($this-&gt;uri-&gt;segment(3) ==
'Column2D') echo 'selected'; ?&gt;&gt;Column 2D&lt;/option&gt;
                                    &lt;option value="Column3D" &lt;?PHP if($this-&gt;uri-&gt;segment(3) ==
</pre>

```

```

'Column3D') echo 'selected'; ?>>Column 3D</option>

    <option value="Line" <?PHP if($this->uri->segment(3) ==
'Line') echo 'selected'; ?>>Line</option>

    <option value="Pie2D" <?PHP if($this->uri->segment(3) ==
'Pie2D') echo 'selected'; ?>>Pie 2D</option>

    <option value="Pie3D" <?PHP if($this->uri->segment(3) ==
'Pie3D') echo 'selected'; ?>>Pie 3D</option>

    <option value="Bar2D" <?PHP if($this->uri->segment(3) ==
'Bar2D') echo 'selected'; ?>>Bar</option>

    <option value="Area2D" <?PHP if($this->uri->segment(3) ==
'Area2D') echo 'selected'; ?>>Area</option>

    <option value="Doughnut2D" <?PHP if($this->uri->segment(3)
== 'Doughnut2D') echo 'selected'; ?>>Doughnut</option>

    <option value="Funnel" <?PHP if($this->uri->segment(3) ==
'Funnel') echo 'selected'; ?>>Funnel</option>
                                    </select>

                                </div>

                                </div>

                                <div class="form-
group">
    <?PHP echo
$chart; ?>
    </div>

                                </form>
                            </div>
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

## 9.2 Membuat Controllers

Buat controller mahasiswa.php dengan script seperti di bawah ini

```

<?PHP
    class Mahasiswa extends CI_Controller
    {
        public function __construct()
        {
            parent::__construct();

            $this->load->database();
            $this->load->helper('url');
            $this->load->helper('FC_Colors');
            $this->load->model('mahasiswa_m');

```

```

        }

    public function index()
    {
        $this->load->view('header');
        $this->load->view('menu');
        $this->chart();
        $this->load->view('footer');
    }

public function cariTahun()
{
    $this->mahasiswa_m->view_jenis_kelamin();
}

public function chart()
{
    $this->load->library('FusionCharts');

    if($this->uri->segment(3) == '')
        $chartType = 'Column2D';
    else
        $chartType = $this->uri->segment(3);

    $width = '600';
    $height = '300';

    $chart = new FusionCharts($chartType, $width,
    $height);

    $caption = 'Grafik Mahasiswa';
    $xAxisName = 'Jenis Kelamin';
    $yAxisName = 'Jumlah';
    $decimalPrecision = '0';
    $formatNumberScale = '0';
    $showNames = '1';

    $data = $this->mahasiswa_m-
>view_jenis_kelamin();

    $strXML = "
<graph caption='".$caption."'
xAxisName='".$xAxisName."' yAxisName='".$yAxisName."'
decimalPrecision='".$decimalPrecision."'
formatNumberScale='".$formatNumberScale."'>
";
    $a=0;
    foreach ($data as $group) {
        $color =
cari_warna("$a");
        $strXML .= "<set
name='".$group[jenis_kelamin]' value='".$group[sex]' color='".$color'
/>";
        $a++;
    }
    $strXML .= "</graph>";
    $data['tahun']=$this->mahasiswa_m->cariTahun();
}

```

```

        $data['chart'] = $chart-
>renderChartHTML(base_url().'charts/FCF_'. $chartType.'.swf', '',
$strXML, 'chartId', $width, $height);
        $this->load->view('mahasiswa_chart_v', $data);
    }
}

?>

```

### 9.3 Membuat Model

Selanjutnya untuk menampilkan data yang telah kita simpan, maka lakukan penambahan script pada folder models dengan nama file mahasiswa\_m.php seperti pada script di bawah ini

```

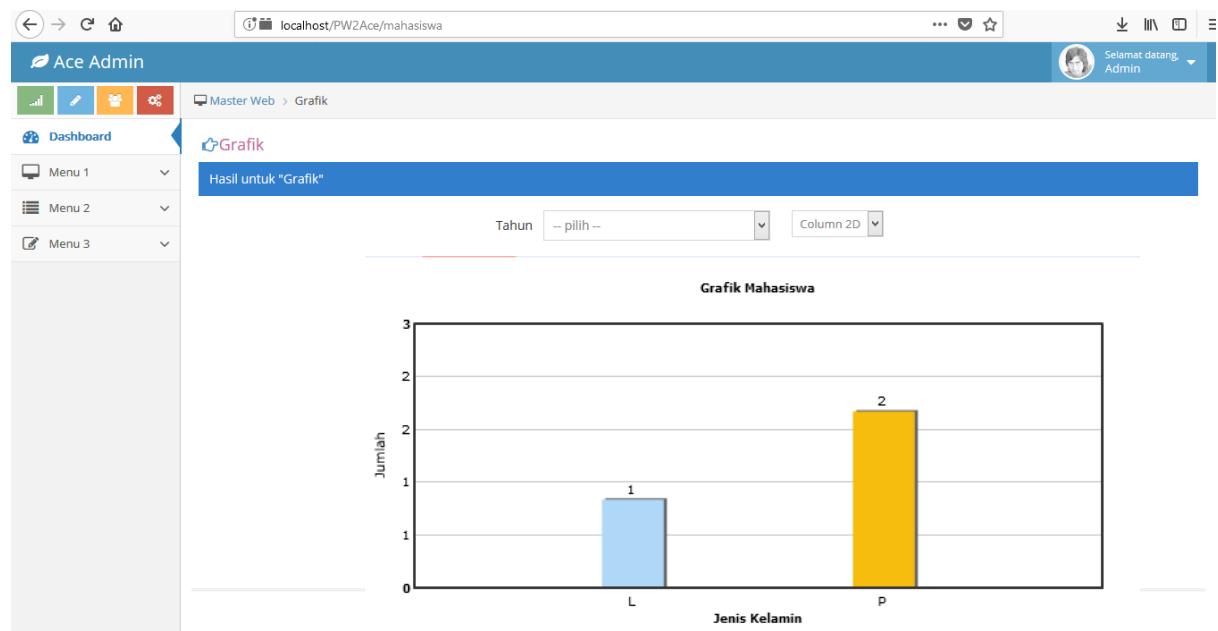
<?PHP
class Mahasiswa_M extends CI_Model
{
    public function view_all()
    {
        $sql = "select * from tbl_mahasiswa";
        return $this->db->query($sql);
    }

    public function view_jenis_kelamin()
    {
        $tahun      = $this->input->post('tahun');
        $cari="";
        if (!empty($tahun)) {
            $cari = "where tahun=$tahun";
        }
        $sql="select jenis_kelamin,count(jenis_kelamin) as sex
from tbl_mahasiswa
                $cari group by jenis_kelamin";
        $query = $this->db->query($sql);
        return $query -> result_array();
    }

    public function cariTahun()
    {
        $sql="select distinct(tahun) as tahun from
tbl_mahasiswa";
        $tahun = $this->db->query($sql);
        return $tahun->result_array();
    }
}
?>

```

## Tampilan yang diharapkan



# LATIHAN

Buatlah tampilan grafik yang Kelompok Anda miliki pada tugas besar dalam Back End kelompok Anda

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

## **AJAX**

| Fakultas               | Program Studi      | Tatap Muka | Kode MK    | Disusun Oleh                |
|------------------------|--------------------|------------|------------|-----------------------------|
| Fakultas Ilmu Komputer | Teknik Informatika | <b>11</b>  | W151700013 | Wawan Gunawan, S.Kom., M.T. |

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Sebelum melakukan pembuatan latihan Ajax, sebaiknya kita persiapkan terlebih dahulu menu Belajar Ajax pada folder views di file menu.php dengan menambahkan script seperti di bawah ini

```
<li class="">
    <a href="php echo site_url('mahasiswa');?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Contoh Grafik
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;

&lt;li class=""&gt;
    &lt;a href="<?php echo site_url('ajax');?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Belajar Ajax
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;

&lt;/ul&gt;
&lt;/li&gt;

&lt;li class=""&gt;
    &lt;a href="#" class="dropdown-toggle"&gt;
        &lt;i class="menu-icon fa fa-pencil-square-o"&gt;&lt;/i&gt;
        &lt;span class="menu-text"&gt; Menu 3 &lt;/span&gt;
        &lt;b class="arrow fa fa-angle-down"&gt;&lt;/b&gt;
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;</pre
```

Sebelum melanjutkan proses scripting maka persiapkan terlebih dahulu beberapa file seperti di bawah ini

## 9.1 Membuat Views

Selanjutnya untuk membuat view maka kita harus mempersiapkan beberapa hal.

Pastikan arahan file kita sesuai dengan folder-folder file tersebut. Script yang harus kita persiapkan seperti pada gambar di bawah ini dengan nama file nama ajax\_view.php

```
<!-- HEADER TITLE -->
<div class="main-content">
    <div class="main-content-inner">

        <div class="breadcrumbs ace-save-state"
id="breadcrumbs">
            <ul class="breadcrumb">
```

```
<li>
    <i class="fa fa-desktop"></i>
    <a href="#">Master Web</a>
</li>
<li class="active">Ajax</li>
</ul><!-- /.breadcrumb -->
<div class="nav-search" id="nav-search">
</div><!-- /.nav-search -->
</div>
<div class="page-content">
    <div class="row">
        <div class="col-xs-12">
            <div class="clearfix">
                <h4 class="pink">
                    <i class="ace-icon fa fa-hand-o-right icon-animated-hand-pointer blue"></i>
                    Belajar Ajax
                <div class="pull-right"
tableTools-container"></div>
                    </h4>
                </div>
                <div class="table-header">
                    Hasil untuk "Belajar
Ajax"
                </div>
                <!-- BATAS HEADER TITLE -->
                <div class="ln_solid"></div>
                <!-- PAGE CONTENT BEGINS -->
                <div class="modal-body">
                    <form class="form-horizontal"
role="form" name="f_modal" id="f_modal" action="php echo
base_url('subGroupProduct/simpan'); ?&gt;" method="post"
enctype="multipart/form-data"&gt;
                        &lt;div class="form-
group"&gt;
                            &lt;label
class="col-sm-4 control-label no-padding-right"&gt; Kelompok Produk
&lt;/label&gt;
                            &lt;div class="col-
sm-3"&gt;
                                &lt;select
name="cmbGroup" id="cmbGroup" onchange="cariSub()"&gt;
                                    &lt;option value=""&gt;&lt;/option&gt;
                                    &lt;?php
  foreach ($data as $row) {
  echo "&lt;option value='".$row-&gt;GroupProduct_auto'&gt;$row-
&gt;GroupProduct_Name&lt;/option&gt;";
  }
                                    ?&gt;
                                &lt;/select&gt;
                            &lt;/div&gt;
                        &lt;/div&gt;
                    &lt;/form&gt;
                &lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre
```

```

                </select>
            </div>
        </div>
        <div class="form-
group">
            <label
class="col-sm-4 control-label no-padding-right"> Sub Kelompok
Produk </label>
            <div class="col-
sm-3">
                <select
name="cmbSubGroup" id="cmbSubGroup"

                    class="form-control" required data-parsley-error-
message="Field ini harus diisi">

                    <option value="" selected>-- Pilih --</option>
                </select>
            </div>
        </div>

        <div class="form-
group">
            <label
class="col-sm-7 control-label no-padding-right">
                <button
class="btn btn-white btn-info btn-bold" id="btnSimpan"
name="btnSimpan" type="submit"><i class="ace-icon fa fa-floppy-o
bigger-120 blue"></i> Simpan</button>
                <a
href="<?php echo base_url(); ?>subGroupProduct" target="_self"
class="btn btn-white btn-default btn-round" data-dismiss="modal"
aria-hidden="true"><i class="fa fa-minus-circle"></i> Tutup</a>
            </label>
        </div>

        </form>
    </div>
</div>
</div>
</div>
</div>

```

## 9.2 Membuat Controllers

Buat controller ajax.php dengan script seperti di bawah ini

```

<?php if(! defined('BASEPATH')) exit('No direct script access
allowed');

class ajax extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
    }
}

```

```

        $this->load->helper('url');
        $this->load->model('groupProduct_model');
    }

    public function index()
    {
        $data['data'] = $this->groupProduct_model->getData();
        json_encode($data);
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('ajax_view', $data);
        $this->load->view('footer');
    }
}

```

### 9.3 Membuat Model

Selanjutnya untuk menampilkan data yang telah kita simpan, maka lakukan penambahan script pada folder models dengan nama file groupProduct.php seperti pada script di bawah ini yang diberi warna merah

```

public function edit()
{
    $kodeGroup=$this->input->post('id');
    $query = $this->db->query("SELECT * from m_groupproduct where
groupproduct_auto='$kodeGroup'");
    return $query->result();
}

public function getData()
{
    $groupproduct = $this->db->query("SELECT * from
m_groupproduct");
    return $groupproduct->result();
}

public function cariSubData($id)
{
    $groupproduct = $this->db->query("SELECT * from
m_subgroupproduct where ssub_groupid='$id'");
    return $groupproduct->result();
}

}

```

## Tampilan yang diharapkan

Hasil untuk "Belajar Ajax"

Kelompok Produk

Sub Kelompok Produk

Simpan Tutup

Wawan Gunawan, S.Kom., M.T. Application © 2018 [Twitter](#) [Facebook](#) [RSS](#)

Selanjutnya tambahkan script untuk proses ajax pada view dan letakan pada posisi paling bawah dengan script seperti di bawah ini

```
<script type="text/javascript">
    function cariSub(){
        var id=document.getElementById('cmbGroup').value;
        $.ajax({
            url : "<?php echo base_url();?>ajax/get_sub_groupproduct",
            method : "POST",
            data : {id: id},
            async : false,
            dataType : 'json',
            success: function(data){
                var html = '';
                var i;
                html += '<option>-- Pilih --</option>';
                for(i=0; i<data.length; i++){
                    html += 
'<option>' + data[i].sSub_Name + '</option>';
                }
                $('#cmbSubGroup').html(html);
                i--;
            }
        });
        document.getElementById('txtKodeProduk').value=data[i].sSub_Name;
    }
</script>
```

Selanjutnya tambahkan script pada controller ajax.php dengan tambahan script seperti di bawah ini dengan penambahan pada diberi warna merah di bawah ini

```
<?php if(! defined('BASEPATH')) exit('No direct script access allowed');

class ajax extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('groupProduct_model');
    }

    public function index()
    {
        $data['data'] = $this->groupProduct_model->getData();
        json_encode($data);
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('ajax view', $data);
        $this->load->view('footer');
    }

    function get_sub_groupproduct(){
        $id=$this->input->post('id');
        $data= $this->groupProduct_model->cariSubData($id);
        echo json_encode($data);
    }
}
```

Selanjutnya akan tampil seperti pada gambar di bawah ini

The screenshot shows a web application interface. At the top right, there is a user profile icon with the text "Selamat datang, Admin". Below the header, a breadcrumb navigation bar shows "Master Web > Ajax". A pink ribbon-like banner displays the text "Belajar Ajax". The main content area has a blue header bar with the text "Hasil untuk 'Belajar Ajax'".

The form contains two dropdown menus:

- Kelompok Produk:** A dropdown menu set to "Alat Tulis".
- Sub Kelompok Produk:** A dropdown menu currently showing "Alat Tulis" (selected), "Correction Tape", and "Pulpen Mickey".

At the bottom of the form are two buttons: "Simpan" (Save) and "Tutup" (Close).

Wawan Gunawan, S.Kom., M.T. Application © 2018 [Twitter](#) [Facebook](#) [RSS](#)

This screenshot shows the same web application interface as the previous one, but with a different state of the dropdown menu.

The "Sub Kelompok Produk" dropdown menu is now empty, showing only the placeholder text "-- Pilih --".

At the bottom of the form are two buttons: "Simpan" (Save) and "Tutup" (Close).

Wawan Gunawan, S.Kom., M.T. Application © 2018 [Twitter](#) [Facebook](#) [RSS](#)

# LATIHAN

Buatlah tampilan grafik yang Kelompok Anda miliki pada tugas besar dalam Back End kelompok Anda

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

## **Konversi Data**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>12</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Sebelum melakukan pembuatan laporan, sebaiknya kita persiapkan terlebih dahulu menu Laporan Group Produk pada folder views di file menu.php dengan menambahkan script seperti di bawah ini

```
<li class="">
    <a href="#" class="dropdown-toggle">
        <i class="menu-icon fa fa-list"></i>
        <span class="menu-text"> Menu 2 </span>
        <b class="arrow fa fa-angle-down"></b>
    </a>
    <b class="arrow"></b>
    <ul class="submenu">

<li class="">
    <a href="php echo site_url('laporan'); ?&gt;"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Laporan
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;

&lt;li class=""&gt;
    &lt;a href="#"&gt;
        &lt;i class="menu-icon fa fa-caret-right"&gt;&lt;/i&gt;
        Sub Menu 2.2
    &lt;/a&gt;
    &lt;b class="arrow"&gt;&lt;/b&gt;
&lt;/li&gt;
&lt;/ul&gt;
&lt;/li&gt;</pre
```

## 9.1 Membuat Views

Selanjutnya untuk membuat konversi data ke Excel dan juga ke Pdf maka kita harus mempersiapkan beberapa hal.

Pastikan arahan file kita sesuai dengan folder-folder file tersebut. Script yang harus kita persiapkan seperti pada gambar di bawah ini

```
<!-- HEADER TITLE -->
<div class="main-content">
    <div class="main-content-inner">

        <div class="breadcrumbs ace-save-state" id="breadcrumbs">
            <ul class="breadcrumb">
                <li>
```

```

        <i class="fa fa-desktop"></i>
        <a href="#">Master Web</a>
    </li>
    <li class="active">Laporan</li>
</ul><!-- /.breadcrumb -->
<div class="nav-search" id="nav-search">
</div><!-- /.nav-search -->
</div>
<div class="page-content">
    <div class="row">
        <div class="col-xs-12">
            <div class="clearfix">
                <h4 class="pink">
                    <i class="ace-icon fa fa-hand-o-right icon-animated-hand-pointer blue"></i>
                    Laporan
                    <div class="pull-right tableTools-container"></div>
                </h4>
            </div>
            <div class="table-header">
                Hasil untuk "Laporan"
            </div>
            <!-- BATAS HEADER TITLE -->
            <div class="ln_solid"></div>

            <!-- PAGE CONTENT BEGINS -->
            <div class="modal-body">
                <form class="form-horizontal" role="form"
name="f_modal" id="f_modal" action=<?php echo
base_url('laporan/proses'); ?>" method="post"
enctype="multipart/form-data">
                    <div class="form-group">
                        <label class="col-sm-4 control-
label no-padding-right"> Periode Laporan </label>
                        <div class="col-sm-3">
                            <input type="date"
id="tglawal" class="form-control" />
                        </div>
                        <div class="col-sm-1">
                            s.d
                        </div>
                        <div class="col-sm-3">
                            <input type="date"
id="tglakhir" class="form-control" />
                        </div>
                    </div>

                    <div class="form-group">
                        <label class="col-sm-7 control-
label no-padding-right">
                            <button class="btn btn-white
btn-info btn-bold" id="btnSimpan" name="btnSimpan"
type="submit"><i class="ace-icon fa fa-floppy-o bigger-120
blue"></i> Proses</button>
                            <a href=<?php echo
base_url(); ?>laporan" target="_self" class="btn btn-white btn-
default btn-round" data-dismiss="modal" aria-hidden="true"><i
class="fa fa-minus-circle"></i> Tutup</a>
                        </label>
                    </div>
                </form>
            </div>
        </div>
    </div>
</div>

```

```

        </form>
    </div>
</div>
</div>
</div>
</div>

```

## 9.2 Membuat Controllers

Buat controller laporan.php dengan script seperti gambar di bawah ini

```

<?php if(! defined('BASEPATH')) exit('No direct script access
allowed');

class laporan extends CI_Controller {
    public function __construct()
    {
        parent::__construct();
        $this->load->helper('url');
        $this->load->model('laporan_model');
    }

    public function index()
    {
        $this->load->view('header');
        $this->load->view('menu');
        $this->load->view('laporan_view');
        $this->load->view('footer');
    }

    public function proses()
    {
        //load plugin
        include APPPATH.'third_party/PHPExcel/PHPExcel.php';

        //panggil class
        $excel = new PHPExcel();

        //setting file excel
        $excel->getProperties()->setCreator('Ryan Pratama')
            ->setLastModifiedBy('Ryan Pratama')
            ->setTitle("Group Produk")
            ->setSubject("Group Produk")
            ->setDescription("Laporan Group Produk");

        // Buat sebuah variabel untuk menampung pengaturan style dari
        // header tabel
        $style_col = array(
            'font' => array('bold' => true), // Set font nya jadi
            bold
            'alignment' => array(
                'horizontal' =>
                PHPExcel_Style_Alignment::HORIZONTAL_CENTER, // Set text center
                'vertical' =>
                PHPExcel_Style_Alignment::VERTICAL_CENTER // Set text secara vertical

```

```

(middle)
    ),
    'borders' => array(
        'top' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN), // Set border top dengan garis
tipis
        'right' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN), // Set border right dengan
garis tipis
        'bottom' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN), // Set border bottom dengan
garis tipis
        'left' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN) // Set border left dengan garis
tipis
    )
);

// Buat sebuah variabel untuk menampung pengaturan style dari
isi tabel
$style_row = array(
    'alignment' => array(
        'vertical' =>
PHPExcel_Style_Alignment::VERTICAL_CENTER // Set text secara vertical
(middle)
    ),
    'borders' => array(
        'top' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN), // Set border top dengan garis
tipis
        'right' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN), // Set border right dengan
garis tipis
        'bottom' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN), // Set border bottom dengan
garis tipis
        'left' => array('style' =>
PHPExcel_Style_Border::BORDER_THIN) // Set border left dengan garis
tipis
    )
);

$excel->setActiveSheetIndex(0)->setCellValue('A1', "DATA
GROUP"); // Set kolom A1 dengan tulisan "DATA GROUP"
$excel->getActiveSheet()->mergeCells('A1:C1'); // Set Merge
Cell pada kolom A1 sampai C1
$excel->getActiveSheet()->getStyle('A1')->getFont()-
>setBold(TRUE); // Set bold kolom A1
$excel->getActiveSheet()->getStyle('A1')->getFont()-
>setSize(15); // Set font size 15 untuk kolom A1
// Set text center untuk kolom A1
$excel->getActiveSheet()->getStyle('A1')->getAlignment()-
>setHorizontal(PHPExcel_Style_Alignment::HORIZONTAL_CENTER);

// Buat header tabel nya pada baris ke 3
$excel->setActiveSheetIndex(0)->setCellValue('A3', "NO"); //
Set kolom A3 dengan tulisan "NO"
$excel->setActiveSheetIndex(0)->setCellValue('B3', "KODE");
// Set kolom B3 dengan tulisan "KODE"
$excel->setActiveSheetIndex(0)->setCellValue('C3', "NAMA");
// Set kolom C3 dengan tulisan "NAMA"

```

```

        // Apply style header yang telah kita buat tadi ke masing-
masing kolom header
        $excel->getActiveSheet()->getStyle('A3')-
>applyFromArray($style_col);
        $excel->getActiveSheet()->getStyle('B3')-
>applyFromArray($style_col);
        $excel->getActiveSheet()->getStyle('C3')-
>applyFromArray($style_col);

        // Panggil function rptView yang ada di rptGroupModel
        $groupModel = $this->laporan_model->rptView();
        $no = 1;           // Untuk penomoran tabel, di awal set dengan
1
        $numrow = 4;      // Set baris pertama untuk isi tabel adalah
baris ke 4
        foreach($groupModel as $data){ // Lakukan looping pada
variabel siswa
            $excel->setActiveSheetIndex(0)->setCellValue('A' . $numrow,
$numrow);
            $excel->setActiveSheetIndex(0)->setCellValue('B' . $numrow,
$data->GroupProduct_auto);
            $excel->setActiveSheetIndex(0)->setCellValue('C' . $numrow,
$data->GroupProduct_Name);

            // Apply style row yang telah kita buat tadi ke masing-
masing baris (isi tabel)
            $excel->getActiveSheet()->getStyle('A' . $numrow)-
>applyFromArray($style_row);
            $excel->getActiveSheet()->getStyle('B' . $numrow)-
>applyFromArray($style_row);
            $excel->getActiveSheet()->getStyle('C' . $numrow)-
>applyFromArray($style_row);

            $no++; // Tambah 1 setiap kali looping
            $numrow++; // Tambah 1 setiap kali looping
        }
        // Set width kolom
        $excel->getActiveSheet()->getColumnDimension('A')-
>setWidth(5); // Set width kolom A
        $excel->getActiveSheet()->getColumnDimension('B')-
>setWidth(15); // Set width kolom B
        $excel->getActiveSheet()->getColumnDimension('C')-
>setWidth(25); // Set width kolom C

        // Set height semua kolom menjadi auto (mengikuti height isi
dari kolomnya, jadi otomatis)
        $excel->getActiveSheet()->getDefaultRowDimension()-
>setRowHeight(-1);
        // Set orientasi kertas jadi LANDSCAPE
        $excel->getActiveSheet()->getPageSetup()-
>setOrientation(PHPExcel_Worksheet_PageSetup::ORIENTATION_LANDSCAPE);

        // Set judul file excel nya
        $excel->getActiveSheet(0)->setTitle("Laporan Group Product");
        $excel->setActiveSheetIndex(0);

        // Proses file excel
        header('Content-Type: application/vnd.openxmlformats-
officedocument.spreadsheetml.sheet');
        header('Content-Disposition: attachment; filename="Data

```

```

Group" );
}

// Set nama file excel nya
header('Cache-Control: max-age=0');
//PHPExcel_Settings::setZipClass(PHPExcel_Settings::PCLZIP);
//zip pada php ini diaktifkan
$write = PHPExcel_IOFactory::createWriter($excel,
'Excel2007');
$write->save('php://output');
}
}
}

```

### 9.3 Membuat Models

Selanjutnya kita tambahkan pada model laporan\_model.php dengan script seperti dibawah ini

```

<?php if ( ! defined('BASEPATH')) exit('No direct script access
allowed');

class laporan_model extends CI_Model {

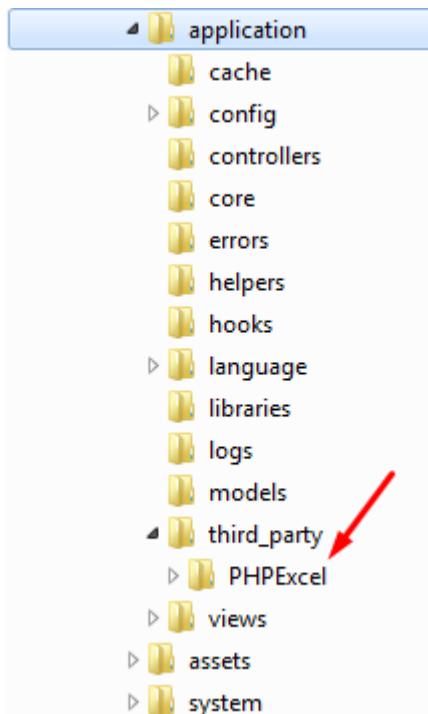
    public function __construct()
    {
        parent::__construct();
        $this->load->database();
    }

    public function rptView()
    {
        $sql = "SELECT * FROM m_groupproduct WHERE l=1 ";
        //Tampilkan semua data yang ada di tabel
        return $this->db->query($sql)->result();
    }
}

```

## 9.4 Export Excel

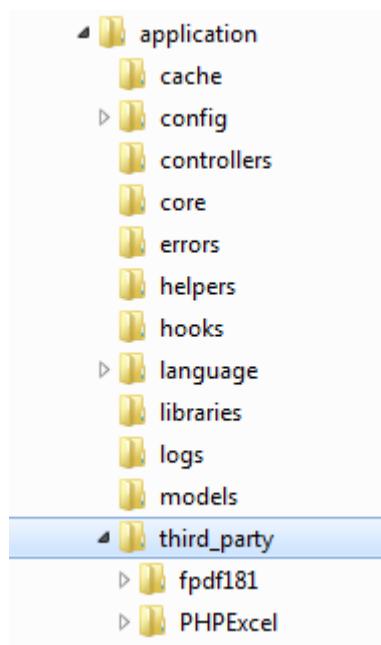
Agar pembuatan export data ke excel dapat berjalan dengan baik, maka kita harus mempersiapkan terlebih dahulu plugin PHPExcel yang dapat kita unduh di <https://github.com/PHPOffice/PHPExcel> dan selanjutnya kita extrak hasil download file tersebut ke dalam folder application/third\_party/ dan akan membentuk struktur seperti gambar di bawah ini



Gambar 0.1 Tampilan library PHPExcel

## 9.5 Export Pdf

Sebelum melakukan pembuatan export data ke Pdf, maka kita harus mempersiapkan terlebih dahulu plugin Pdf yang dapat kita unduh di [www.fpdf.org/en/dl.php?v=181&f=zip](http://www.fpdf.org/en/dl.php?v=181&f=zip) dan selanjutnya kita extrak hasil download file tersebut ke dalam folder application/third\_party/ dan akan membentuk struktur seperti gambar di bawah ini



**Gambar 0.2 Tampilan File third\_party fpdf181**

Selanjutnya tambahkan function exportPdf pada controller laporan.php dengan script seperti di bawah ini

```
public function exportPDF() {  
    include APPPATH.'third_party/fpdf181/fpdf.php';  
  
    $pdf = new FPDF('l', 'mm', 'A5');  
    $pdf->AddPage();  
  
    $pdf->SetFont('Arial', 'B', 16);  
    $pdf->Cell(190, 7, 'Laporan Data Group', 0, 1, 'C');  
  
    $pdf->SetFont('Arial', 'B', 12);  
    $pdf->Cell(190, 7, 'PT XYZ', 0, 1, 'C');  
  
    $pdf->Cell(10, 7, '', 0, 1);  
  
    $pdf->SetFont('Arial', 'B', 10);  
    $pdf->Cell(20, 6, 'No', 1, 0, 'C');  
    $pdf->Cell(27, 6, 'Kode', 1, 0, 'C');  
    $pdf->Cell(85, 6, 'Keterangan', 1, 0, 'C');  
  
    $pdf->SetFont('Arial', '', 10);
```

```
$groupModel = $this->laporan_model->rptView();
$no = 1;
foreach ($groupModel as $data) {
    $pdf->Cell(10, 6, '', 0, 1);
    $pdf->Cell(20, 6, $no, 1, 0);
    $pdf->Cell(27, 6, $data->GroupProduct_auto, 1, 0);
    $pdf->Cell(85, 6, $data->GroupProduct_Name, 1, 0);

    $no++;
}

$pdf->Output();
```

# LATIHAN

Buatlah laporan dari studi kasus sebelumnya yang mengambil periode tanggal dan juga pilihan menggunakan combobox

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

## **Template Engine SMARTY**

<b>Fakultas</b>	<b>Program Studi</b>	<b>Tatap Muka</b>	<b>Kode MK</b>	<b>Disusun Oleh</b>
Fakultas Ilmu Komputer	Teknik Informatika	<b>13</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Dennings memberi catatan khusus untuk bidang BioInformatik sebagai bidang baru yang merupakan gabungan antara Ilmu Komputer dan Biologi, dan saat ini mengalami perkembangan yang cukup signifikan.

Kemudian seiring dengan peningkatan ilmu dan teknologi, ada kemungkinan matriks ini akan mengalami perbaikan lagi di kelak kemudian hari. Baris dalam matriks Denning ini menggambarkan bidang-bidang dalam ilmu komputer. Sedangkan kolom pada matriks menggambarkan paradigma bidang-bidang tersebut, yang direfleksikan dalam tiga hal: Teori (Theory), Abstraksi (Abstraction), dan Desain (Design).

Teori: adalah berlandaskan pada pendekatan matematika, dimana untuk mendapatkan suatu teori yang valid, harus melalui proses-proses sbb:

- pendefinisian (definition)
- pembuatan teorema (theorem)
- pembuktian (proof)
- penginterpretasian hasil (interpret result)

Abstraksi: atau pemodelan (modeling), adalah berlandaskan pada metode eksperimen ilmiah, dimana dalam melakukan invesitigasi terhadap suatu fenomena, harus melalui proses-proses sbb:

- membentuk hipotesa (form a hypothesis)
- membuat suatu model dan melakukan prediksi (construct a model and make a prediction)
- mendesain eksperimen dan mengumpulkan data (design an experiment and collect data)
- menganalisa hasil (analyze result)

Desain: adalah berlandaskan pada pendekatann engineering (teknik), dimana pada saat mendesain sebuah sistem atau device untuk memecahkan masalah, harus melalui prosesproses sebagai berikut:

- menyatakan requirement (state requirements),
- menyatakan spesifikasi (state specifications)
- melakukan desain dan implementasi sistem (design and implement the system)

- melakukan pengetesan terhadap sistem (test the system)

Dari penjelasan diatas, bisa kita pahami bahwa yang bergerak dalam masalah penelitian ilmu komputer akan banyak berhubungan dengan dua kolom pertama matriks (Teori dan Abstraksi). Sedangkan yang bergerak dalam masalah yang lebih teknis dengan memakai pendekatan engineering,

akan lebih banyak berkecimpung dalam ruang lingkup dua kolom terakhir matriks (Abstraksi dan Desain).

Istilah ontology berasal dari Yunani ontos, untuk keberadaan, dan logos, untuk kata. Dalam filsafat, ontology digunakan sebagai subbidang dari filsafat, yang mempelajari sifat alami dari keberadaan, cabang dari metafisik yang berkaitan dengan identifikasi, atau secara umum, jenis-jenis benda yang secara actual ada, dan bagaimana memaparkannya. Sebagai contoh, observasi yang dilakukan pada objek tertentu yang mengelompokkan menjadi kelas-kelas abstrak berdasarkan pada sifat-sifat bersama merupakan komitmen ontology secara tipe.

Adabanyak definisi dari konsep ontology dalam AI dan dalam komputasi secara umum. Yang paling banyak dikutip adalah :

“Ontologi adalah spesifikasi dari sebuah konseptualisasi” (Gru1993) Definisi adalah definisi yang paling ringkas dan membutuhkan beberapa klarifikasi. Konseptualisasi berarti sebuah abstrak, pandangan dari dunia yang disederhanakan. Jika sebuah basis pengetahuan dari sebuah sistem cerdas harus merepresentasikan dunia untuk beberapa tujuan, maka basis pengetahuan itu harus melakukan beberapa konseptualisasi. Setiap konseptualisasi didasarkan pada konsep-konsep, obyek-obyek, dan entitas-entitas lain yang diasumsikan ada dalam beberapa area yang menarik dan hubungan yang dipenuhi diantara mereka.

Spesifikasi berarti representasi secara formal dan deklaratif. Dalam struktur data yang merepresentasikan sebuah ontology, tipe dari konsep yang digunakan dan batasan penggunaan dinyatakan secara deklaratif, eksplisit dan menggunakan bahasa formal tertentu. Representasi formal berimplikasi bahwa ontologi seharusnya dapat dibaca mesin. Namun demikian, ontologi tidaklah aktif; ontologi tidak dapat dijalankan seperti program. Ontologi secara deklaratif merepresentasikan beberapa pengetahuan untuk digunakan oleh program yang berbeda.

Ontology adalah himpunan istilah-istilah pengetahuan, termasuk kosa kata, interkoneksi semantik dan beberapa kaidah inferensi dan logika untuk beberapa topik tertentu.

Bagian penting dari definisi Hendler adalah interkoneksi semantik dan inferensi dan logika. Interkoneksi semantik berarti ontologi menspesifikasikan arti dari relas-relasi diantara konsep-konsep yang digunakan. Hal ini juga dapat diinterpretasikan sebagai saran bahwa ontologi sendiri diinterkoneksi dengan baik; sebagai contoh, ontologi tangan dan lengan mungkin dibangun secara logika, secara semantik dan secara formal terinterkoneksi.

Inferensi dan logika berarti bahwa ontologi membolehkan untuk melakukan beberapa penalaran. Sebagai contoh, ontologi musisi akan menspesifikasikan musisi dengan instrumen yang digunakan untuk bermain. Dengan menggunakan kategori ini, memungkinkan menggunakan sistem cerdas untuk melakukan penalaran, sebagai contoh, pianis adalah musisi.

Dalam dunia Semantic Web, definisi operasional ontologi dari W3C OWL Requirements Documents :

Sebuah ontologi mendefinisikan istilah-istilah yang digunakan untuk memaparkan dan merepresentasikan sebuah area pengetahuan.

Ada beberapa aspek dari definisi ini yang perlu diklarifikasi. Yang pertama, definisi ini menyatakan bahwa ontologi digunakan untuk memaparkan dan merepresentasikan sebuah area pengetahuan. Dengan kata lain, ontologi adalah khusus pada suatu domain; ontology tidak untuk merepresentasikan semua pengetahuan, tetapi sebuah area pengetahuan. Sebuah domain adalah penyederhanaan area subyek tertentu, seperti fotografi, medis, real estate, pendidikan dan lain-lain.

Kedua, ontology memuat istilah-istilah dan hubungan diantara istilah-istilah. Istilah-istilah sering disebut kelas atau konsep;kata-kata ini dapat saling dipertukarkan. Hubungan diantara kelas-kelas dapat diekspresikan dengan menggunakan struktur hirarki: kelas super merepresentasikan konsep dengan level lebih tinggi dan subkelas merepresentasikan konsep yang lebih tajam, dan semua subkelas mempunyai semua atribut dan fitur yang dimiliki konsep yang lebih tinggi.

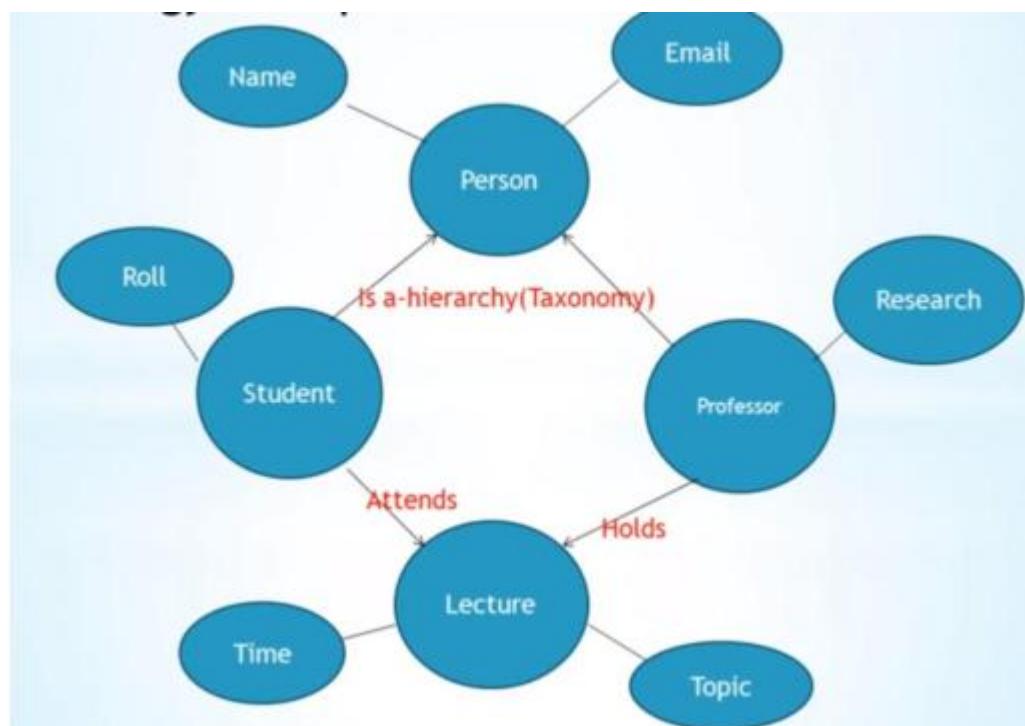
Ketiga, disamping hubungan diantara kelas, ada tingkat hubungan yang diekspresikan dengan menggunakan kelompok istilah tertentu : properti. Istilah properti ini memaparkan berbagai fitur dan atribut dari konsep dan mereka juga dapat digunakan untuk mengasosiasikan kelas-kelas berbeda menjadi bersama. Oleh karena itu, hubungan diantara kelas-kelas tidak hanya hubungan kelas super dan subkelas, tetapi juga hubungan yang diekspresikan dalam bentuk properti.

Dengan mendefinisikan secara jelas istilah-istilah dan hubungan diantara mereka, sebuah ontologi mengkodekan pengetahuan dari domain sedemikian rupa sehingga ontology dapat dimengerti oleh sebuah komputer. Ini adalah tujuan dasar dari sebuah ontology.

Dalam dunia Semantic Web, ditemukan konsep yang lain : taksonomi. Taksonomi dan ontology sering sekali digunakan secara bergantian; namun demikian, keduanya merupakan konsep yang berbeda. Ontologi mendefinisikan tidak hanya kelas-kelas tetapi juga propertinya. Ontologi selanjutnya mengindikasikan tipe dari harga properti yang dipunyai dan kelas apa yang berasosiasi.

Di sisi lain, taksonomi terutama memperhatikan klasifikasi – tidak property – untuk mengekspresikan batasan dan hubungan. Taksonomi adalah klasifikasi berdasarkan subyek yang menyusun istilah-istilah menjadi sebuah hirarki. Hirarki dari taksonomi memuat hubungan orang tua – anak, seperti “subkelas dari” atau “superkelas dari”.

Sebuah ontology terdiri dari sebuah daftar istilah terbatas dan hubungan diantara istilah-istilah ini. Istilah menandakan pentingnya konsep (kelas dari objek) dari suatu domain. Sebagai contoh, diperlihatkan hirarki klasifikasi Camera. Hubungan (relationship) mencakup hirarki dari kelas-kelas. Sebuah hirarki menspesifikasikan sebuah kelas menjadi subkelas C“ yang lain jika setiap objek dalam C juga termasuk dalam C“. Sebagai contoh, dalam sebuah hirarki untuk domain universitas, staff akademik, staff administrasi dan staff teknisi termasuk dalam pegawai.



Terpisah dari hubungan subkelas, ontology mencakup informasi seperti :

- Property (staff akademik mengajar mahasiswa)
- Pembatasan harga (hanya staff akademik yang dapat mengajar)
- Pernyataan disjoin (staff akademik dan staff administrasi adalah disjoint)
- Spesifikasi hubungan logika diantara objek (staff akademik mengajar mahasiswa, jika banyaknya mahasiswa sama dengan atau lebih dari 10 dan sama dengan atau kurang dari 40)

Ontology meningkatkan keakuratan pencarian Web. Mesin pencari dapat mencari halaman-halaman yang menunjuk ke konsep yang tepat dalam sebuah ontology. Mesin pencari web juga dapat mengeksplorasi informasi generalisasi atau spesialisasi. Jika suatu query gagal untuk menemukan dokumen yang relevan, mesin pencari dapat menyarankan pemakaian sebuah query yang lebih general. Atau untuk mencegah terlalu banyak jawaban yang diberikan, maka mesin pencari dapat menyarankan pencarian yang khusus (spesialisasi).

Bahasa Ontologi membolehkan pemakaian untuk menulis secara eksplisit, mengkonseptualisasi secara formal suatu domain model. Kebutuhan utama itu adalah :

- Sintak yang terdefinisi baik (well-defined syntax)
  - Semantic formal (formal semantics)
  - Dukungan penalaran yang efisien
- Semantik formal memaparkan arti dari pengetahuan secara tepat. Secara tepat disini berarti bahwa semantik tidak menunjuk pada intuisi subjektif, atau terbuka terhadap interpretasi yang berbeda oleh orang (mesin) yang berbeda.

Salah satu penggunaan semantic formal adalah untuk melakukan penalaran tentang pengetahuan. Untuk pengetahuan ontology, dapat dilakukan penalaran tentang :

Keanggotaan Kelas (Class membership). Jika  $x$  adalah sebuah anggota dari kelas C, dan C adalah subkelas dari D, maka dapat disimpulkan bahwa  $x$  adalah anggota dari D. Ekuivalensi dari kelas-kelas. Jika kelas A ekuivalen dengan kelas B, dan kelas B ekuivalen dengan kelas C, maka A ekuivalen dengan C.

Konsistensi. Anggap  $x$  dideklarasikan menjadi anggota dari kelas A dan A adalah subkelas dari B n C, A adalah subkelas dari D, B dan D adalah kelas yang disjoin. Maka ada inkonsistensi karena seharusnya A adalah kosong, tetapi mempunyai anggota  $x$ . Ini adalah sebuah kesalahan dalam ontology.

Klasifikasi. Jika dideklarasikan bahwa pasangan harga-sifat tertentu adalah kondisi yang menyatakan keanggotaan dalam sebuah kelas A, maka jika sebuah individu  $x$  memenuhi beberapa kondisi, maka dapat disimpulkan bahwa  $x$  anggota dari A.

Semantik adalah prasyarat untuk melakukan bantuan penalaran. Bantuan penalaran penting karena membolehkan seseorang untuk :

- menguji konsistensi dari ontology dan pengetahuan
- menguji hubungan yang tidak diharapkan diantara kelas-kelas
- secara otomatis mengklasifikasi anggota dalam kelas-kelas

Pengertian ontology sangat beragam dan berubah sesuai dengan berjalananya waktu, ada beberapa definisi ontology. Nechoes dan rekannya memberikan definisi awal tentang ontology yaitu: “Sebuah ontology merupakan definisi dari pengertian dasar dan relasi vocabulari dari sebuah area sebagaimana aturan dari kombinasi istilah dan relasi untuk mendefinisikan vokabulari”.

Kemudian Gruber memberikan definisi yang sering digunakan oleh beberapa orang, definisi tersebut adalah “Ontology merupakan sebuah spesifikasi eksplisit dari konseptualisme”.

Berdasarkan definisi Gruber tersebut banyak orang yang mengemukakan definisi tentang ontology diantaranya Guarino dan Giaretta yang pada tahun 1995 mengumpulkan hingga tujuh definisi yang berkoresponden dengan syntactic dan semantic interpretasi. Sedangkan pada tahun 1997, Borst melakukan penambahan dari definisi Gruber dengan mengatakan “Sebuah ontology adalah spesifikasi formal dari sebuah konseptual yang diterima (share)”.

Kemudian oleh Studer mencoba mengemukakan definisi tentang ontology yang mengambil acuan dari definisi yang dikemukakan oleh Gruber dan Borst, definisi tersebut adalah : “Konseptualisasi mengacu kepada sebuah model abstrak dari beberapa fenomena di dunia dengan memiliki identifikasi konsep yang relevan dari fenomena tersebut. Yang dimaksud dengan eksplisit adalah tipe dari konsep yang digunakan, dan batasan dari eksplisit yang digunakan. Shared adalah merefleksikan bahwa sebuah ontology mencoba menangkap pengetahuan secara konsensus yang tidak merupakan hal yang hanya terkait pada individu tetapi diterima oleh sebuah group/domain”.

Barnaras pada proyek KACTUS memberikan definisi ontology yang berdasarkan pada pengembangan ontology. Definisi yang diberikan adalah : “Sebuah ontology memberikan pengertian untuk penjelasan secara eksplisit dari konsep terhadap representasi pengetahuan pada sebuah knowledge base”. Proyek SENSUS juga memberikan definisi : “Sebuah ontology adalah sebuah struktur hierarki dari istilah untuk menjelaskan sebuah domain yang dapat digunakan sebagai landasan untuk sebuah knowledge base”.

Ada buku yang memberikan definisi tentang ontology, salah satunya adalah “The Semantic Web”, definisi dari ontology adalah :

1. Salah satu cabang metafisika yang terfokus pada alam dan hubungan antara makhluk hidup;
2. Teori tentang sifat alami makhluk hidup. Ontology merupakan suatu teori tentang makna dari suatu objek, property dari suatu objek, serta relasi objek tersebut yang mungkin terjadi pada suatu domain pengetahuan.

Pada tinjauan filsafat, ontology adalah studi tentang sesuatu yang ada. Selain itu ontology adalah sebuah konsep yang secara sistematis menjelaskan tentang segala sesuatu yang ada atau nyata. Dalam bidang Artificial Intelligence (AI) ontology memiliki dua pengertian yang berkaitan. Pertama ontology merupakan kosakata representasi yang sering dikhususkan untuk domain atau subjek pembahasan tertentu. Kedua, sebagai suatu body of knowledge untuk menjelaskan suatu bahasan tertentu.

Literature yang berisi tentang Artificial Intelligence banyak menjelaskan tentang definisi ontology, banyak yang bertentangan satu dengan yang lainnya. Tetapi dapat diambil kesimpulan, ontology adalah sebuah uraian formal yang menjelaskan tentang sebuah konsep dalam sebuah domain tertentu (Classes, terkadang disebut concepts), properties dari masing-masing konsep menjelaskan bermacam-macam corak dan atribut dari sebuah concept (Slots, terkadang disebut roles atau properties), dan batasan-batasan (facets, terkadang disebut role restrictions). Sebuah ontology bersama dengan beberapa set instances dari class membentuk sebuah knowledge base.

Secara umum, ontology digunakan pada Artificial Intelligence (AI) dan persentasi pengetahuan. Segala bidang ilmu yang ada di dunia, dapat menggunakan metode ontology untuk dapat berhubungan dan saling berkomunikasi dalam hal pertukaran informasi antara sistem-sistem yang berbeda.

Ontology menggunakan banyak variasi struktur, tergantung dari penggunaan bahasa ontology termasuk sintaksis yang digunakan. Perlu diingat adalah ontology tidak melakukan apapun, fungsi perhitungan dan lainnya yang memproses ontology tidak hanya tergantung dari data yang terdapat dalam ontology tersebut, tetapi juga tergantung kepada aplikasi yang digunakan.

Ontology memiliki beberapa komponen yang dapat menjelaskan ontology tersebut, diantaranya [14]:

- Konsep (Concept)

Digunakan dalam pemahaman yang luas. Sebuah konsep dapat sesuatu yang dikatakan, sehingga dapat pula merupakan penjelasan dari tugas, fungsi, aksi, strategi, dan sebagainya. Concept juga dikenal sebagai classes, object dan categories.

- Relasi (relation)

Merupakan representasi sebuah tipe dari interaksi antara konsep dari sebuah domain. Secara formal dapat didefinisikan sebagai subset dari sebuah produk dari n set,  $R:C_1 \times C_2 \times \dots \times C_n$ . Sebagai contoh dari relasi binary termasuk subclass-of dan connected-to.

- Fungsi (functions)

Adalah sebuah relasi khusus dimana elemen ke-n dari relasi adalah unik untuk elemen ke n1.  $F:C_1 \times C_2 \times \dots \times C_{n-1} \rightarrow C_n$ , contohnya adalah Mother-of.

- Aksiom(axioms)

Digunakan untuk memodelkan sebuah sentence yang selalu benar.

- Instances

Digunakan untuk merepresentasikan elemen.

Untuk dapat digunakan, sebuah ontology harus diekspresikan dalam notasi yang nyata. Sebuah bahasa ontology adalah sebuah bahasa formal dari sebuah pengembangan ontology. Beberapa komponen yang menjadi struktur ontology, antara lain:

- XML (Extensible Markup Language)

Menyediakan sintaksis untuk output dokumen terstruktur, tetapi belum dipaksakan untuk dokumen XML menggunakan semantic constrains.

- XML Schema

Bahasa untuk pembatasan struktur dari dokumen XML.

- RDF (Resource Description Framework)

Model data untuk objek („resources“) dan relasi diantaranya, menyediakan semantic yang sederhana untuk model data tersebut, dan data model ini dapat disajikan dalam sintaksis XML.

- RDF Schema

Adalah kosakata untuk menjelaskan properties dan classes dari sumber RDF, dengan sebuah semantics untuk hirarki penyamarataan dari properties dan classes.

- OWL (Ontology Web Language)

Menambahkan beberapa kosakata untuk menjelaskan properties dan classes, antara lain : relasi antara classes (misalkan disjointness), kardinalitas (misalkan „tepat satu“), equality, berbagai tipe dari properties, karakteristik dari properties (misalkan symmetry), menyebutkan satu persatu classes.

Berbagai bahasa yang menyusun ontology, seperti yang telah dijelaskan di atas memiliki kedudukan tertentu dalam struktur ontology. Setiap layer akan memiliki fungsi tambahan dan kompleksitas

tambahan dari layer sebelumnya. Pengguna atau user yang memiliki fungsi pemrosesan layer paling rendah dapat memahami walaupun tidak seluruh ontology yang terletak di layer atasnya.

Extensible Markup Language (XML) adalah sebuah format teks yang sederhana yang berdasarkan SGML(ISO 8879), yang didesain untuk mempertemukan berbagai macam sumber informasi dalam dunia web.

### A. Fungsi dan Tujuan XML

XML sudah banyak dikenal oleh banyak orang, dan adalah dasar untuk pengembangan Software yang meningkat dengan pesat. XML adalah dokumen yang menyimpan data dalam struktur-struktur yang dapat berubah-ubah, hal ini berbeda dengan html yang didesain untuk dokumen hypertext dengan struktur yang baku. Struktur XML yang baik menciptakan struktur yang berbentuk hirarki terstruktur yang memiliki pasangan tags awal dan akhir, yang dapat terdiri dari beberapa atribut yang berpasangan. Tidak ada aturan kosakata tags yang baku atau pasangan tags yang diperbolehkan, jadi hal ini diatur di setiap aplikasi.

### B. Sintaksis dan Elemen XML

Sintak dokumen XML yang sederhana terdiri dari deklarasi XML dan elemen puncak. Deklarasi XML merupakan tempat untuk menyatakan Versi dari XML dan encoding untuk dokumen tersebut. Untuk dokumen XML standar versi yang tersedia adalah versi “1.0” dan menggunakan ISO-8859-1 (Latin-1/West European) sebagai encodingnya.

Bagian yang selanjutnya adalah elemen-elemen yang menyusun dokumen XML tersebut. Setiap elemen tersebut memiliki tags penutup. Tags dalam XML memperhatikan penggunaan huruf atau dalam arti Case Sensitive. Dalam dokumen XML setiap elemen harus tersusun dengan benar. Dalam arti setiap elemen harus benar-benar terkuring.

Sebuah dokumen XML harus memiliki elemen utama. Sedangkan setiap nilai atribut dari elemen tersebut harus menggunakan tanda kutip dua (“ ”).

Berikut ini adalah contoh sintak XML yang sederhana :

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

## **Pengembangan Projek**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>14</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Buatlah projek masing-masing dengan ketentuan sebagai berikut

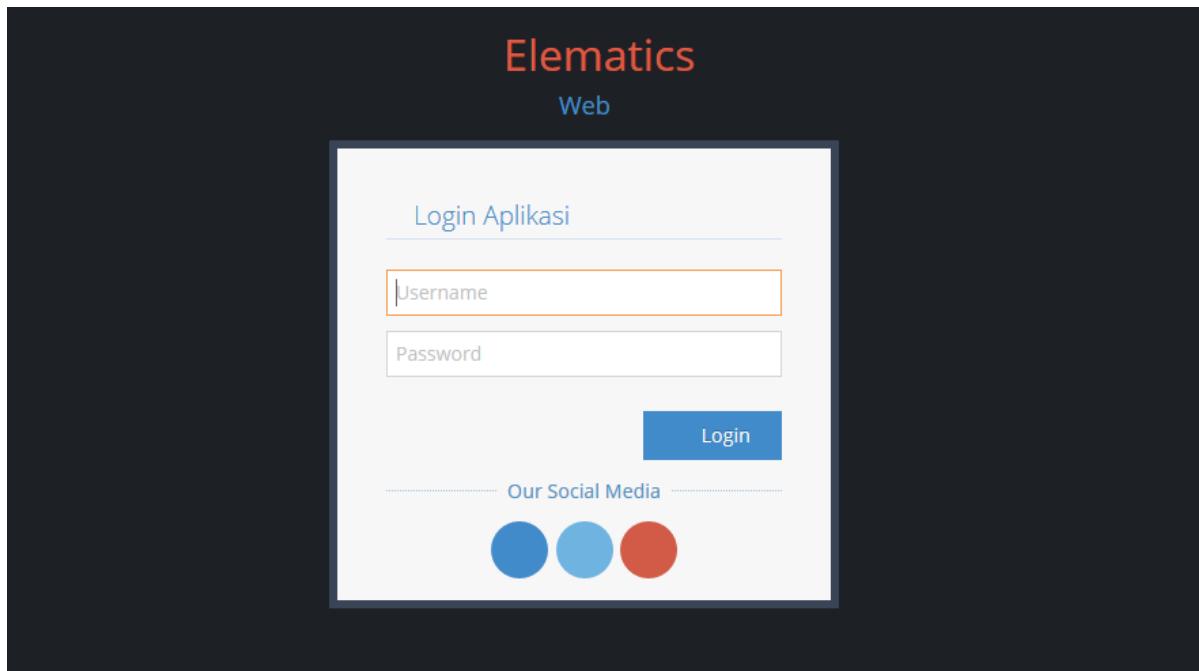
Tema	Nama Projek
1	E-commerce
2	Ritail
3	Company Profile
4	Pendidikan
5	Travel
6	Koperasi Simpan Pinjam
7	Penanganan Complain
8	Indeks Kepuasan Pelanggan

Buat menjadi delapan kelompok dengan ketentuan masing-masing kelompok membuat dari sisi back-end dan juga front-end.

Gunakan template yang diberikan selama pembelajaran yaitu menggunakan Ace-Master untuk back-end, dan gunakan tempalte lain untuk front-end sesuai dengan tema masing-masing.

Sehingga tampilan yang dihasilkan akan nampak pada beberapa gambar-gambar berikut ini

## A. Log-In



## B. Tampilan Awal

A screenshot of a web application titled "Wawan Gunawan". The header features a logo and the name "Wawan Gunawan". On the right, there is a welcome message "Selamat datang, Administrator" with a profile picture. The left sidebar has a navigation menu with items like "Dashboard", "Master Web", "Kelola Website", "Pengaturan", "Gallery", "Layanan Pengunjung", and "Kursus On-Line". The main content area shows a "Dashboard" section with a message "Selamat datang Administrator, Berusaha dan berdoa". Below this is a "Kelola Halaman Web" section with three cards: "Komentar" (0), "Testimoni" (0), and "pageviews" (6868). At the bottom, there is a footer with the text "Wawan Gunawan, S.Kom., M.T. Application © 2018" and social media links.

## C. Data Pengguna

### a. Query

The screenshot shows a web-based application for managing users. At the top, there's a navigation bar with 'Pengaturan > Setting Pengguna'. Below it, a blue header bar says 'Hasil untuk "Setting Pengguna"'. The main area is a table with columns: No, Foto, Alamat, Status, and actions. There are two rows of data:

No	Foto	Alamat	Status	
1		08179827384	Aktif	
2		089673566326	Aktif	

### b. Form

The screenshot shows a form for creating or updating a user. The top part has fields for Name, Gender (Female), Access Level (Petugas), Username, Password, and Confirmation Password. Below that is a large rich text editor for Address. Further down are fields for Telpon, Gambar (with a browse button), and Status (set to Tidak Aktif). At the bottom are 'Simpan' and 'Tutup' buttons.

Fields visible in the form:

- Nama Lengkap
- Jenis Kelamin: Perempuan
- Level Akses: Petugas
- Username
- Password
- Konfirmasi Password
- Alamat (Rich Text Editor)
- Telp
- Gambar (Browse button)
- Status: Tidak Aktif
- Buttons: Simpan, Tutup

## D. Data Kelompok

### a. Query

The screenshot shows a web application titled "Master Web > Kategori Berita". A search bar at the top right contains the text "Kategori Berita". Below it is a blue header bar with the text "Hasil untuk 'Kategori Berita'". A table displays 10 records, each with a "No" column (1-10), a "Kategori Berita" column, and edit/delete icons. The data is as follows:

No	Kategori Berita	
1	Dunia Dosen	
2	Me and Fams	
3	Dunia Ngoding	
4	Dunia Mahasiswa	
5	Pertemuan Akademik	
6	Seminar	
7	Artikel Ilmiah	
8	Buku Pemrograman	
9	Tri Dharma Perguruan Tinggi	
10	Software	

### b. Form

The screenshot shows a modal dialog box titled "Kategori Berita" with a close button "X". Inside the dialog, there is a text input field labeled "Kategori Berita". At the bottom right of the dialog are two buttons: "Simpan" (Save) and "Tutup" (Close). The background shows the same list of news categories as the previous screenshot. The data is as follows:

No	Kategori Berita
1	Dunia Dosen
2	Me and Fams
3	Dunia Ngoding
4	Dunia Mahasiswa

## E. Terdapat Combo Box

### a. Query

Kelola Website > Kelola Website

### Kelola Website

Hasil untuk "Kelola Website"

No	Menu - Kategori	Isi Website	Status	Pengunjung	
1	 Berita Seminar Tidak halaman depan	<p>Sosialisasi dan Workshop Panduan Penulisan Proposal PPM Hibah Dikti Semangat perguruan tinggi dalam membangun negeri melalui program pengabdian kapada masyarakat yang diadakan pada Kamis tanggal 02 Mei 2019 di ruang Prof. Harun Zein dengan pembicara bapak Dr. Wisnu Nurcahyo dari Universitas Gajah Mada yang juga merupakan Reviewer Nasional Kemenristekdikti.</p> <p>Beberapa yang harus diperhatikan dalam penulisan proposal PPM Hibah Dikti ini harus berbeda dengan proposal penelitian. Selain itu juga penulisan judul harus diperhatikan dengan memperhatikan <b>[Tingkat PPM] [eye catching]</b> <b>[Lokasi Mitra]</b> dan juga harus dari multidisiplin. Untuk pengabdian multi tahun ikutsertakan mahasiswa sebanyak 4 orang dan untuk mono tahun sebanyak 2 orang mahasiswa. Contoh judul <b>PKM KELOMPOK GURU PAUD DENGAN POLA ASUH HOLISTIK DI PAUD CEMPAKA KELURAHAN PANGGUNGKHARJO</b>.</p> <p><b>Perhatikan:</b></p> <ol style="list-style-type: none"><li>1. Pengabdian masyarakat harus melakukan analisis secara terukur, dengan memperhatikan <i>Before After</i> Pengmas.</li><li>2. Perbaiki CV di Google Scholar, dan masukkan secara manual</li></ol> 	Aktif	34	 

### b. Form

Kelola Website > Kelola Website

### Kelola Website

Hasil untuk "Kelola Website"

Menu:

Judul:

Isi: 

Gambar:  

Kategori:

Tampilkan Album:

Status:  

Halaman Depan:  

## F. Terdapat Laporan

Master Web > Laporan

### Laporan

Hasil untuk "Laporan"

Periode Laporan  s.d

Wawan Gunawan, S.Kom., M.T. Application © 2018 [Twitter](#) [Facebook](#) [RSS](#)

## G. Penanganan File

Master Web > Handling File

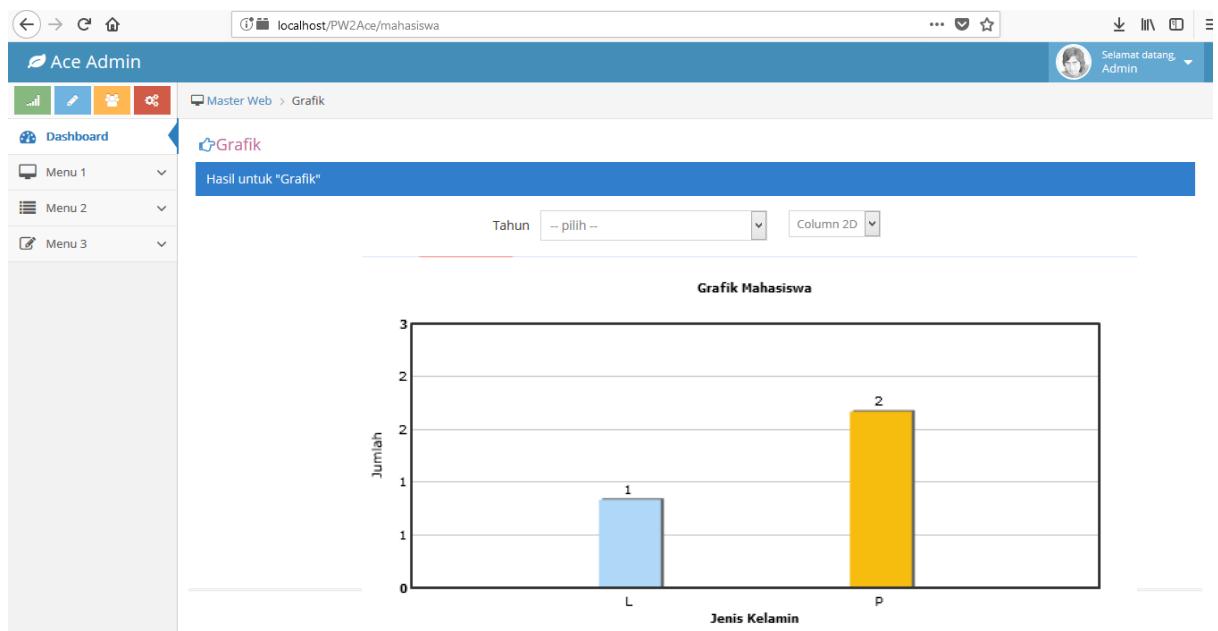
### Handling File

Hasil untuk "Handling File"

File  No files selected.



## H. Terdapat Grafik



# LATIHAN

Presentasikan hasil kerja kelompok

## Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.



## **MODUL PERKULIAHAN**

# **Pemrograman Web 2**

## **Pengembangan Projek (Lanjutan)**

Fakultas	Program Studi	Tatap Muka	Kode MK	Disusun Oleh
Fakultas Ilmu Komputer	Teknik Informatika	<b>15</b>	W151700013	Wawan Gunawan, S.Kom., M.T.

### **Abstract**

Membuat halaman HTML sederhana

### **Kompetensi**

Mahasiswa mampu membuat halaman sederhana dengan berbagai macam tag

# Pembahasan

Buatlah projek masing-masing dengan ketentuan sebagai berikut

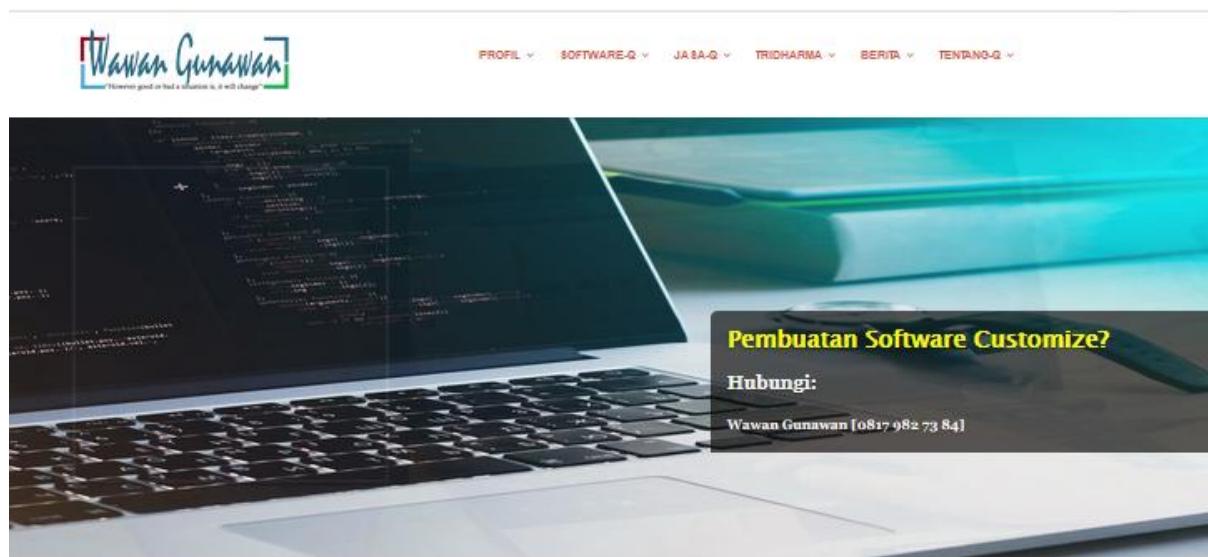
Tema	Nama Projek
1	E-commerce
2	Ritail
3	Company Profile
4	Pendidikan
5	Travel
6	Koperasi Simpan Pinjam
7	Penanganan Complain
8	Indeks Kepuasan Pelanggan

Buat menjadi delapan kelompok dengan ketentuan masing-masing kelompok membuat dari sisi back-end dan juga front-end.

Gunakan template yang diberikan selama pembelajaran yaitu menggunakan Ace-Master untuk back-end, dan gunakan tempalte lain untuk front-end sesuai dengan tema masing-masing.

Sehingga tampilan yang dihasilkan akan nampak pada beberapa gambar-gambar berikut ini untuk Front-End

## A. Tampilan Awal



### Produk Kami



#### Inventory Control

Apa itu Inventory ???? Menurut Mulyadi (2010) dalam bukunya "Sistem Akuntansi" menjelaskan Persediaan/Inventory meliputi barang yang dibeli dan dimiliki untuk dijual kembali, misalnya, barang dagangan yang dibeli oleh pengecer untuk dijual kembali, atau pengadaan tanah dan properti selengkapnya....



#### Koperasi Simpan Pinjam

SOFTWARE KOPERASI SIMPAN PINJAM software koperasi adalah aplikasi untuk membantu dalam pengelolaan data dan keuangan koperasi, dilengkapi dengan aplikasi pertokoan. Software koperasi dapat mempercepat dan memperlancar transaksi di Koperasi. Manfaat utama Software Membantu petugas adminis.....



#### Key Performance Indicator

Penilaian terhadap suatu pelajaran dalam sebuah perusahaan merupakan suatu tahap evaluasi kerja yang dapat meningkatkan kualitas pekerjaan bagi kelangsungan aktivitas perusahaan di dalamnya. Jika proses pemberdayaan melalui training telah dilaksanakan, pentinglah memantau perkembangannya dan



#### Point of Sales

Program kasir yang dilengkapi dengan antar muka yang khusus dibuat untuk kemudahan kasir berbagai jenis toko, outlet, butik, toko minuman, toko sembako, mini-market, maupun retailer dan toko grosir. Dengan fitur yang lengkap mendukung multi kasir, multi user, multi gudang, multi katalog promosi &..... [Baca](#)

## B. Contoh Menu



## C. Tampilan Gambar

### a. Galery

[Software Inventory Control](#)

[Software Koperasi Simpan Pinjam](#)

[Software Key Performance Indicator](#)

[Software Akuntansi](#)

[Software Point of Sales](#)

[Security System](#)

[Peserta Les Pemrograman](#)

[Implementasi Sistem](#)

[Data Supplier](#)

[Form Supplier](#)

[Form Pelanggan](#)

[Form Pembelian](#)

Form ini digunakan untuk melakukan pendaftaran data pelanggan pada perusahaan dengan langkah yang baru ....baca selengkapnya

**Data Pelanggan**

## b. Detil Gambar

2. Jika sudah sesuai, lakukan proses penyimpanan dengan memilih tombol Simpan

## Data Pelanggan

Form ini digunakan untuk melakukan pendaftaran data pelanggan pada perusahaan dengan langkah yang harus dilakukan ketika pengguna ingin melakukan penambahan data dengan kondisi Kode Pelanggan yang terbentuk merupakan hasil auto generate:

1. Isikan data pada seluruh isian yang telah disediakan pada form isian untuk penggunaan dari manfaat data pelanggan. Khusus isian yang diberi warna pink wajib diisi karena bersifat mandatory

## D. Terdapat Contact

### a. Alamat dan Nara hubung

**Wawan Gunawan, S.Kom., M.T.**



**Kp. Kelapa Indah RT 003 RW 005 No. 05, Kelurahan Kelapa Indah - Kota Tangerang 15118 - Provinsi Banten - Indonesia**

Phone: 0817-982-73-84

e-mail: wawan@wawangunawan.web.id

b. Form

---

## Kirim Pesan

Nama Anda

Email:

Subjek:

Pesan Anda:

**Kirim Pesan**

## E. Terdapat Laporan

Master Web > Laporan

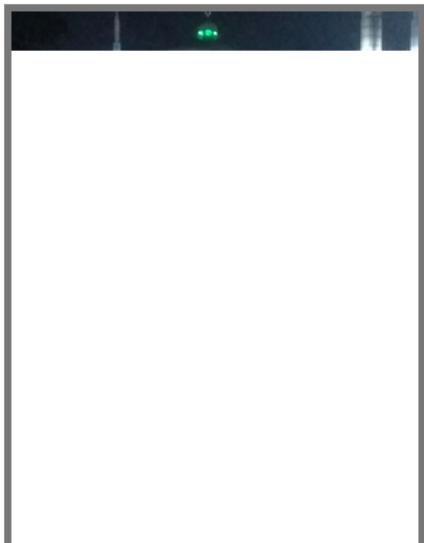
Laporan

Hasil untuk "Laporan"

Periode Laporan    mm / dd / yyyy    s.d    mm / dd / yyyy

Proses     Tutup

## F. Visi dan Misi



**Wawan Gunawan** merupakan anak ke empat dari 4 bersaudara yang terlahir dari pasangan ibu dan bapak keturunan Betawi. Ia dilahirkan di Kota Tangerang pada tahun 1981.

Kali pertama sekolah pada tahun 1987 - 1993 di SDN Cikokol 2 Kota Tangerang, selanjutnya melanjutkan studi di SMP Negeri 3 Kota Tangerang pada tahun 1993 - 1996. Setelah menamatkan pendidikan di tingkat pertama, selanjutnya Ia melanjutkan studi di SMA PGRI 22 Serpong pada tahun 1996 - 1999. Organisasi yang pernah Ia ikuti ketika tingkat SMA adalah Pramuka dan OSIS.

Setelah lulus SMA, Ia melanjutkan program profesi 1 tahun pada jurusan Komputer Akuntansi dan selesai pada tahun

[redacted] tahun 2001 mulai memasuki dunia kerja di salah satu unit usaha Kompas

data from [www.wawangunawan.web.id...](http://www.wawangunawan.web.id...)

## G. Kumpulan Berita

The screenshot shows the header of the website. On the left is the logo "Wawan Gunawan" with the tagline "However good or bad a situation is, it will change". On the right are menu links: PROFIL, SOFTWARE-Q, JASA-Q, TRIDHARMA, and BERITA. Below the header, there is a dark banner with white text: "Berita" and "Kumpulan Berita Wawan Gunawan, S.Kom., M.T.". To the right of the banner is a stylized "W" logo.



02 May 2019 10:05:52 / 0 Comments / Seminar / 34 /

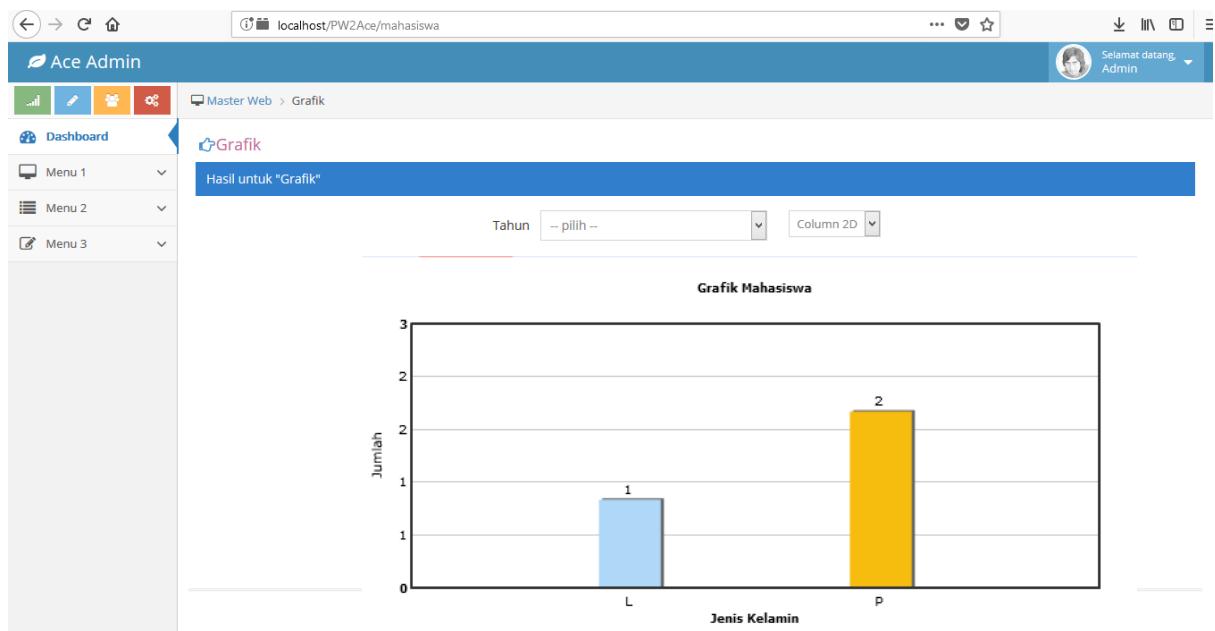
Administrator

### Sosialisasi dan Workshop Panduan Penulisan Proposal PPM Hibah Dikti

Semangat perguruan tinggi dalam membangun negeri melalui program pengabdian kapada masyarakat yang diadakan pada Kamis tanggal 02 Mei 2019 di ruang Prof. Harun Zein dengan pembicara bapak Dr. Wisnu Nurcahyo dari Universitas Gajah Mada yang juga

[redacted]  
data from [www.wawangunawan.web.id...](http://www.wawangunawan.web.id...)

## H. Terdapat Grafik



# LATIHAN

Presentasikan hasil kerja kelompok

# Daftar Pustaka

Bahra. 2005. *Analisis dan Desain Sistem Informasi*. Tangerang: Graha Ilmu.

Hermawan, Benny . 2005. Menguasai Java 2 & Object Oriented Programming. Yogyakarta: Andi Yogyakarta.

Hakim, Rachmad dan Sutarto . 2009. Konsep Pemrograman Java dan Penerapannya untuk Membuat Software Aplikasi. Jakarta: Elex Media Komputindo.

Jogiyanto. 2005. *Analisis & Disain Sistem Informasi: Pendekatan Terstruktur Teori dan Praktek Aplikasi Bisnis*. Yogyakarta: Andi Yogyakarta.