

PEMROGRAMAN WEB LANJUT

(ARRAY, FUNGSI DAN CRUD DENGAN CODEIGNITER)



AGUS SETIYO BUDI N



Diterbitkan Atas Kerjasama
DeePublish dengan Politeknik Banjarmasin



PEMROGRAMAN WEB LANJUT

(ARRAY, FUNGSI DAN CRUD DENGAN CODEIGNITER)

UU No 28 tahun 2014 tentang Hak Cipta

Fungsi dan sifat hak cipta Pasal 4

Hak Cipta sebagaimana dimaksud dalam Pasal 3 huruf a merupakan hak eksklusif yang terdiri atas hak moral dan hak ekonomi.

Pembatasan Pelindungan Pasal 26

Ketentuan sebagaimana dimaksud dalam Pasal 23, Pasal 24, dan Pasal 25 tidak berlaku terhadap:

- i. penggunaan kutipan singkat Ciptaan dan/atau produk Hak Terkait untuk pelaporan peristiwa aktual yang ditujukan hanya untuk keperluan penyediaan informasi aktual;
- ii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk kepentingan penelitian ilmu pengetahuan;
- iii. Penggandaan Ciptaan dan/atau produk Hak Terkait hanya untuk keperluan pengajaran, kecuali pertunjukan dan Fonogram yang telah dilakukan Pengumuman sebagai bahan ajar; dan
- iv. penggunaan untuk kepentingan pendidikan dan pengembangan ilmu pengetahuan yang memungkinkan suatu Ciptaan dan/atau produk Hak Terkait dapat digunakan tanpa izin Pelaku Pertunjukan, Produser Fonogram, atau Lembaga Penyiaran.

Sanksi Pelanggaran Pasal 113

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).

PEMROGRAMAN WEB LANJUT

(ARRAY, FUNGSI DAN CRUD DENGAN CODEIGNITER)

Agus Setiyo Budi Nugroho



**PEMROGRAMAN WEB LANJUT
(ARRAY, FUNGSI DAN CRUD DENGAN CODEIGNITER)**

Penulis :
Agus Setiyo Budi Nugroho

e-ISBN :
978-623-92412-4-7 (PDF)

Editor dan Penyunting :
Reza Fauzan

Desain Sampul dan Tata Letak :
Eko Sabar Prihatin; Rahma Indera

Penerbit :
POLIBAN PRESS
Anggota APPTI (Asosiasi Penerbit Perguruan Tinggi Indonesia)
no.004.098.1.06.2019

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang
Dilarang memperbanyak karya tulis ini dalam bentuk
dan dengan cara apapun tanpa ijin tertulis dari penerbit

Redaksi :
Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry,
Pangeran, Komp. Kampus ULM, Banjarmasin Utara
Telp : (0511)3305052
Email : press@poliban.ac.id

Dicetak oleh :
PERCETAKAN DEEPUBLISH
Jl.Rajawali, G. Elang 6, No 3, Drono, Sardonoarjo, Ngaglik, Sleman
Jl.Kaliurang Km.9,3 – Yogyakarta 55581
Telp/Faks: (0274) 4533427
Website: www.deepublish.co.id
www.penerbitdeepublish.com
E-mail: cs@deepublish.co.id

Katalog Dalam Terbitan (KDT)
**Agus Setiyo Budi Nugroho—Cet. I. — Pemrograman Web Lanjut (Array, Fungsi dan
Crud dengan Codeigniter) : Poliban Press, 2019.**

xii ; 70 hlm.; 15.5x23 cm

Ucapan Terima Kasih

Ucapan terima kasih kami sampaikan kepada Poliban Press karena telah mempercayakan proses percetakan buku Pemrograman Web Lanjut (Array, Fungsi dan Crud dengan Codeigniter) kepada Penerbit Deepublish. Semoga buku ini dapat memberikan manfaat kepada seluruh pembaca dan kerja sama ini dapat terus terjalin.



Kata Pengantar

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan karunianya sehingga Buku Pemrograman Web Lanjut tahun 2019 telah dapat diselesaikan. Buku ini merupakan pengantar pada mata kuliah pemrograman web bagi mahasiswa Program Studi Teknik Informatika.

Terima kasih disampaikan kepada Joni Riadi S.ST., M.T. selaku Direktur Politeknik Negeri Banjarmasin dan Nurmahaludin, S.T., M.T. selaku Ketua Pusat Penelitian dan Pengabdian Masyarakat beserta sekretaris dan staf. Terima kasih juga disampaikan kepada Faris Ade Irawan, Reza Fauzan, Eko Sabar Prihatin dan Rahma Indera yang telah berkontribusi dalam editing serta seluruh tim Poliban Press dan semua pihak yang telah ikut membantu dalam penyelesaian buku ini.

Kami menyadari masih terdapat kekurangan dalam buku ini untuk itu kritik dan saran terhadap penyempurnaan buku ini sangat diharapkan. Semoga buku ini dapat memberi manfaat bagi mahasiswa Politeknik khususnya dan bagi semua pihak yang membutuhkan.

Banjarmasin, Agustus 2019

Ketua Poliban Press

Prakata

Alhamdulillah, akhirnya buku pemrograman web lanjut yang membahas *framework codeigniter* pada mata kuliah pemrograman web pada kampus POLIBAN (Politeknik Negeri Banjarmasin) Prodi TI dapat diselesaikan.

Sebelum membahas CI, buku ini memulai pembahasan tentang array, fungsi, prosedur dan file. Hal ini dikarenakan, teramat banyak array dan fungsi yang dipakai dalam CI, sehingga pemahaman tentang hal tersebut mutlak diperlukan.

Metode buku ini adalah, membaca, memahami dan mencoba. Jika tidak mencoba, kecil kemungkinan akan paham dan mengerti, mengingat banyak faktor X yang muncul saat kita melakukan coding.

Akhir kata, semoga buku ini bermanfaat kepada para pembaca semuanya dalam memahami cara kerja framework Codeigniter untuk membangun sistem informasi sederhana CRUD untuk dua tabel yang berelasi.

Banjarmasin, Juni 2019

Penulis

Daftar Isi

Ucapan Terima Kasih	v
Kata Pengantar	vi
Prakata.....	vii
Daftar Isi	viii
BAB 1 ARRAY.....	1
1.1. Konsep Array.....	1
1.2. Array dengan Index Urut dan Otomatis.....	2
1.3. Mencetak Array dengan FOR.....	2
1.4. Mencetak Array dengan FOREACH	3
1.5. Array dengan Indeks Tidak Beraturan (<i>Assosiative</i>)	4
1.6. Mencetak Array dengan FOR.....	5
1.7. Mencetak Array dengan FOREACH	6
1.8. Latihan	7
BAB 2 OPERASI ARRAY.....	9
2.1. Mencari Nilai Min Max di Array	9
2.2. Menyisipkan Atau Menghapus Elemen Array.	10
2.3. Mengurutkan Isi Array.	11
2.4. Memecah String Menjadi Array.	13
2.5. Contoh Array Kasus dengan HTML.....	13
2.6. Latihan	15
BAB 3 SUBPROGRAM.....	17
3.1. Fungsi Sederhana Tanpa Variabel.....	17
3.2. Fungsi Dengan Mengirim Variabel.....	18
3.3. Fungsi dengan Megembalikan Variabel.....	19
3.4. Latihan	19

BAB 4	AKSES FILE	21
4.1.	Menyimpan String ke File	21
4.2.	Membaca File Text ke String	22
4.3.	Menyimpan Array ke File	22
4.4.	Membaca file ke array	23
4.5.	Latihan	23
BAB 5	FRAMEWORK CODEIGNITER (CI)	25
5.1.	Konsep MCV (Model, Controller, View)	25
5.2.	Instalasi CI	26
5.3.	Struktur Folder CI	28
5.4.	Memulai CI	29
5.5.	Mengirim Data dari Controller ke View	30
5.6.	Latihan	31
BAB 6	TEMPLATE	33
6.1.	Membuat Template	33
6.1.1.	Config	33
6.1.2.	Controller	34
6.1.3.	View	34
6.1.4.	Jalankan	36
6.2.	Membuat Prosedur dan Efisiensi	36
6.3.	Latihan	37
BAB 7	MENAMPILKAN ISI TABEL (READ)	38
7.1.	Membuat Database Kampus Dan Tabel Mahasiswa	38
7.2.	Coding CI Untuk Menampilkan Isi Tabel	40
7.2.1.	Config	40
7.2.2.	Controller	40
7.2.3.	Model	41
7.2.4.	View	41
7.2.5.	Jalankan di Browser	42
BAB 8	CRUD (LANJUTAN)	43
8.1.	Default Controller	43
8.1.1.	Config	43
8.2.	<i>Create</i> (Tambah Data)	44

8.2.1.	Config	44
8.2.2.	Controller	44
8.2.3.	Model.....	45
8.2.4.	Views	45
8.2.5.	Jalankan.....	46
8.3.	<i>Update</i> (<i>Edit/Ubah data</i>)	48
8.3.1.	Config	48
8.3.2.	Controller	49
8.3.3.	Model.....	49
8.3.4.	Views	50
8.3.5.	Jalankan.....	50
8.4.	<i>Delete</i> (<i>Hapus data</i>).....	52
8.4.1.	Config	52
8.4.2.	Controller	52
8.4.3.	Model.....	52
8.4.4.	View	52
8.4.5.	Jalankan.....	53
8.5.	Latihan	53
BAB 9	RELASI TABEL.....	55
9.1.	Pembahasan Pembuatan CRUD untuk Tabel Beasiswa	55
9.1.1.	Controller	55
9.1.2.	Berikut Model Beasiswa.....	56
9.1.3.	Modifikasi <i>Views</i>	57
9.2.	Merelasikan Tabel Mahasiswa dengan Beasiswa.....	59
9.3.	<i>Coding</i> CRUD Berelasi.	60
9.3.1.	Controller	61
9.3.2.	Model.....	62
9.3.3.	View	62
9.3.4.	Jalankan.....	63
BAB 10	CETAK DAN SEARCH	65
10.1.	Cetak.....	65
10.1.1.	Controller	65
10.1.2.	View	66

10.2. <i>Search</i>	67
10.2.1. Modifikasi <i>Controller</i>	68
10.2.2. Modifikasi <i>Model</i>	68
10.2.3. Modifikasi <i>Views</i>	68
10.2.4. Hasil	69
Daftar Pustaka	70

BAB 1

ARRAY

Capaian Pembelajaran:

1. Mampu memahami prinsip variabel array
2. Mampu memahami array dengan indeks urut dan tidak
3. Mampu membuat array di PHP
4. Membuat Aplikasi array di PHP

Array adalah kumpulan beberapa data yang disimpan dalam satu variabel. Jadi, berbeda dengan variabel tunggal yang biasa dipakai sebelumnya, array memiliki banyak kelebihan dalam pengolahan data.

1.1. Konsep Array

Perhatikan contoh berikut untuk membedakan array dan variabel tunggal.

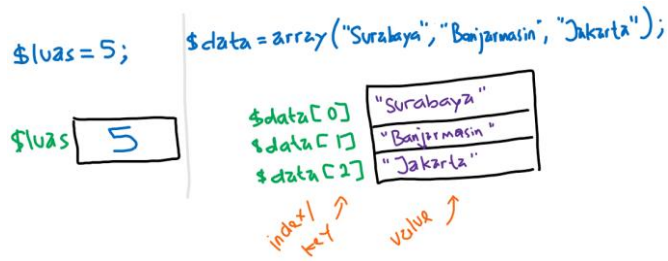
Variabel tunggal: Hitunglah Luas sebuah persegi.

```
$Luas = $pj * $lb;
```

Variabel array : Carilah nilai maksimal dan beberapa data yang dimasukkan.

```
$data=array(3,5,2,8,6,3);
```

Berikut ini contoh lain dalam bentuk ilustrasi untuk membedakan variabel tunggal dan array.



Gambar. 1. (kiri) Ilustrasi variabel tunggal (kanan) ilustrasi variabel array

1.2. Array dengan Index Urut dan Otomatis

Sebuah variabel array yang dibuat dengan mengisi value saja tanpa indeks, akan menghasilkan sebuah array dengan indeks urut yang dimulai dari 0 hingga panjang array. Juga sering disebut *indexed array*.

Berbeda dengan `echo` yang dapat menampilkan sebuah variabel, untuk mencetak sebuah variabel array kita memerlukan perintah `print_r()`. Berikut ini contoh array dengan indeks otomatis dengan 3 value.

```

1 <?php
2 $data=array("Surabaya","Banjarmasin","Jakarta");
3 print_r($data);
4 ?>

```

Tab Size: 4 PHP

localhost/buku/01_array.php

Array ([0] => Surabaya [1] => Banjarmasin [2] => Jakarta)

Gambar 2. Membuat Array Index Otomatis

1.3. Mencetak Array dengan FOR

Ada beberapa cara untuk mencetak array. Prinsipnya harus menggunakan perulangan. Perulangan dengan `for`, hanya akan dapat berjalan jika kita dapat mengetahui nilai batas akhir array (dimensi array). Untuk itu kita akan menggunakan perintah `count()` pada array tersebut untuk mengetahui berapa banyak elemen array yang ada.

```
1 <?php
2 $data=array("Surabaya","Banjarmasin","Jakarta");
3 echo "Panjang array = ".count($data);
4 echo "<br>Isi array: <br>";
5 for($i=0;$i<count($data);$i++)
6 {
7     echo $data[$i];
8     echo "<br>";
9 }
10 ?>
```



Gambar 3. Menampilkan Isi Array dengan Perintah For

1.4. Mencetak Array dengan FOREACH

Untuk menangani perulangan dengan array, beberapa bahasa pemrograman sudah membuat perintah standar yaitu `foreach()`. Sebuah perintah perulangan yang didalamnya sudah tidak perlu lagi mendefinisikan variabel kunci, baik nilai awal hingga nilai akhir kunci. Berikut ini contoh `foreach()` untuk menampilkan nilai saja.

```
1 <?php
2 $data=array("Surabaya","Banjarmasin","Jakarta");
3 foreach($data as $d)
4 {
5     echo $d;
6     echo "<br>";
7 }
8 ?>
```

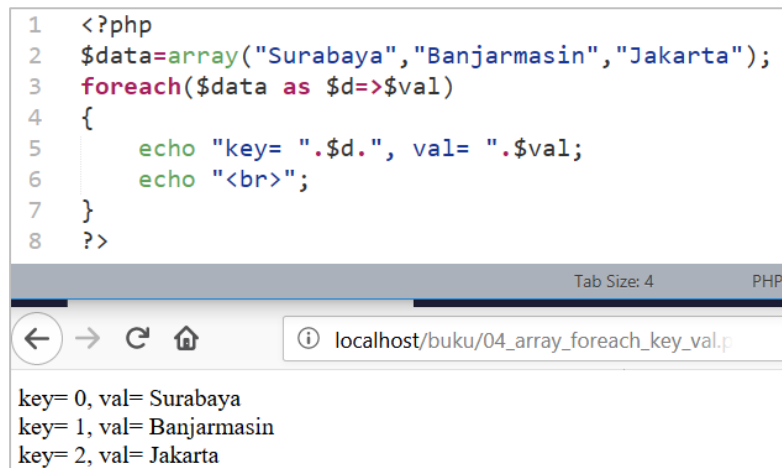


Gambar 4. Mencetak Nilai Array Menggunakan Foreach()

Pada perintah `foreach($data as $d)` diatas, `$data` adalah variabel array dan `$d` adalah nama bebas variabel terserah kita yang digunakan untuk menampilkan didalam loop `foreach()`.

Berikutnya akan ditunjukkan `foreach()` untuk menampilkan indeks dan nilai dari array yang dibuat.

```
1 <?php
2 $data=array("Surabaya","Banjarmasin","Jakarta");
3 foreach($data as $d=>$val)
4 {
5     echo "key= ".$d.", val= ".$val;
6     echo "<br>";
7 }
8 ?>
```



key= 0, val= Surabaya
key= 1, val= Banjarmasin
key= 2, val= Jakarta

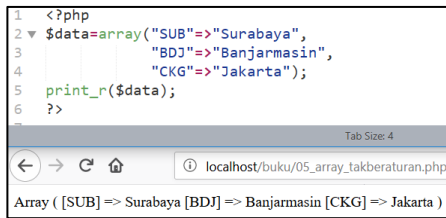
Gambar 5. Foreach() Untuk Menampilkan Key Dan Value

Perbedaan perintah diatas ada pada perintah `foreach($data as $d)` dan `foreach($data as $d=>$val)`.

1.5. Array dengan Indeks Tidak Beraturan (*Assosiative*)

Selain beraturan, kunci atau indeks array pada bahasa php dapat dibuat tanpa harus beraturan, atau dapat disebut *assosiative array*. Artinya kita mendefinisikan sendiri kuncinya. Hal ini untuk mempermudah membaca isi value. Pada contoh di atas, bandar udara Banjarmasin disingkat BJD, Jakarta disingkat CKG (Cengkareng) dan seterusnya. Untuk kasus seperti ini, lebih tepat jika arraynya menggunakan array kunci sendiri atau array dengan kunci tidak beraturan.

Berikut contoh program untuk membuat array tidak beraturan.



```
1 <?php
2 $data=array("SUB"=>"Surabaya",
3           "BDJ"=>"Banjarmasin",
4           "CKG"=>"Jakarta");
5 print_r($data);
6 ?>
```

localhost/buku/05_array_takberaturan.php

Array ([SUB] => Surabaya [BDJ] => Banjarmasin [CKG] => Jakarta)

Gambar 6. Membuat Array Tak Beraturan.

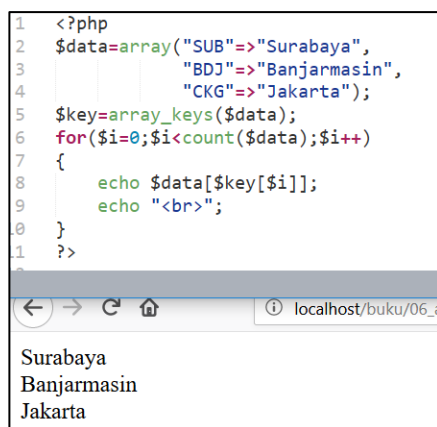
Penulisan array ke bawah ditujukan untuk mempermudah pembacaan nilai array. Meskipun jika anda menuliskan mendatar pun tidak mengapa.

```
$data=array("SUB"=>"Surabaya",
"BDJ"=>"Banjarmasin", "CKG"=>"Jakarta");
```

Terserah selera *programmer* masing-masing.

1.6. Mencetak Array dengan FOR

Untuk array tidak beraturan dengan for kurang bijaksana. Karena indexnya tidak lagi urut, dan untuk menampilkannya, kita harus mengambil indeksnya terlebih dahulu dengan perintah `array_keys()`.



```
1 <?php
2 $data=array("SUB"=>"Surabaya",
3           "BDJ"=>"Banjarmasin",
4           "CKG"=>"Jakarta");
5 $key=array_keys($data);
6 for($i=0;$i<count($data);$i++)
7 {
8     echo $data[$key[$i]];
9     echo "<br>";
10 }
11 ?>
```

localhost/buku/06_a

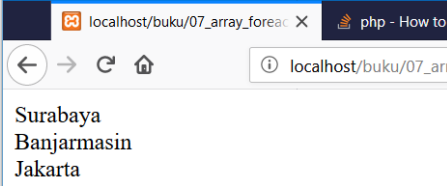
Surabaya
Banjarmasin
Jakarta

Gambar 7. Menampilkan Isi Array dengan Perintah for()

1.7. Mencetak Array dengan FOREACH

Sama dengan contoh sebelumnya, array tidak beraturan atau *array assosiatif* lebih mudah ditampilkan dengan `foreach()`. Dapat value saja atau key value.

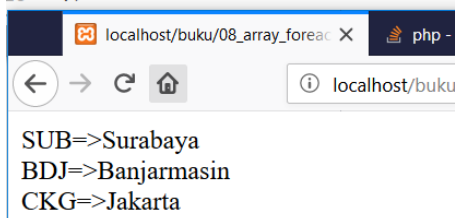
```
1 <?php
2 $data=array("SUB"=>"Surabaya",
3             "BDJ"=>"Banjarmasin",
4             "CKG"=>"Jakarta");
5 foreach($data as $d)
6 {
7     echo $d;
8     echo "<br>";
9 }
10 ?>
```



Gambar 8. Mencetak Value Array Menggunakan Foreach()

Dan berikut ini perintah `foreach()` untuk menampilkan key dan value.

```
1 <?php
2 $data=array("SUB"=>"Surabaya",
3             "BDJ"=>"Banjarmasin",
4             "CKG"=>"Jakarta");
5 foreach($data as $key=>$d)
6 {
7     echo $key."=>".$d;
8     echo "<br>";
9 }
10 ?>
```



Gambar 9. Foreach() untuk Menampilkan Key dan Value

1.8. Latihan

1. Buatlah *coding* untuk membuat sebuah variabel array *indexed* atau beraturan yang berisi 3 nama buah. Tampilkan dengan perintah `print_r()`?
2. Error apa yang terjadi jika array no 1 di cetak dengan perintah `echo()`?
3. Buatlah dua perulangan untuk menampilkan array no 1 dengan `for` dan `foreach`.
4. Buatlah *coding* untuk membuat sebuah variabel array yang berisi 3 nama buah. Tampilkan dengan perintah `print_r()`?
5. Buatlah *coding* untuk membuat sebuah variabel *associative array* atau array tak beraturan yang berisi 5 buah singkatan dan kepanjangannya. Singkatan disimpan sebagai *key*, kepanjangannya disimpan sebagai *value*. Tampilkan dengan perintah `print_r()`?
6. Cetak dengan perintah `foreach()`, *key* dan *valuenya*.

BAB 2

OPERASI ARRAY

Capaian Pembelajaran:

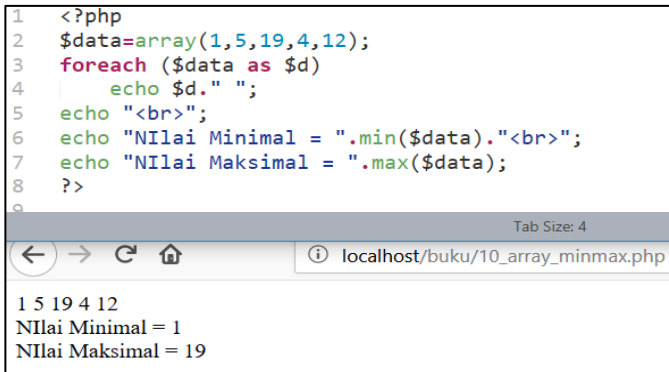
1. Mampu memahami operasi-operasi array.
2. Mampu memahami pencarian nilai *min*, *max*, *push*, *slice*, *sort* dan *explode*.
3. Mampu mengimplementasikan operasi array.

Dalam proses pemrograman, akan ada banyak manipulasi array untuk mendapatkan hasil data sesuai yang diinginkan.

2.1. Mencari Nilai Min Max di Array

Pada data array angka, pencarian nilai minimal dan maksimal sangat mudah dilakukan. Hal ini dikarenakan php sudah menyediakan library untuk mencari nilai minimal dan maksimal. Perintahnya adalah `min()` dan `max()`.

```
1 <?php
2 $data=array(1,5,19,4,12);
3 foreach ($data as $d)
4     echo $d." ";
5 echo "<br>";
6 echo "Nilai Minimal = ".min($data)."<br>";
7 echo "Nilai Maksimal = ".max($data);
8 ?>
```



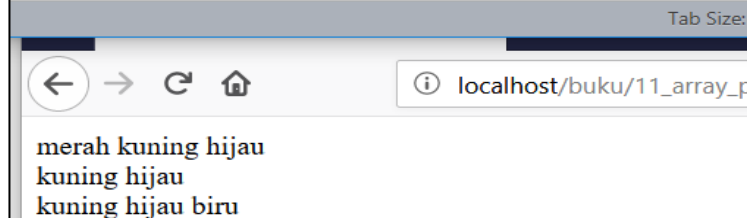
1 5 19 4 12
Nilai Minimal = 1
Nilai Maksimal = 19

Gambar 10. Mencari Nilai Minimal dan Maksimal

2.2. Menyisipkan Atau Menghapus Elemen Array.

Untuk keperluan menyisipkan kita dapat menggunakan perintah `push()`, dan untuk menghapus elemen array dapat menggunakan perintah `slice()`

```
1 <?php
2 $data = array("merah","kuning","hijau");
3 foreach ($data as $d)
4     echo $d." ";
5 echo "<br>";
6
7 $data = array_slice($data,1);
8 foreach ($data as $d)
9     echo $d." ";
10 echo "<br>";
11
12 array_push($data, "biru");
13 foreach ($data as $d)
14     echo $d." ";
15 echo "<br>";
16 ?>
```



Gambar 11. Memotong dan menambah elemen array.

Perintah `push()`, berguna untuk menambah elemen array pada bagian akhir. Sedangkan pada perintah `slice()`, formatnya adalah sebagai berikut.

```
Array_slice(array, start, length, preserve);
```

Array = Nama Array

Start = Posisi yang akan di potong


Length = Panjang array yang di ambil. Jika di kosongkan maka sisa array akan diambil.

2.3. Mengurutkan Isi Array.

Ada perbedaan untuk melakukan pengurutan antara *indexed* dan *assosiatif array*. Hal ini dikarenakan pada *assosiatif* ada key pada array yang tidak urut seperti *indexed array*. Untuk *ascending* dan *descending* dibedakan pada perintah, bukan parameter.

Berikut *coding* untuk melakukan urut pada *indexed array*.

```
1 <?php
2 $data = array("Kambing", "Ayam", "Sapi");
3 foreach ($data as $d)
4     echo $d." ";
5 echo "<br>";
6
7 sort($data);
8 foreach ($data as $d)
9     echo $d." ";
10 echo "<br>";
11
12 rsort($data);
13 foreach ($data as $d)
14     echo $d." ";
15 echo "<br>";
16 ?>
```



Gambar 12. Mengurutkan Indexed Array .

Selanjutnya akan diberikan contoh untuk mengurutkan *array assosiatif* secara *ascending* dan *descending* berdasarkan *value*-nya.


```

1 <?php
2 $data=array("1"=>"Kambing",
3           "3"=>"Ayam",
4           "2"=>"Sapi");
5 print_r($data);
6 echo "<br>";
7
8 asort($data);
9 print_r($data);
10 echo "<br>";
11
12 arsort($data);
13 print_r($data);
14 echo "<br>";
15 ?>
16

```

Gambar 13. Mengurutkan assosiatif array berdasar value.

Dan ini adalah perintah untuk mengurutkan *indexed array* berdasarkan *key*-nya.

```

1 <?php
2 $data=array("1"=>"Kambing",
3           "3"=>"Ayam",
4           "2"=>"Sapi");
5 print_r($data);
6 echo "<br>";
7
8 ksort($data);
9 print_r($data);
10 echo "<br>";
11
12 krsort($data);
13 print_r($data);
14 echo "<br>";
15 ?>
16

```

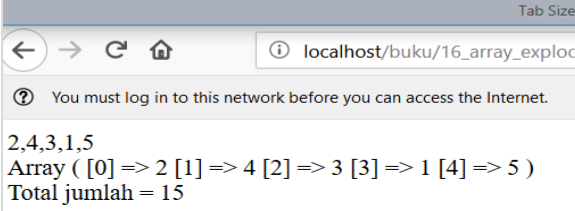
Gambar 14. Mengurutkan Assosiatif Array Berdasar Key.

2.4. Memecah String Menjadi Array.

Input user biasanya berbentuk *string*. Dan untuk proses tertentu, kita harus memecah *string* tersebut agar dapat menjadi array untuk proses lebih lanjut.

Sebagai contoh, ada *string* yang terdiri dari 5 angka yang dipisahkan koma. Dari *string* tersebut akan kita konversi menjadi array dan kita hitung nilai jumlah angka-angkanya. Perintah memecah itu adalah `explode()`.

```
1 <?php
2 $teks="2,4,3,1,5";
3 echo $teks."<br>";
4
5 $data=explode(",",$teks);
6
7 print_r($data);echo "<br>";
8
9 echo "Total jumlah = ".array_sum($data);
10 ?>
```



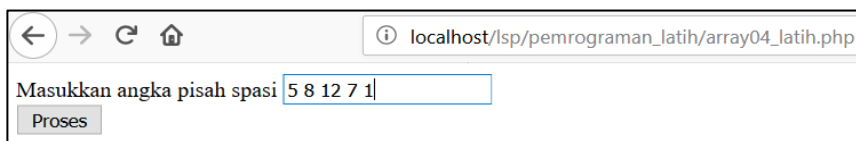
2,4,3,1,5
Array ([0] => 2 [1] => 4 [2] => 3 [3] => 1 [4] => 5)
Total jumlah = 15

Gambar 15. Perintah Memecah String Menjadi Array. `explode()`

2.5. Contoh Array Kasus dengan HTML

Untuk lebih memahami aplikasi sederhana, maka akan dibuatkan contoh *entry* data *string* dari HTML, di proses menjadi array dan di urutkan.

Masukkan beberapa angka. Tampilkan angka apa adanya dan yang sudah di urut seperti dibawah ini.



Masukkan angka pisah spasi

Proses

Masukkan angka pisah spasi

Proses

Data Yang Di input =
5
8
12
7
1

Data Yang Di Urut =
1
5
7
8
12

Cobalah kerjakan dulu sebelum melihat jawabannya di sebelah.

```

1  <HTML>
2  <body>
3      <form method=POST>
4          Masukkan angka pisah spasi
5          <input type=text name=angka><br>
6          <input type="submit" value="Proses">
7      </form>
8      <?php
9          if(!empty($_POST['angka'])){
10             $data=$_POST['angka']; //Ambil variabel POST
11             $data=explode(" ",$data); //Pecah ke array
12             echo "Data Yang Di input = <br>";
13             foreach ($data as $d) { //Tampilkan
14                 echo $d;
15                 echo "<br>";
16             }
17             echo "<hr>";
18             echo "Data Yang Di Urut = <br>";
19             sort($data);
20             foreach ($data as $d) {
21                 echo $d;
22                 echo "<br>";
23             }
24         }
25     ?>
26 </body>
27 </HTML>

```

2.6. Latihan

1. Buatlah coding untuk membuat sebuah variabel array *indexed* atau beraturan yang isinya:
satu, dua, tiga, empat, lima
Tampilkan dengan perintah `print_r()`?
2. Tuliskan perintah *slice* yang tepat agar tersisa:
dua, tiga, empat
3. Lanjutkan lagi, tuliskan perintah agar array menjadi:
dua, tiga, empat, sepuluh
4. Buatlah sebuah aplikasi sederhana untuk memasukkan angkatan-angka dan proseslah untuk menampilkan nilai rata-ratanya.

BAB 3

SUBPROGRAM

Capaian Pembelajaran:

1. Mampu memahami konsep subprogram.
2. Mampu memahami prosedur dan fungsi.
3. Mampu membuat aplikasi dengan fungsi dan prosedur.

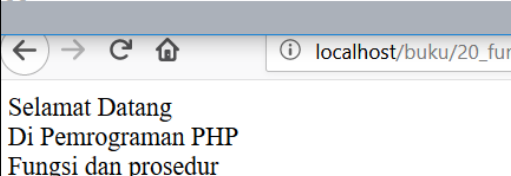
Saat *coding* yang diketikkan memiliki banyak kesamaan, maka baris *coding-coding* yang sama tadi dapat dijadikan satu menjadi sebuah sub program.

Sub program sendiri dibagi menjadi dua, fungsi dan prosedur. Secara sederhana mereka memiliki perbedaan. Prosedur tidak mengembalikan nilai, tidak ada perintah *return*. Sedangkan fungsi mengembalikan nilai, ada perintah *return*. Dalam PHP, hanya mengenal fungsi, sehingga semua subprogram bernama `function nama_fungsi()`.

3.1. Fungsi Sederhana Tanpa Variabel

Sebuah fungsi sederhana dapat saja tanpa mengirim variabel sama sekali. Seperti contoh dibawah ini.

```
1 <?php
2 echo "Selamat Datang";enter();
3 echo "Di Pemrograman PHP";enter();
4 echo "Fungsi dan prosedur";
5
6 function enter()
7 {
8     echo "<br>";
9 }
10 ?>
```

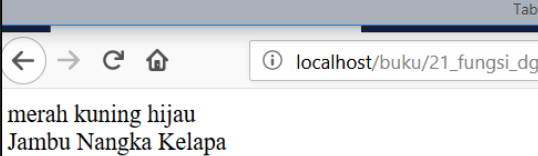


Gambar 16. Function Sederhana

3.2. Fungsi Dengan Mengirim Variabel

Berikutnya, akan diberikan contoh sebuah fungsi sederhana untuk mengirimkan sebuah variabel array dan mencetaknya ditambah *enter* di akhir fungsi.

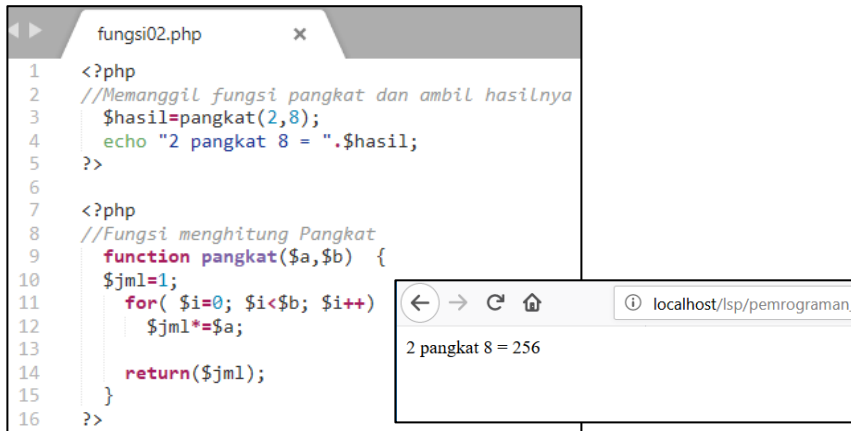
```
1 <?php
2 $a = array("merah","kuning","hijau");
3 $b = array("Jambu","Nangka","Kelapa");
4 cetak($a);
5 cetak($b);
6
7 function cetak($data)
8 {
9     foreach ($data as $d)
10         echo $d." ";
11     echo "<br>";
12 }
13 ?>
```



Gambar 17. Function Sederhana dengan Mengirim Variabel

3.3. Fungsi dengan Mengembalikan Variabel

Selanjutnya adalah fungsi yang sering sekali dipakai, yaitu fungsi yang mengembalikan variabel ke pemanggilnya.



```
1 <?php
2 //Memanggil fungsi pangkat dan ambil hasilnya
3 $hasil=pangkat(2,8);
4 echo "2 pangkat 8 = ".$hasil;
5 ?>
6
7 <?php
8 //Fungsi menghitung Pangkat
9 function pangkat($a,$b) {
10     $jml=1;
11     for( $i=0; $i<$b; $i++)
12         $jml*=$a;
13
14     return($jml);
15 }
16 ?>
```

localhost/lsp/pemrograman_...
2 pangkat 8 = 256

Gambar 18. Function yang Mengembalikan Nilai

3.4. Latihan

1. Buatlah program untuk memasukkan angka-angka, carilah nilai rata-ratanya. Gunakan Fungsi.

BAB 4

AKSES FILE

Capaian Pembelajaran:

1. Mampu memahami prinsip akses berkas (*file*).
2. Mampu memahami *mode write* dan *read* File
3. Mampu membuat *coding* untuk menulis dan membaca ke File

Proses penyimpanan data di RAM komputer akan hilang jika komputer dimatikan. Salah satu cara agar tidak hilang adalah dengan menyimpan datanya dalam bentuk file.

4.1. Menyimpan String ke File

Untuk menyimpan file, ada dua hal utama yang harus disiapkan. *Pointer* ke file sekaligus membuat file dengan perintah `fopen()`, dan format data yang akan disimpan.

Pointer ke file menggunakan perintah `fopen()`, dan data yang disimpan menggunakan String.

```
1 <?php
2 $myfile = fopen("data01.txt", "w") or die("Unable to open file!");
3 $txt = "Belajar Akses File\nMenuliskan Data";
4 fwrite($myfile, $txt);
5 fclose($myfile);
6 ?>
```



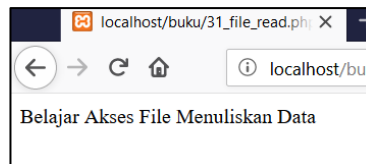
Gambar 19. Akses File Menyimpan String ke File TXT

4.2. Membaca File Text ke String

Untuk membaca file, ada dua hal utama yang harus disiapkan. Pointer ke file sekaligus mencari file dengan perintah `fopen()`, dan format data yang akan disimpan.

Pointer ke file menggunakan perintah `fopen()`, dan data yang disimpan menggunakan string.

```
1 <?php
2 $myfile = fopen("data01.txt", "r") or die("Unable to open file!");
3 echo fread($myfile, filesize("data01.txt"));
4 fclose($myfile);
5 ?>
```



Gambar 20. Akses file Membaca String dari File TXT

Tampak ada permasalahan antara notepad dan *browser*. Hal itu terjadi karena notepad menggunakan `/r/n` untuk ganti baris. Sedangkan *browser* menggunakan `
` untuk ganti baris. Untuk itu harus dilakukan konversi, perintah phpnya adalah `nl2br`.

Ganti baris 3 menjadi:

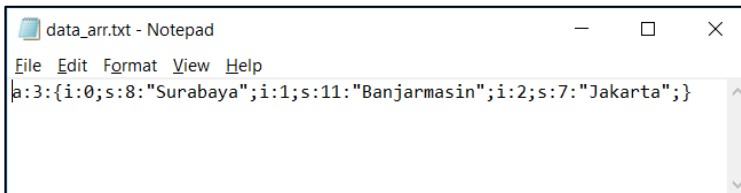
```
3 echo nl2br(fread($myfile, filesize("data01.txt")));
```

4.3. Menyimpan Array ke File

Dari awal kita sudah membahas array, dan untuk menyimpannya memerlukan teknik khusus, yaitu perintah `serialize()`, dan menyimpan dengan perintah `file_put_contents()`.

Perhatikan bahwa file teks yang dibuat menyimpan data-data yang agak berbeda dengan array yang biasa kita lihat di *browser*.

```
1 <?php
2     $data=array("Surabaya","Banjarmasin","Jakarta");
3     print_r($data);
4     $simpan = serialize($data);
5     file_put_contents('data_arr.txt', $simpan);
6 ?>
```

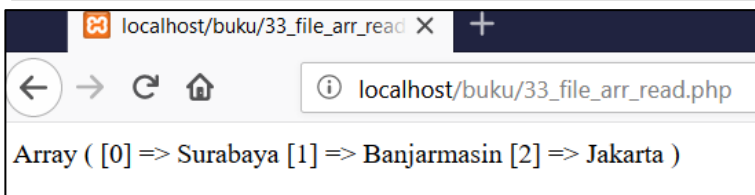


Gambar 21. Akses file Menyimpan Array ke File txt

4.4. Membaca file ke array

Sebaliknya, untuk membaca file ke array, kita akan menggunakan perintah `file_get_contents()` ;.

```
1 <?php
2     $simpan = file_get_contents('data_arr.txt');
3     $data = unserialize($simpan);
4     print_r($data);
5 ?>
```



Gambar 22. Akses File Membaca File Text ke Array

4.5. Latihan

1. Buatlah program untuk memasukkan teks berupa kata-kata. Kemudian simpanlah kedalam sebuah File.
2. Buatlah program untuk membaca file teks yang telah dibuat dilatihan Nomor 1.

BAB 5

FRAMEWORK CODEIGNITER (CI)

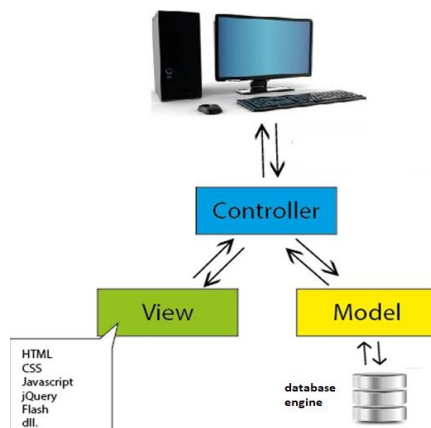
Capaian Pembelajaran:

1. Mampu memahami Konsep Framework CI
2. Mampu memahami Konsep MCV
3. Mampu Menginstalasi CI
4. Membuat Aplikasi Controller View sederhana

CI adalah sebuah *framework* PHP yang membantu kita untuk mempercepat dan memilah *coding* menjadi terstruktur dan mudah dipahami.

5.1. Konsep MCV (Model, Controller, View)

CI bersifat *open source* dan menggunakan metode MCV (*Model, Control, View*).



Gambar 23. Konsep MCV

Secara sederhana, penjelasan cara kerja dan peran masing-masing adalah sebagai berikut:

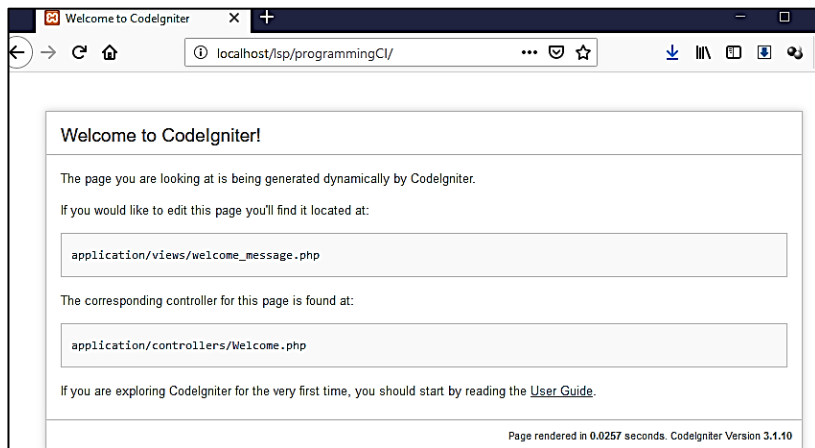
- *Controller* : Sebagai kontrol Pengatur utama. *Controller* inilah yang mengatur lintas data dari *interface* HTML ke database dan sebaliknya.
Menurut aturan, Jangan ada echo, atau query disini.
- *Model* : Khusus memanggil atau menulis data ke tabel. Data-data dari tabel langsung diarahkan *Controller*. Dan tidak ke *view*.
- *View* : *Coding* khusus HTML. Semua tampilan di web, dituliskan dibagian *view* ini. *View* menerima dan mengirim data langsung ke *controller*. Sebaiknya tidak langsung ke *model*.

5.2. Instalasi CI

Download CI di website: <https://www.codeigniter.com/>

Extract file nya langsung ke folder yang htdocs. Biasanya akan dibuatkan *folder* baru sesuai versi yang di-*download*, misalkan CodeIgniter-3.1.10. Rename folder tersebut sesuai dengan keinginan anda.

Setelah itu buka *browser* dan arahkan ke folder yang baru dibuat tadi, maka akan muncul tampilan sebagai berikut.



Gambar 24. Tampilan Awal Codeigniter Setelah di Install Pertama Kali

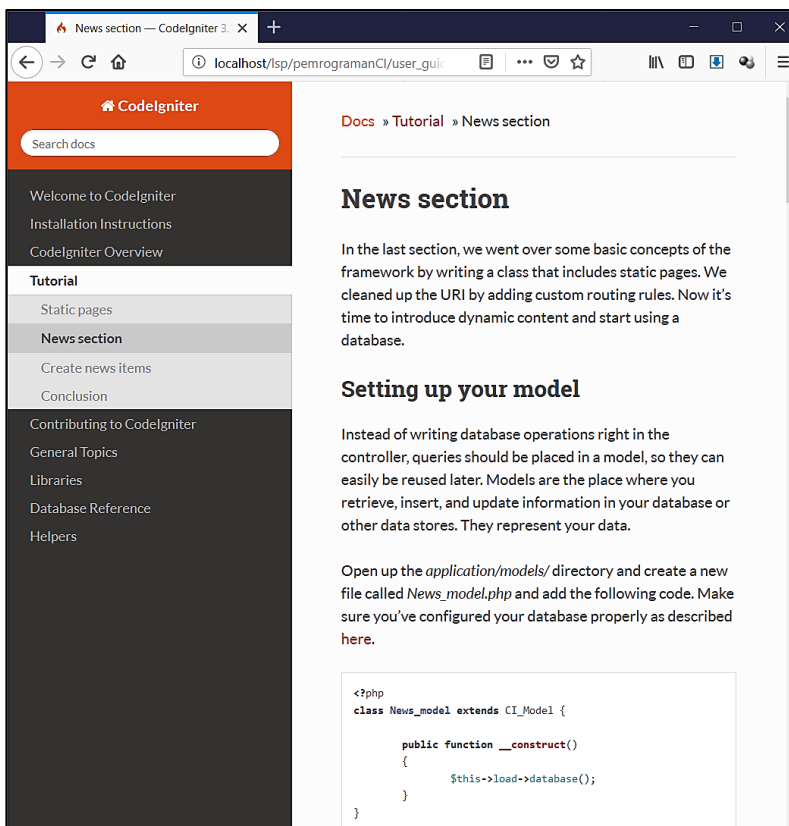
Contoh diatas, *folder project* ada di

C:\xampp\htdocs\lsp\programmingCI

Dan untuk memanggilnya di, url ketikkan :

http://localhost/lsp/pemrogramanCI/

Jika anda ingin benar-benar menguasai CI, bukalah *user_guide* yang linknya ada di kanan bawah, dan bacalah semuanya, karena semua informasi, aturan, fungsi dan *library codeigniter* sudah tertuliskan lengkap di *user_guide*.

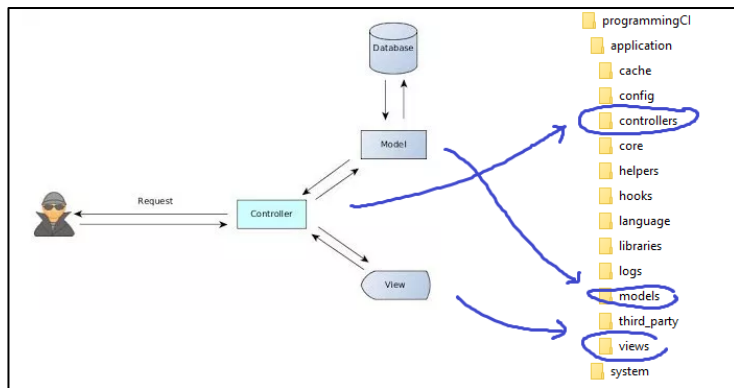


Gambar 25. Tampilan *user_guide* yang Sangat Simple Dan Lengkap.

Untuk memulai *coding*, Bagian Tutorial sub *News Section* sangat baik untuk memulai pemahaman tentang Codeigniter. Sedangkan untuk database, *Database Reference sub Query Builder Class* sangat tepat sebagai referensi.

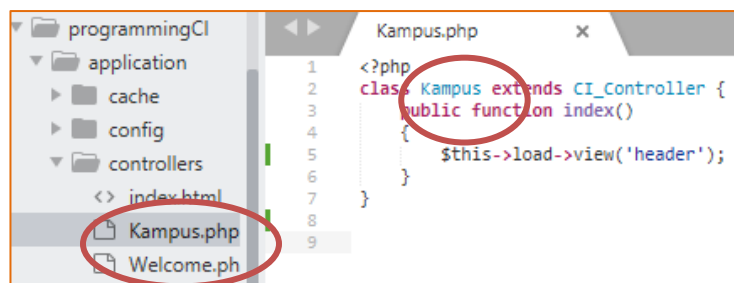
5.3. Struktur Folder CI

Hubungan MCV dengan *folder* sangatlah jelas dan mudah dipahami. Hal ini dikarenakan nama *folder* sama dengan singkatan MCV.



Gambar 26. Folder CI dan MCV

Satu-satunya yang harus diperhatikan dalam penulisan nama file adalah khusus untuk *controllers*. Nama file harus diawali dengan huruf besar. Begitu juga nama *Class*-nya harus huruf besar. Nama file yang lain huruf kecil semua.



Gambar 27 Nama file dan Class harus diawali huruf besar.

5.4. Memulai CI

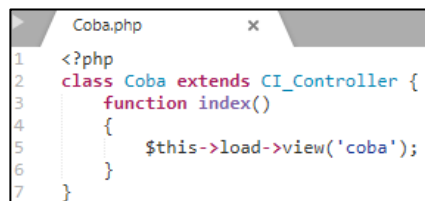
Langkah pertama untuk menampilkan tulisan sederhana di *browser*, yang sesuai dengan kaidah CI adalah dengan membuat dua buah file. Yaitu file *controller* dan *view*.

Controller akan di eksekusi pertama kali oleh url, dan *controller* memerintahkan *view* untuk menampilkan tulisan.

- 1) Buatlah sebuah file controller di folder:

```
application/controllers/Coba.php
```

Perhatikan, nama file controller HURUF BESAR. View huruf kecil saja. Ketikkan *coding* berikut:



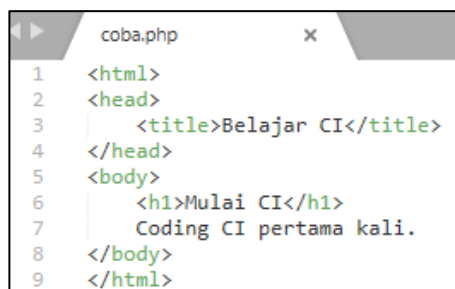
```
Coba.php x
1 <?php
2 class Coba extends CI_Controller {
3     function index()
4     {
5         $this->load->view('coba');
6     }
7 }
```

Gambar 28 Coding Controller, Nama File Dan Class Huruf Besar

- 2) Buatlah sebuah file *view* di *folder*:

```
application/view/coba.php
```

Ketikkan *coding* berikut:

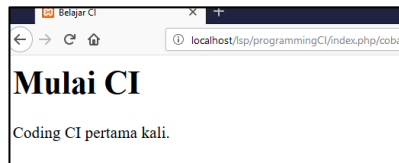


```
coba.php x
1 <html>
2 <head>
3     <title>Belajar CI</title>
4 </head>
5 <body>
6     <h1>Mulai CI</h1>
7     Coding CI pertama kali.
8 </body>
9 </html>
```

Gambar 29 Coding View, Nama File dan Class Huruf Biasa.

3) Buka browser dan jalankan ketikkan di url:

localhost/lsp/programming/index.php/coba



5.5. Mengirim Data dari Controller ke View

Yang sering terjadi adalah data akan dikirim dari *controller* ke *view* agar *view* dapat menampilkan data-data yang diperlukan. Berikut contoh menampilkan data Judul yang dikirim ke *view*.

1) application/controllers/Coba.php

```
Coba.php x
1 <?php
2 class Coba extends CI_Controller {
3     function index()
4     {
5         $data['judul']='Memulai CI';
6         $this->load->view('coba',$data);
7     }
8 }
```

Gambar 30 Data Array di Controller Dikirim ke View.

2) application/view/coba.php

```
coba.php x
1 <html>
2 <head>
3     <title>Belajar CI</title>
4 </head>
5 <body>
6     <h1><?php echo $judul; ?></h1>
7     Coding CI pertama kali.
8 </body>
9 </html>
```

Gambar 31 Coding View, Menampilkan Data \$Judul.

- 3) Buka *browser* dan jalankan ketikkan di url:

localhost/lsp/programming/index.php/coba



Gambar 32 Hasil tampilan coding CI pertama kali.

5.6. Latihan

1. Perhatikan *controller* berikut yang memiliki dua variabel:

```
$data['kode_mk']='A0123';  
$data['nama_mk']='Algoritma dan Pemrograman Dasar';
```

Buatlah *coding view* untuk menampilkan kedua variabel ini.

2. Sebuah *controller* memiliki variabel array seperti dibawah ini.

```
<?php  
class Coba extends CI_Controller {  
    function index()  
    {  
        $kota=array("Surabaya","Banjarmasin","Jakarta");  
        $data['nama_kota']=$kota;  
        $this->load->view('coba',$data);  
    }  
}
```

Buatlah *coding view* untuk menampilkan isi array agar muncul sebagai berikut:

```
Surabaya  
Banjarmasin  
Jakarta
```


BAB 6

TEMPLATE

Capaian Pembelajaran:

1. Mampu memahami cara kerja *template* html pada CI.
2. Mampu memahami cara membangun *template*.
3. Mampu membuat aplikasi sesuai *template* yang dibangun.

Template adalah sebuah susunan file yang dirancang sedemikian rupa agar mudah di duplikasi dan dikembangkan untuk kepentingan *project-project* serupa. *Template* disusun dengan nama-nama yang sangat mudah dipahami dan diingat.

6.1. Membuat Template

Berikutnya kita akan membuat *template* untuk menampilkan sistem informasi kita. Sekaligus, kita akan melakukan efisiensi program dengan menggunakan prosedur dan menggunakan *reuser function*.

6.1.1. Config

Bukalah folder

```
C:\xampp\htdocs\lsp\programmingCI\application\  
config\  

```

Lakukan pengeditan beberapa baris berikut:

- 1) `config.php` → baris 26.
`$config['base_url'] =
'http://localhost/lsp/programmingCI';`

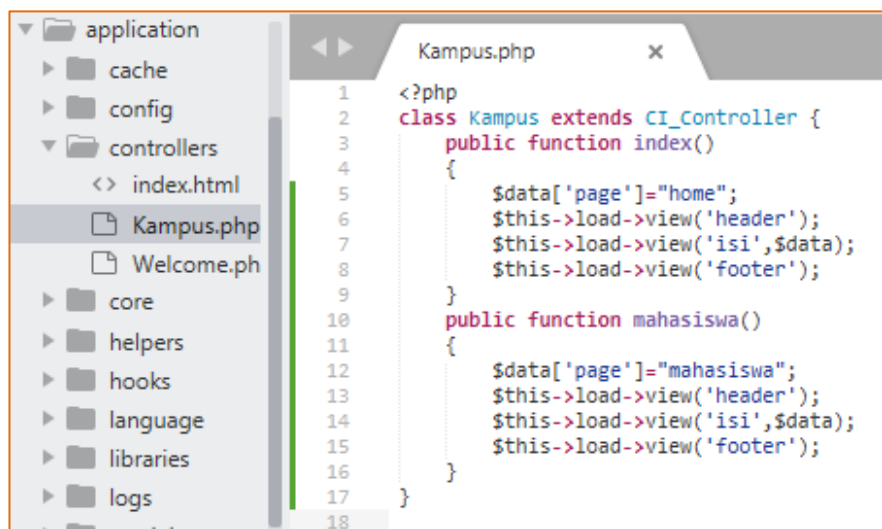
2) autoload.php → baris 92.
`$autoload['helper'] = array('url');`

6.1.2. Controller

Bukalah folder

`C:\xampp\htdocs\lsp\programmingCI\application\controller\`

Buatlah file Kampus.php



```
1 <?php
2 class Kampus extends CI_Controller {
3     public function index()
4     {
5         $data['page']="home";
6         $this->load->view('header');
7         $this->load->view('isi',$data);
8         $this->load->view('footer');
9     }
10    public function mahasiswa()
11    {
12        $data['page']="mahasiswa";
13        $this->load->view('header');
14        $this->load->view('isi',$data);
15        $this->load->view('footer');
16    }
17 }
18
```

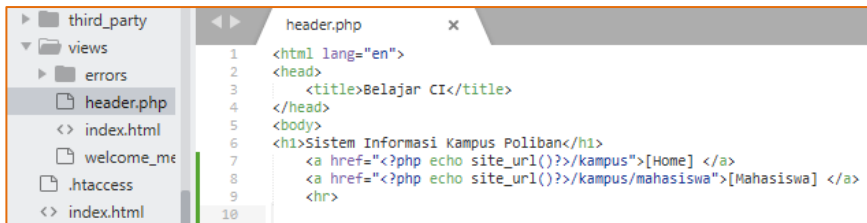
Gambar 33 Coding Controller Kampus, Kontrol Utama di CI.

6.1.3. View

Bukalah folder

`C:\xampp\htdocs\lsp\programmingCI\application\view\`

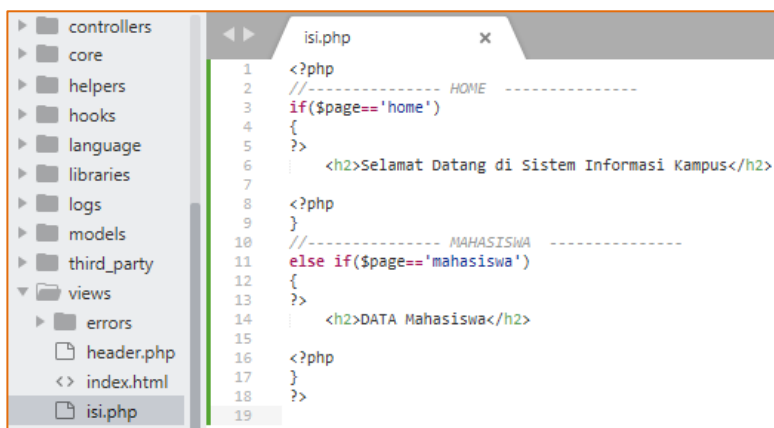
Buatlah file header.php



```
1 <html lang="en">
2 <head>
3   <title>Belajar CI</title>
4 </head>
5 <body>
6   <h1>Sistem Informasi Kampus Poliban</h1>
7   <a href="<?php echo site_url()?>/kampus">[Home] </a>
8   <a href="<?php echo site_url()?>/kampus/mahasiswa">[Mahasiswa] </a>
9   <hr>
10
```

Gambar 34 Menampilkan Data Judul dan Navigasi

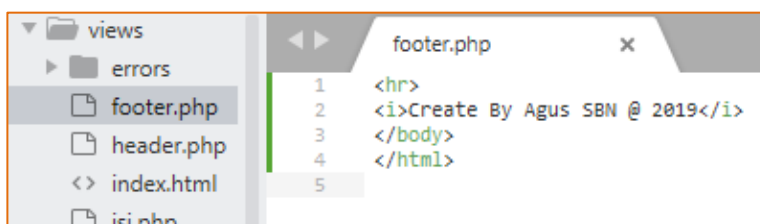
Buatlah file isi.php



```
1 <?php
2 //----- HOME -----
3 if($page=='home')
4 {
5   ?>
6   <h2>Selamat Datang di Sistem Informasi Kampus</h2>
7
8   <?php
9 }
10 //----- MAHASISWA -----
11 else if($page=='mahasiswa')
12 {
13   ?>
14   <h2>DATA Mahasiswa</h2>
15
16 <?php
17 }
18 ?>
19
```

Gambar 35 Menampilkan Isi Sesuai dengan Menu Terpilih

Buatlah file footer.php



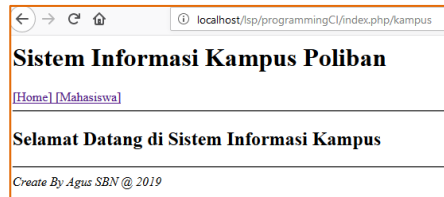
```
1 <hr>
2 <i>Create By Agus SBN @ 2019</i>
3 </body>
4 </html>
5
```

Gambar 36 Menampilkan Halaman Footer.

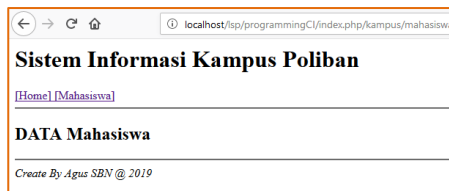
6.1.4. Jalankan

Bukalah Browser dan ketikkan di URL:

`http://localhost/lsp/programmingCI/index.php/kampus`



Gambar 37 Tampilan Awal -> Home



Gambar 38 Tampilan Mahasiswa Setelah Click Mahasiswa

Coba perhatikan cara kerja *template* dan konsep di atas. Karena ini adalah kerangka dasar cara berpikir Codeigniter kita selanjutnya. Berikutnya kita akan membuat prosedur sendiri dan melakukan efisiensi *coding*.

6.2. Membuat Prosedur dan Efisiensi

Perhatikan *controller* Kampus.php, ada kesamaan perintah pada dua fungsi *index* dan *mahasiswa*. Baris 6, 7, 8 sama dengan baris 13, 14, 15. Tiga baris ini dapat dibuat menjadi satu baris saja menjadi sebuah fungsi baru. Kita beri nama saja fungsi baru ini adalah `function tampil()`. Karena baris 7 dan baris 14 mengirim variabel data, maka fungsi ini juga kita siapkan untuk menerima data sehingga menjadi `function tampil($data)`

Hal ini jelas akan menghemat *coding* dan baris program.

[controllers/Kampus.php](#)

```
1 <?php
2 class Kampus extends CI_Controller {
3     public function index()
4     {
5         $data['page']="home";
6         $this->tampil($data);
7     }
8     public function mahasiswa()
9     {
10        $data['page']="mahasiswa";
11        $this->tampil($data);
12    }
13    function tampil($data)
14    {
15        $this->load->view('header');
16        $this->load->view('isi',$data);
17        $this->load->view('footer');
18    }
19 }
```

6.3. Latihan

Buatlah sebuah *template* untuk menampilkan *link web* sederhana, namun dengan tombol navigasi di sebelah kiri. Gunakanlah teknik *template* yang telah dijelaskan di bab ini.

BAB 7

MENAMPILKAN ISI TABEL (READ)

Capaian Pembelajaran:

1. Mampu membuat database dan tabel di MySQL.
2. Mampu memahami alur CI untuk menampilkan tabel.
3. Mampu membuat coding untuk menampilkan tabel.

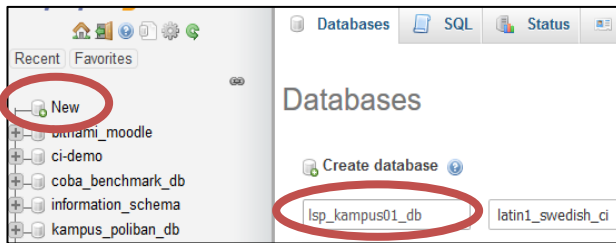
Dalam materi ini tidak dibahas tentang database MySQL. Diasumsikan sudah memahami perintah-perintah DDL dan DML pada MySQL. Mulai dari *Create*, *insert*, *update* dan *delete*. Juga tidak dibahas tentang aplikasi untuk mengakses MySQL. Dibuku ini digunakan *phpmyadmin* sebagai *tools* untuk menangani tabel MySQL.

7.1. Membuat Database Kampus Dan Tabel Mahasiswa

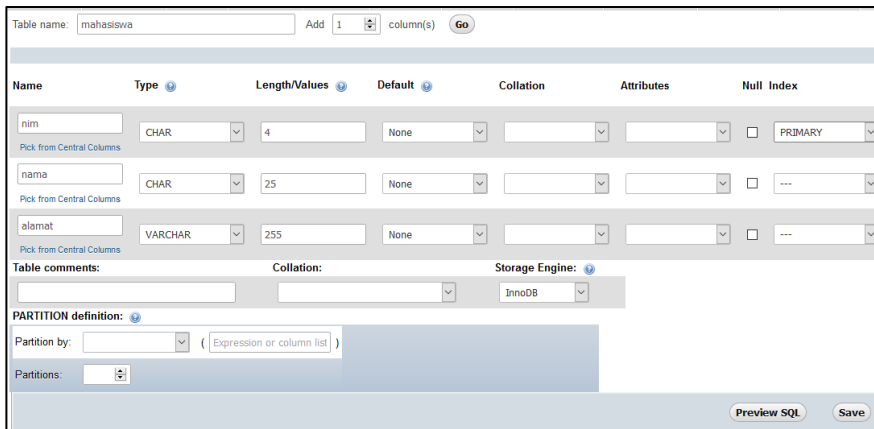
Bukalah *phpmyadmin* dari *browser* anda dan buatlah database dan tabel sebagai berikut.

- **Nama Database** : lsp_kampus01_db
- **Nama Tabel** : mahasiswa
 - **Field :**

nim	char	4	primary
nama	char	25	
alamat	varchar	255	

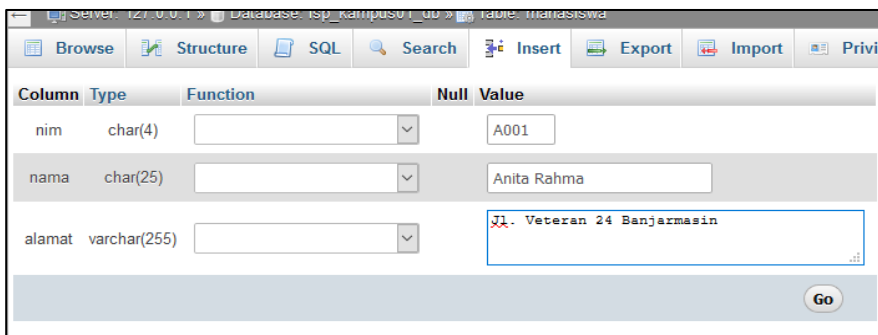


Gambar 39. Tampilan Phpmyadmin untuk Membuat Database



Gambar 40. Tampilan Phpmyadmin untuk Membuat Tabel Mahasiswa

Selanjutnya, agar *coding view* dapat berjalan, isikan tabel yang baru saja dibuat dengan sebuah *record* atau lebih.



Gambar 41. Menambahkan sebuah record pada tabel mahasiswa

Contoh pengisian sampai 3 tabel yang nanti akan ditampilkan pada *coding* CI.

Options		nim		nama		alamat	
<input type="checkbox"/>	Edit	Copy	Delete	A001	Salsabila Anita Rahma	Jl. Veteran 24 Banjarmasin	
<input type="checkbox"/>	Edit	Copy	Delete	A005	Naufal	Jl. Perdagangan No14 Banjarmasin	
<input type="checkbox"/>	Edit	Copy	Delete	A006	Siti Nur Rahma	Jl. Sulatan Adam 29 Banjarmasin	

7.2. Coding CI Untuk Menampilkan Isi Tabel

Berikutnya akan dijelaskan cara menampilkan isi tabel mahasiswa. *Controller* akan memanggil *model*, dan mengambil data tabel mahasiswa dan menyimpan dalam bentuk array. Data array ini kemudian dikirimkan ke *view* untuk ditampilkan di *view*.

7.2.1. Config

1) config/autoload.php

- Baris (61) :
`$autoload['libraries'] = array('database');`
- Baris (135) :
`$autoload['model'] = array('m_kampus');`

2) config/database.php →

- ```
'hostname' => 'localhost',
'username' => 'root',
'password' => '',
'database' => 'lsp_kampus01_db',
'dbdriver' => 'mysqli',
```

### 7.2.2. Controller

Modifikasi *Controller*. Untuk mengambil data dari model, cukup hanya menambah 1 baris saja. Keren kan.. 😊

[controllers/Kampus.php](#)

```
Kampus.php x
1 <?php
2 class Kampus extends CI_Controller {
3 public function index()
4 {
5 $data['page']="home";
6 $this->tampil($data);
7 }
8 public function mahasiswa()
9 {
10 $data['page']="mahasiswa";
11 $data['dt']=$this->m_kampus->ambil_data_mahasiswa();
12 $this->tampil($data);
13 }
14 function tampil($data)
15 {
16 $this->load->view('header');
17 $this->load->view('isi',$data);
18 $this->load->view('footer');
19 }
20 }
```

### 7.2.3. Model

Untuk pertama kali, kita akan mengakses *folder model*. Buatlah file dan ketikkan seperti berikut.

models/M\_kampus.php

**(Perhatikan, Awalan nama File dan nama controller Huruf Besar)**

```
models
 <> index.html
 M_kampus.p
 third_party
 views
 .htaccess
 <> index.html
M_kampus.php x
1 <?php
2 class M_kampus extends CI_Model {
3
4 public function ambil_data_mahasiswa()
5 {
6 $query = $this->db->get('mahasiswa')->result_array();
7 return $query;
8 }
9 }
```

### 7.2.4. View

Kita akan fokus ke `$page=='mahasiswa'`. Saat link navigasi mahasiswa di click, maka if inilah yang akan di akses. Karena controller sudah mengirimkan data `$data['dt']`, kita cukup mengambil variabel `dt` saja (namun dituliskan `$dt` di view). Karena berbentuk array, kita dapat memanfaatkan perintah `foreach` untuk melakukan loop ke seluruh array dan menampilkan isinya.

views\isi.php

```

1 <?php
2 //----- HOME -----
3 if($page=='home')
4 {
5 ?>
6 <h2>Selamat Datang di Sistem Informasi Kampus</h2>
7
8 <?php
9 }
10 //----- MAHASISWA -----
11 else if($page=='mahasiswa')
12 {
13 ?>
14 <h2>DATA Mahasiswa</h2>
15 <table border="1">
16 <tr>
17 <th>Nim</th><th>Nama</th><th>Alamat</th>
18 </tr>
19 <?php foreach ($dt as $d) { ?>
20 <tr>
21 <td><?php echo $d['nim']; ?></td>
22 <td><?php echo $d['nama']; ?></td>
23 <td><?php echo $d['alamat']; ?></td>
24 </tr>
25 <?php } ?>
26 </table>
27
28 <?php
29 }
30 ?>

```

### 7.2.5. Jalankan di Browser

http://localhost/lsp/programmingCI/index.php/kampus

Click [Mahasiswa]

localhost/lsp/programmingCI/index.php/kampus/mahasiswa

## Sistem Informasi Kampus Poliban

[\[Home\]](#) [\[Mahasiswa\]](#)

### DATA Mahasiswa

| Nim  | Nama                  | Alamat                           |
|------|-----------------------|----------------------------------|
| A001 | Salsabila Anita Rahma | Jl. Veteran 24 Banjarmasin       |
| A005 | Naufal                | Jl. Perdagangan No14 Banjarmasin |
| A006 | Siti Nur Rahma        | Jl. Sultan Adam 29 Banjarmasin   |

Create By Agus SBN @ 2019

## BAB 8

### CRUD (LANJUTAN)

---

#### Capaian Pembelajaran:

1. Mampu memahami alur *Create, Update* dan *Delete*.
2. Mampu memahami peran *Controller, Model* dan *View* untuk CRUD
3. Mampu membuat coding untuk melakukan aksi CRUD

Sebelumnya sudah dibahas untuk menampilkan (*Read*) isi tabel menggunakan CI di bab sebelumnya. Berikut ini akan dibahas berturut-turut mulai dari *Create, Edit* dan *Delete*.

#### 8.1. Default Controller

Pada bab sebelumnya, kita harus mengetikkan nama *controller* `http://localhost/lsp/programmingCI/index.php/kampus` untuk menjalankan *controller* *kampus*. Agar otomatis langsung jalan saat kita ketikkan `http://localhost/lsp/programmingCI`. Hal yang harus kita lakukan adalah sebagai berikut.

##### 8.1.1. Config

config/router.php

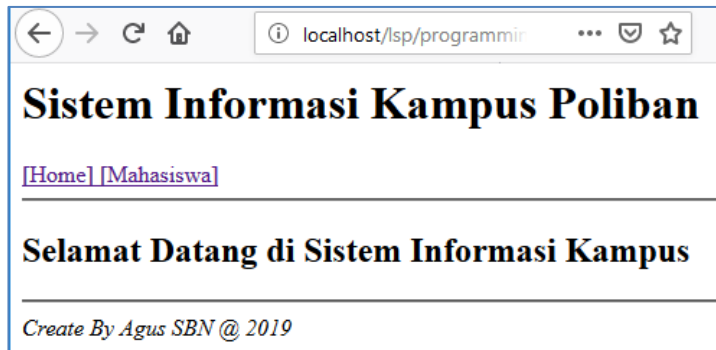
➤ Baris (52) :

```
$route['default_controller'] = 'kampus';
```

Jalankan di Browser. Cukup

```
http://localhost/lsp/programmingCI
```





## 8.2. *Create* (Tambah Data)

Alur berpikir untuk menambah data pada tabel atau *Create* dapat dilakukan dengan logika berpikir CI sebagai berikut.

- controllers*: persiapan validasi, cek *form*, jika *form* sudah valid maka panggil model *Tambah*,
- views*: buat *form*, terima masukan *user*, kirim balik ke *controller*,
- model*: simpan ke tabel saja (tidak melakukan pemeriksaan *form*, karena tugas pemeriksaan di *controller*).

### 8.2.1. Config

[config/autoload.php](#)

- Baris (61) :  
`$autoload['libraries']=array('database','form_validation');`
- Baris terakhir (135) :  
`$autoload['model'] = array('m_kampus');`

*Form\_validation* dari *library* adalah fitur untuk melakukan cek apakah sebuah *form* yang kita pilih sudah terisi atau tidak, jika belum di isi dan kita konfigurasi harus diisi, maka akan muncul *warning field cannot empty*.

### 8.2.2. Controller

[controllers/Kampus.php](#)

```

<?php
class Kampus extends CI_Controller {
//----- Halaman Depan -----
 public function index()
 {
 $data['page']="home";
 $this->tampil($data);
 }
//----- Halaman Mahasiswa -----
 public function mahasiswa()
 {
 $data['page']="mahasiswa";
 $data['dt']=$this->m_kampus->ambil_data_mahasiswa();
 $this->tampil($data);
 }
 public function mahasiswa_tambah()
 {
 $data['page']='mahasiswa_tambah';
 $this->form_validation->set_rules('nim', 'NIM', 'required');
 $this->form_validation->set_rules('nama', 'Nama Mhs', 'required');
 if ($this->Form_validation->run() == FALSE)
 $this->tampil($data);
 else
 {
 if($this->m_kampus->simpan_data_mahasiswa()){
 ?>
 <script>alert("Data Berhasil Ditambah");</script>
 <?php
 redirect('kampus/mahasiswa', 'refresh');
 }
 else
 die("Ada masalah penambahan Data");
 }
 }
//----- Koleksi Prosedur -----
 function tampil($data)
 {
 $this->load->view('header');
 $this->load->view('isi',$data);
 $this->load->view('footer');
 }
}

```

### 8.2.3. Model

models/M\_kampus.php

```

<?php
class M_kampus extends CI_model {
 public function ambil_data_mahasiswa()
 {
 $query = $this->db->get('mahasiswa')->result_array();
 return $query;
 }
 public function simpan_data_mahasiswa()
 {
 $data = array(
 'nim' => $this->input->post('nim'),
 'nama' => $this->input->post('nama'),
 'alamat' => $this->input->post('alamat')
);
 return $this->db->insert('mahasiswa', $data);
 }
}

```

### 8.2.4. Views

views/isi.php

```

<?php
//----- HOME -----
if($page=='home')
{
?>
<h2>Selamat Datang di Sistem Informasi Kampus</h2>

<?php
}
//----- MAHASISWA -----
else if($page=='mahasiswa')
{
?>
<h2>DATA Mahasiswa</h2>
 <button>Tambah</button>
<table border="1">
<tr>
<th>Nim</th><th>Nama</th><th>Alamat</th>
</tr>
<?php foreach ($dt as $d) { ?>
<tr>
<td><?php echo $d['nim']; ?></td>
<td><?php echo $d['nama']; ?></td>
<td><?php echo $d['alamat']; ?></td>
</tr>
<?php } ?>
</table>

<?php
}
else if($page=='mahasiswa_tambah')
{
?>
<h2>TAMBAH PESERTA</h2>
<?php echo validation_errors(); ?>
<?php echo form_open('kampus/mahasiswa_tambah'); ?>
Nim<input type="text" name="nim"/>

Nama<input type="text" name="nama"/>

Alamat<input type="text" name="alamat"/>

<input type="submit" value="Simpan" />
</form>
<?php
}
?>

```

### 8.2.5. Jalankan

1. Jalankan `http://localhost/lsp/programmingCI`  
Click Tambah
2. Tanpa di isi nim nama, klik kirim maka akan ada konfirmasi *field* harus di isi.
3. Isi, dan klik kirim.
4. Ada Konfirmasi.
5. Data seharusnya akan bertambah.

**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**DATA Mahasiswa**

| Nim  | Nama        | Alamat                           |
|------|-------------|----------------------------------|
| A001 | Anita Rahma | Jl. Veteran 24 Banjarmasin       |
| A005 | Naufal      | Jl. Perdagangan No14 Banjarmasin |

*Create By Agus SBN @ 2019*



**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**TAMBAH PESERTA**

The NIM field is required.

The Nama Mhs field is required.

Nim

Nama

Alamat

*Create By Agus SBN @ 2019*



**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**TAMBAH PESERTA**

The NIM field is required.

The Nama Mhs field is required.

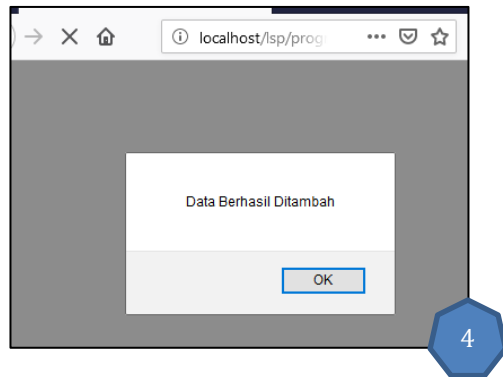
Nim

Nama

Alamat

*Create By Agus SBN @ 2019*





**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**DATA Mahasiswa**

Tambah

| Nim  | Nama        | Alamat                           |
|------|-------------|----------------------------------|
| A001 | Anita Rahma | Jl. Veteran 24 Banjarmasin       |
| A005 | Naufal      | Jl. Perdagangan No14 Banjarmasin |
| A006 | Tono        | Jl. Arjuna 22 Banjarbaru         |

*Create By Agus SBN @ 2019*

### 8.3. Update (Edit/Ubah data)

Alur berpikir untuk *EDIT* data pada tabel atau *UPDATE* dapat dilakukan dengan logika berpikir CI sebagai berikut:

- 1) Buat *Link Edit* dengan id *primary* tabel yang akan di edit
- 2) *Controllers* :
  - a. Panggil seluruh isi *record* yang terpilih menggunakan *model*.
  - b. Persiapan validasi, cek *form*, jika form sudah valid-> panggil model *Edit*.
- 3) *Views* : Buat *form*, terima masukan user, kirim balik ke *controller*
- 4) *Model* : *Update* ke tabel sesuai ID.

#### 8.3.1. Config

Sudah tidak ada yang perlu dikonfigurasi.

### 8.3.2. Controller

Karena sudah mahir, silahkan tambah *coding* ini di *controller*.  
Tempatkan sesuai kaidah pemrograman.

[controllers/Kampus.php](#)

```
public function mahasiswa_edit($id)
{
 $data['page']='mahasiswa_edit';
 $data['d']=$this->m_kampus->ambil_satu_data_mahasiswa($id);
 $this->form_validation->set_rules('nim', 'NIM', 'required');
 $this->form_validation->set_rules('nama', 'Nama Mhs', 'required');
 if ($this->form_validation->run() == FALSE)
 $this->tampil($data);
 else
 {
 if($this->m_kampus->update_data_mahasiswa()){
 ?>
 <script>alert("Data Berhasil Di Update");</script>
 <?php
 redirect('kampus/mahasiswa', 'refresh');
 }
 else
 die("Ada masalah Update Data");
 }
}
```

### 8.3.3. Model

Menambah coding di Model. Ada dua fungsi yang dibuat, yaitu sebagai berikut.

- 1) untuk memanggil satu record mahasiswa,
- 2) dan untuk UPDATE data ke tabel.
- 3)

[models/M\\_kampus.php](#)

```
public function ambil_satu_data_mahasiswa($id)
{
 $query = $this->db->get_where('mahasiswa', array('nim' => $id))->row_array();
 return $query;
}
public function update_data_mahasiswa()
{
 $nim = $this->input->post('nim');
 $data = array(
 'nama' => $this->input->post('nama'),
 'alamat' => $this->input->post('alamat')
);
 return $this->db->update('mahasiswa', $data, array('nim' => $nim));
}
```

### 8.3.4. Views

views/isi.php

```
1 <?php
2 //----- HOME -----
3 if($page=='home')
4 {
5 }
6 <h2>Selamat Datang di Sistem Informasi Kampus</h2>
7
8 <?php
9 }
10 //----- MAHASISWA -----
11 else if($page=='mahasiswa')
12 {
13 }
14 <h2>DATA Mahasiswa</h2>
15 >>button>Tambah</button>
16 <table border="1">
17 <tr>
18 <th>Nim</th><th>Nama</th><th>Alamat</th><th>Aksi</th>
19 </tr>
20 <?php foreach ($dt as $d) { ?>
21 <tr>
22 <td><?php echo $d['nim']; ?></td>
23 <td><?php echo $d['nama']; ?></td>
24 <td><?php echo $d['alamat']; ?></td>
25 <td>
26 Edit
27 Hapus
28 </td>
29 </tr>
30 <?php } ?>
31 </table>
32
33 <?php
34 }
35 else if($page=='mahasiswa_tambah')
36 {
37 }
38 <h2>TAMBAH PESERTA</h2>
39 <?php echo validation_errors(); ?>
40 <?php echo form_open('kampus/mahasiswa_tambah'); ?>
41 Nim <input type="text" name="nim"/>

42 Nama <input type="text" name="nama"/>

43 Alamat<input type="text" name="alamat"/>

44 <input type="submit" value="Simpan" />
45 </form>
46 <?php
47 }
48 else if($page=='mahasiswa_edit')
49 {
50 }
51 <h2>EDIT PESERTA</h2>
52 <?php echo validation_errors(); ?>
53 <?php echo form_open('kampus/mahasiswa_edit/'.$d['nim']); ?>
54 Nim <input type="text" name="nim" value=?php echo $d['nim'];?> readonly >

55 Nama <input type="text" name="nama" value=?php echo $d['nama'];?> >

56 Alamat<input type="text" name="alamat" value=?php echo $d['alamat'];?> >

57 <input type="submit" value="Simpan" />
58 </form>
59 <?php
60 }
61 ?>
```

### 8.3.5. Jalankan

1. Pilih data yang akan di *edit*, Klik *Edit*.
2. *Edit* datanya. *Click Simpan*.

3. Ada konfirmasi
4. Data sudah berubah.

**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**DATA Mahasiswa**

Tambah

| Nim  | Nama        | Alamat                           | Aksi                       |
|------|-------------|----------------------------------|----------------------------|
| A001 | Anita Rahma | Jl. Veteran 24 Banjarmasin       | <a href="#">Edit Hapus</a> |
| A005 | Naufal      | Jl. Perdagangan No14 Banjarmasin | <a href="#">Edit Hapus</a> |
| A006 | Tono        | Jl. Arjuna 22 Banjarbaru         | <a href="#">Edit Hapus</a> |

Create By Agus SBN @ 2019



**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**EDIT PESERTA**

Nim

Nama

Alamat

---

Create By Agus SBN @ 2019



localhost/isp/program

Data Berhasil Di Update



**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**DATA Mahasiswa**

Tambah

| Nim  | Nama                  | Alamat                           | Aksi                       |
|------|-----------------------|----------------------------------|----------------------------|
| A001 | Salsabila Anita Rahma | Jl. Veteran 24 Banjarmasin       | <a href="#">Edit Hapus</a> |
| A005 | Naufal                | Jl. Perdagangan No14 Banjarmasin | <a href="#">Edit Hapus</a> |
| A006 | Tono                  | Jl. Arjuna 22 Banjarbaru         | <a href="#">Edit Hapus</a> |

Create By Agus SBN @ 2019





#### 8.4. Delete (Hapus data)

Bagian terakhir dari CRUD adalah *delete*. Karena di *Edit* data, kita sudah sekalian membuat *link* untuk hapus data, maka kita sekarang tinggal menambah proses hapus saja.

##### 8.4.1. Config

Sudah tidak ada yang perlu dikonfigurasi.

##### 8.4.2. Controller

Karena sudah mahir, silahkan tambah *coding* ini di *controller*. Tempatkan sesuai kaidah pemrograman.

[controllers/Kampus.php](#)

```
public function mahasiswa_hapus($id)
{
 if($this->m_kampus->hapus_data_mahasiswa($id)){
 ?>
 <script>alert("Data Berhasil Di HAPUS");</script>
 <?php
 redirect('kampus/mahasiswa', 'refresh');
 }
 else
 die("Ada masalah HAPUS Data");
}
```

##### 8.4.3. Model

Menambah coding di Model untuk hapus satu record mahasiswa.

[models/M\\_kampus.php](#)

```
public function hapus_data_mahasiswa($id)
{
 return $this->db->delete('mahasiswa', array('nim' => $id));
}
```

##### 8.4.4. View

Views tidak ada modifikasi.

### 8.4.5. Jalankan

Buka data mahasiswa, Klik Hapus di salah satu data.

**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**DATA Mahasiswa**

Tambah

| Nim  | Nama                  | Alamat                           | Aksi                                       |
|------|-----------------------|----------------------------------|--------------------------------------------|
| A001 | Salsabila Anita Rahma | Jl. Veteran 24 Banjarmasin       | <a href="#">Edit</a> <a href="#">Hapus</a> |
| A005 | Naufal                | Jl. Perdagangan No14 Banjarmasin | <a href="#">Edit</a> <a href="#">Hapus</a> |
| A006 | Tono                  | Jl. Arjuna 22 Banjarbaru         | <a href="#">Edit</a> <a href="#">Hapus</a> |

Create By Agus SBN @ 2019



localhost/lsp/program

Data Berhasil Di HAPUS



**Sistem Informasi Kampus Poliban**

[\[Home\]](#) [\[Mahasiswa\]](#)

---

**DATA Mahasiswa**

Tambah

| Nim  | Nama                  | Alamat                           | Aksi                                       |
|------|-----------------------|----------------------------------|--------------------------------------------|
| A001 | Salsabila Anita Rahma | Jl. Veteran 24 Banjarmasin       | <a href="#">Edit</a> <a href="#">Hapus</a> |
| A005 | Naufal                | Jl. Perdagangan No14 Banjarmasin | <a href="#">Edit</a> <a href="#">Hapus</a> |

Create By Agus SBN @ 2019

### 8.5. Latihan

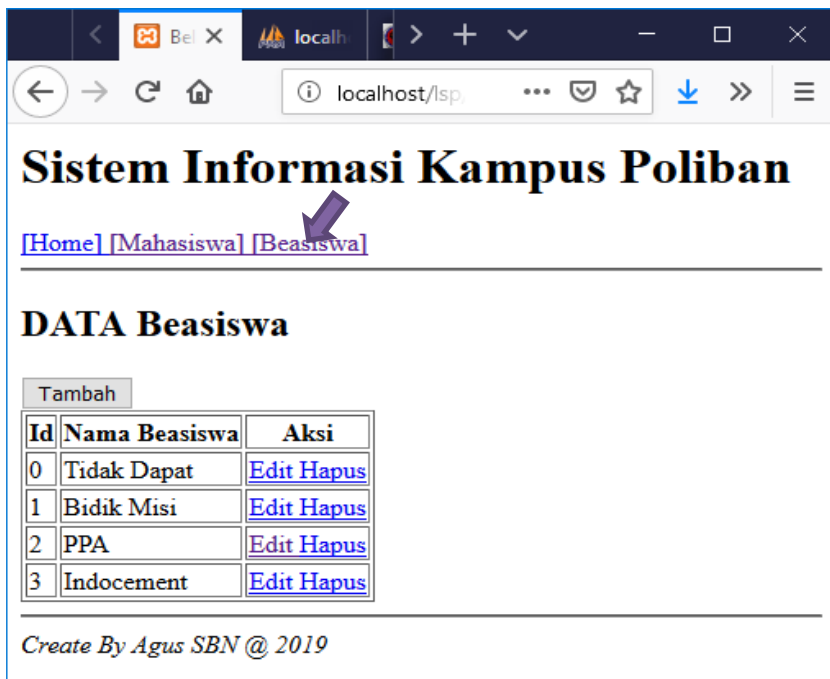
Latihan ini adalah sebuah *project* lanjutan. Buatlah sebuah tabel beasiswa sebagai berikut.

```
MariaDB [lsp_kampus01_db]> describe beasiswa;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| id_beasiswa | tinyint(4) | NO | PRI | NULL | auto_increment |
| nama_beasiswa | char(20) | NO | | NULL | |
+-----+-----+-----+-----+-----+-----+
```

```
MariaDB [lsp_kampus01_db]> select * from beasiswa;
+-----+-----+
| id_beasiswa | nama_beasiswa |
+-----+-----+
0	Tidak Dapat
1	Bidik Misi
2	PPA
3	Indocement
+-----+-----+
```

1. Buatlah Link di navigasi beasiswa
2. Buatlah CRUD untuk tabel beasiswa tersebut.

Inilah tampilan yang diharapkan dari latihan ini.



## BAB 9

### RELASI TABEL

---

#### Capaian Pembelajaran:

1. Mampu memahami alur menampilkan tabel yang berelasi.
2. Mampu memahami menggunakan perintah *join* di CI.
3. Mampu memahami konsep CRUD dengan tabel *join*.
4. Mampu memahami cara kerja *dropdown* pada CI.
5. Mampu membuat *coding* CRUD tabel berelasi.

Salah satu desain tabel yang baik adalah yang normal. Artinya akan ada relasi pada tabel yang tidak normal. Tabel yang berelasi perlu dilakukan proses *join* untuk menampilkan hasil yang sesuai harapan.

#### 9.1. Pembahasan Pembuatan CRUD untuk Tabel Beasiswa.

Melanjutkan tugas bab sebelumnya, akan dibahas sebagai referensi pengerjaan tabel beasiswa.

##### 9.1.1. Controller

[controllers/Kampus.php](#)

```
//----- Halaman BEASISWA -----
public function beasiswa()
{
 $data['page']="beasiswa";
 $data['dt']=$this->m_kampus->ambil_data_beasiswa();
 $this->tampil($data);
}
public function beasiswa_tambah()
{
 $data['page']='beasiswa_tambah';
 $this->form_validation->set_rules('nama_beasiswa', 'Nama Beasiswa ', 'required');
 if ($this->form_validation->run() == FALSE)
 $this->tampil($data);
 else
 {
 if($this->m_kampus->simpan_data_beasiswa()){
 ?>
 <script>alert("Data Berhasil Ditambah");</script>
 <?php
 redirect('kampus/beasiswa', 'refresh');
 }
 else
 die("Ada masalah penambahan Data");
 }
}
}
```

```
public function beasiswa_edit($id)
{
 $data['page']='beasiswa_edit';
 $data['d']=$this->m_kampus->ambil_satu_data_beasiswa($id);
 $this->form_validation->set_rules('nama_beasiswa', 'Nama Beasiswa', 'required');
 if ($this->form_validation->run() == FALSE)
 $this->tampil($data);
 else
 {
 if($this->m_kampus->update_data_beasiswa()){
 ?>
 <script>alert("Data Berhasil Di Update");</script>
 <?php
 redirect('kampus/beasiswa', 'refresh');
 }
 else
 die("Ada masalah Update Data");
 }
}
public function beasiswa_hapus($id)
{
 if($this->m_kampus->hapus_data_beasiswa($id)){
 ?>
 <script>alert("Data Berhasil Di HAPUS");</script>
 <?php
 redirect('kampus/beasiswa', 'refresh');
 }
 else
 die("Ada masalah HAPUS Data");
}
}
```

### 9.1.2. Berikut Model Beasiswa models/M kampus.php

```

//----- BEASISWA -----
public function ambil_data_beaasiswa()
{
 $query = $this->db->get('beasiswa')->result_array();
 return $query;
}
public function simpan_data_beaasiswa()
{
 $data = array(
 'nama_beaasiswa' => $this->input->post('nama_beaasiswa')
);
 return $this->db->insert('beasiswa', $data);
}
public function ambil_satu_data_beaasiswa($id)
{
 $query = $this->db->get_where('beasiswa', array('id_beaasiswa' => $id))->row_array();
 return $query;
}
public function update_data_beaasiswa()
{
 $id_beaasiswa = $this->input->post('id_beaasiswa');
 $data = array(
 'nama_beaasiswa' => $this->input->post('nama_beaasiswa')
);
 return $this->db->update('beasiswa', $data, array('id_beaasiswa' => $id_beaasiswa));
}
public function hapus_data_beaasiswa($id)
{
 return $this->db->delete('beasiswa', array('id_beaasiswa' => $id));
}
}

```

### 9.1.3. Modifikasi Views

views/header.php

```

<html lang="en">
<head>
 <title>Belajar CI</title>
</head>
<body>
<h1>Sistem Informasi Kampus Poliban</h1>
 <a href="<?php echo site_url()>/kampus">[Home]
 <a href="<?php echo site_url()>/kampus/mahasiswa">[Mahasiswa]
 <a href="<?php echo site_url()>/kampus/beasiswa">[Beasiswa]
<hr>

```

## views/isi.php

```
//----- BEASISWA -----
else if($page=='beasiswa')
{
?>
<h2>DATA Beasiswa</h2>
 <button>Tambah</button>
<table border="1">
 <tr>
 <th>Id</th><th>Nama Beasiswa</th><th>Aksi</th>
 </tr>
 <?php foreach ($dt as $d) { ?>
 <tr>
 <td><?php echo $d['id_beasiswa']; ?></td>
 <td><?php echo $d['nama_beasiswa']; ?></td>
 <td>
 >Edit
 >Hapus
 </td>
 </tr>
 <?php } ?>
</table>

<?php
}
else if($page=='beasiswa_tambah')
{
?>
<h2>TAMBAH BEASISWA</h2>
<?php echo validation_errors(); ?>
<?php echo form_open('kampus/beasiswa_tambah'); ?>
 nama_beasiswa <input type="text" name="nama_beasiswa"/>

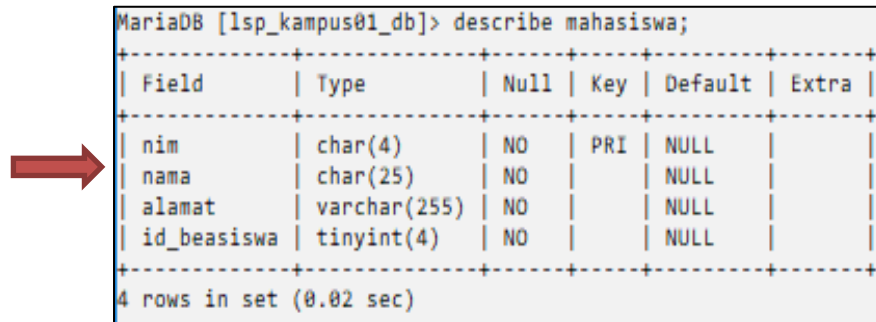
 <input type="submit" value="simpan" />
</form>
<?php
}
else if($page=='beasiswa_edit')
{
?>
<h2>EDIT BEASISWA</h2>
<?php echo validation_errors(); ?>
<?php echo form_open('kampus/beasiswa_edit/'.$d['id_beasiswa']); ?>
 id_beasiswa
 <input type="text" name="id_beasiswa"
 value="<?php echo $d['id_beasiswa'];?>" readonly >

 nama_beasiswa
 <input type="text" name="nama_beasiswa"
 value="<?php echo $d['nama_beasiswa'];?>" >

 <input type="submit" value="simpan" />
</form>
<?php
}
}
```

## 9.2. Merelasikan Tabel Mahasiswa dengan Beasiswa

Agar berelasi, kita akan menambahkan di tabel mahasiswa sebuah *field* `id_basiswa`. Sehingga struktur baru tabel mahasiswa nya menjadi seperti ini.



```
MariaDB [lsp_kampus01_db]> describe mahasiswa;
```

Field	Type	Null	Key	Default	Extra
nim	char(4)	NO	PRI	NULL	
nama	char(25)	NO		NULL	
alamat	varchar(255)	NO		NULL	
id_basiswa	tinyint(4)	NO		NULL	

4 rows in set (0.02 sec)

Contoh isi tabel mahasiswa beserta `id_basiswa` yang isinya harus ada di tabel beasiswa field `id_basiswa`.

```
MariaDB [lsp_kampus01_db]> select * from mahasiswa;
```

nim	nama	alamat	id_basiswa
A001	Salsabila Anita Rahma	Jl. Veteran 24 Banjarmasin	1
A005	Naufal	Jl. Perdagangan No14 Banjarmasin	2
A006	Yusuf	Jl. Sulatan Adam 22 Banjarmasin	0
A008	Wahyudin	Jl. Alalak Dalam 3 Banjarmasin	1

Gambar 42. Isi Tabel Mahasiswa yang Sudah Ada `id_basiswa`

Tabel awal mahasiswa diatas tampak bahwa yang ada adalah `id_basiswa`, di tabel tersebut tidak menampilkan nama data beasiswa. Untuk menampilkan tabel tersebut kita perlu untuk melakukan relasi dengan menggunakan perintah *join* agar `id_basiswa` akan muncul sebagai nama beasiswa di tampilan mahasiswa seperti dibawah ini.



<b>Sistem Informasi Kampus Poliban</b>				
<a href="#">[Home]</a> <a href="#">[Mahasiswa]</a> <a href="#">[Beasiswa]</a>				
<b>DATA Mahasiswa</b>				
<input type="button" value="Tambah"/>				
Nim	Nama	Alamat	Beasiswa	Aksi
A001	Salsabila Anita Rahma	Jl. Veteran 24 Banjarmasin	Bidik Misi	<a href="#">Edit Hapus</a>
A005	Naufal	Jl. Perdagangan No14 Banjarmasin	PPA	<a href="#">Edit Hapus</a>
A006	Yusuf	Jl. Sulatan Adam 22 Banjarmasin	Tidak Dapat	<a href="#">Edit Hapus</a>
A008	Wahyudin	Jl. Alalak Dalam 3 Banjarmasin	Bidik Misi	<a href="#">Edit Hapus</a>

Create By Agus SBN @ 2019

Gambar 43. Tampilan Data Mahasiswa dengan Join

Selain itu, saat kita menambah data beasiswa harus muncul sebagai *pull down*. Dan saat *edit*, *pull down* harus menampilkan pilihan yang sesuai dengan yang ada di tabel mahasiswa.

<b>Sistem Informasi Kampus Poliban</b>		<b>Sistem Informasi Kampus Poliban</b>	
<a href="#">[Home]</a> <a href="#">[Mahasiswa]</a> <a href="#">[Beasiswa]</a>		<a href="#">[Home]</a> <a href="#">[Mahasiswa]</a> <a href="#">[Beasiswa]</a>	
<b>TAMBAH MAHASISWA</b>		<b>EDIT MAHASISWA</b>	
Nim	<input type="text"/>	Nim	<input type="text" value="A008"/>
Nama	<input type="text"/>	Nama	<input type="text" value="Wahyudin"/>
Alamat	<input type="text"/>	Alamat	<input type="text" value="Jl. Alalak Dalam 3 Banjar"/>
Beasiswa	<input type="text" value="Tidak Dapat"/>	Beasiswa	<input type="text" value="Bidik Misi"/>
<input type="button" value="Simpan"/>	<input type="button" value="Tidak Dapat"/>	<input type="button" value="Simpan"/>	
Create By Agus SBN @ 2019		Create By Agus SBN @ 2019	

Gambar 44. Join pada Tambah/Edit adalah Dropdown

### 9.3. Coding CRUD Berelasi.

Untuk mengerjakan *coding* CRUD berelasi dengan perintah *join* dan *dropdown*, kita lanjutkan sebagai berikut.

### 9.3.1. Controller

controllers/Kampus.php

```
public function mahasiswa_tambah()
{
 $data['page']='mahasiswa_tambah';
 $data['pd_basiswa']=$this->m_kampus->pulldown_basiswa();
 $this->form_validation->set_rules('nim', 'NIM', 'required');
 $this->form_validation->set_rules('nama', 'Nama Mhs', 'required');
 if ($this->form_validation->run() == FALSE)
 $this->tampil($data);
 else
 {
 if($this->m_kampus->simpan_data_mahasiswa()){
 ?>
 <script>alert("Data Berhasil Ditambah");</script>
 <?php
 redirect('kampus/mahasiswa', 'refresh');
 }
 else
 die("Ada masalah penambahan Data");
 }
}
```

```
public function mahasiswa_edit($id)
{
 $data['page']='mahasiswa_edit';
 $data['pd_basiswa']=$this->m_kampus->pulldown_basiswa();
 $data['d']=$this->m_kampus->ambil_satu_data_mahasiswa($id);
 $this->form_validation->set_rules('nim', 'NIM', 'required');
 $this->form_validation->set_rules('nama', 'Nama Mhs', 'required');
 if ($this->form_validation->run() == FALSE)
 $this->tampil($data);
 else
 {
 if($this->m_kampus->update_data_mahasiswa()){
 ?>
 <script>alert("Data Berhasil Di Update");</script>
 <?php
 redirect('kampus/mahasiswa', 'refresh');
 }
 else
 die("Ada masalah Update Data");
 }
}
```

### 9.3.2. Model

controllers/Kampus.php

```
<?php
class M_kampus extends CI_model {
//----- MAHASISWA -----
 public function ambil_data_mahasiswa()
 {
 $this->db->select('*');
 $this->db->from('mahasiswa');
 $this->db->join('beasiswa', 'beasiswa.id_beasiswa=mahasiswa.id_beasiswa', 'left');
 $query = $this->db->get()->result_array();
 return $query;
 }
}
```

```
 public function pulldown_beasiswa()
 {
 $query = $this->db->get('beasiswa');
 foreach ($query->result() as $r)
 $q[$r->id_beasiswa] = $r->nama_beasiswa;
 return $q;
 }
}
```

### 9.3.3. View

controllers/Kampus.php

```
else if($page=='mahasiswa_tambah')
{
?>
 <h2>TAMBAH MAHASISWA</h2>
 <?php echo validation_errors(); ?>
 <?php echo form_open('kampus/mahasiswa_tambah'); ?>
 Nim <input type="text" name="nim"/>

 Nama <input type="text" name="nama"/>

 Alamat<input type="text" name="alamat"/>

 Beasiswa <?php echo form_dropdown('id_beasiswa',$pd_beasiswa);?>

 <input type="submit" value="Simpan" />
 </form>
<?php
}
```

```

else if($page=='mahasiswa_edit')
{
?>
<h2>EDIT MAHASISWA</h2>
<?php echo validation_errors(); ?>
<?php echo form_open('kampus/mahasiswa_edit/'.$sd['nim']); ?>
 Nim <input type="text" name="nim" value="<?php echo $d['nim'];?>" readonly >

 Nama <input type="text" name="nama" value="<?php echo $d['nama'];?>" >

 Alamat<input type="text" name="alamat" value="<?php echo $d['alamat'];?>" >

 Beasiswa <?php echo form_dropdown('id_basiswa',$pd_basiswa,$d['id_basiswa']);?>

 <input type="submit" value="Simpan" />
</form>
<?php
}

```

### 9.3.4. Jalankan

Mestinya sudah muncul nama beasiswa di halaman mahasiswa. Dan saat di tambah dan di *edit*, akan muncul *dropdown* yang merupakan isi dari tabel beasiswa, seperti yang telah ditunjukkan pada Gambar 43 dan 44 diatas.



## BAB 10

### CETAK DAN SEARCH

---

#### Capaian Pembelajaran:

1. Mampu memahami konsep mencetak ke *printer*.
2. Mampu memahami konsep pencarian.
3. Mampu membuat aplikasi cetak dan cari.
4. Membuat *coding* cetak dan pencarian.

Cetak pada prinsipnya sama dengan menampilkan ke layar, hanya saja perlu menambah sebuah perintah *javascript* untuk mengarahkan perintah cetak ke layar menjadi cetak ke *printer*.

#### 10.1. Cetak

Penjelasan berikutnya akan menjelaskan tentang cara mencetak tabel mahasiswa ke *printer*. Sehingga kita akan fokus ke tabel mahasiswa dan *coding* yang berkenaan dengan `$page=='mahasiswa'` saja.

##### 10.1.1. Controller

[controllers/Kampus.php](#)

```
public function mahasiswa_cetak()
{
 $data['page']="mahasiswa";
 $data['dt']=$this->m_kampus->ambil_data_mahasiswa();
 $this->load->view('c_mahasiswa',$data);
}
```

## 10.1.2. View

### Views/isi.php

```
//----- MAHASISWA -----
else if($page=='mahasiswa')
{
?>
<h2>DATA Mahasiswa</h2>
 <button>Tambah</button>
 <button>Cetak</button>
<table border="1">
 <tr>
 <th>Nim</th><th>Nama</th><th>Alamat</th><th>Bebasiswa</th><th>Aksi</th>
 </tr>
 <?php foreach ($dt as $d) { ?>
 <tr>
 <td><?php echo $d['nim']; ?></td>
 <td><?php echo $d['nama']; ?></td>
 <td><?php echo $d['alamat']; ?></td>
 <td><?php echo $d['nama_bebasiswa']; ?></td>
 <td>
 >Edit
 >Hapus
 </td>
 </tr>
 <?php } ?>
</table>

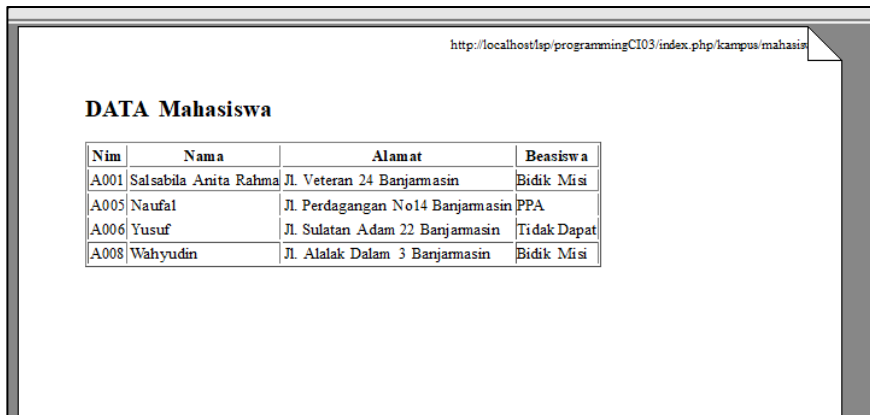
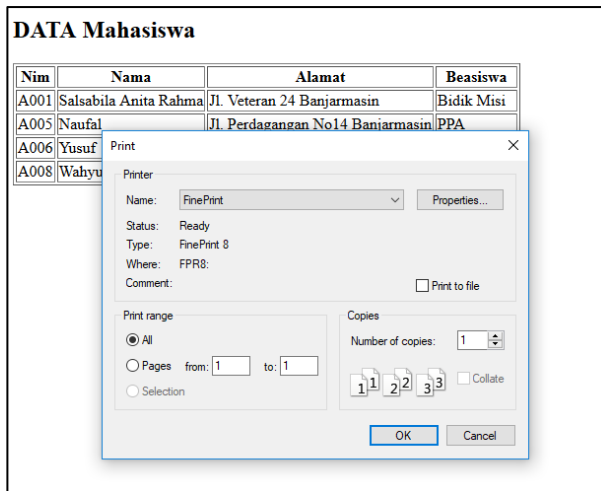
<?php
}
```

Buat **File BARU** di *VIEW*

### Views/c mahasiswa.php

```
<HTML>
<head>
</head>
<BODY onLoad="javascript:window.print()">
 <h2>DATA Mahasiswa</h2>
 <table border="1">
 <tr>
 <th>Nim</th><th>Nama</th><th>Alamat</th><th>Bebasiswa</th>
 </tr>
 <?php foreach ($dt as $d) { ?>
 <tr>
 <td><?php echo $d['nim']; ?></td>
 <td><?php echo $d['nama']; ?></td>
 <td><?php echo $d['alamat']; ?></td>
 <td><?php echo $d['nama_bebasiswa']; ?></td>
 </tr>
 <?php } ?>
 </table>
</body>
</HTML>
```

## Hasilnya



### 10.2. Search

Pencarian adalah proses menampilkan tabel sepertihalnya proses *Read*. Hanya saja ditambahkan perintah *where* pada bagian sql di *model*. Untuk itu diperlukan sebuah *form* dan *text input* untuk memasukkan kata yang akan dicari.

Pada program ini akan ditunjukkan pencarian data berdasarkan nama yang sesuai dengan yang dicari. Agar pencarian lebih luas,



digunakan perintah *like* dan parameter % *standart* ci, sehingga semua nama dengan awalan dan akhiran yang ada dipencarian akan ditampilkan. Untuk pencarian yang lebih spesifik, silahkan dipelajari fungsi-fungsi *where* dan *like* yang lain. Sama seperti cetak, pencarian akan dicontohkan pada tabel mahasiswa.

### 10.2.1. Modifikasi *Controller*

controllers/Kampus.php

```
//----- Halaman Mahasiswa -----
public function mahasiswa()
{
 $data['page']="mahasiswa";
 $data['cari']=$this->input->post('nama');
 $data['dt']=$this->m_kampus->ambil_data_mahasiswa($data['cari']);
 $this->tampil($data);
}
```

### 10.2.2. Modifikasi *Model*

Memodifikasi *coding* di *Model* untuk Memanggil tabel mahasiswa dengan *like* dan *where*.

models/M\_kampus.php

```
public function ambil_data_mahasiswa($cari)
{
 $this->db->select('*');
 $this->db->from('mahasiswa');
 $this->db->join('beasiswa','beasiswa.id_beasiswa=mahasiswa.id_beasiswa','left');
 $this->db->like('nama',$cari);
 $query = $this->db->get_where()->result_array();
 return $query;
}
```

### 10.2.3. Modifikasi *Views*

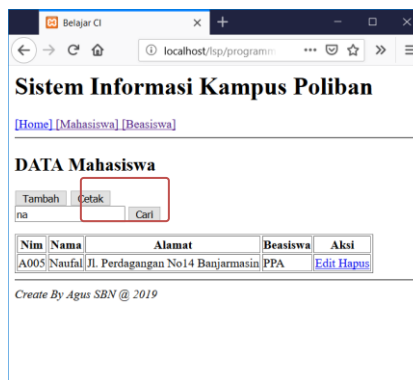
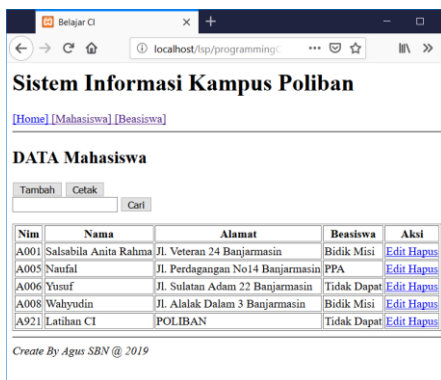
views/isi.php

```

//----- MAHASISWA -----
else if($page=='mahasiswa')
{
?>
<h2>DATA Mahasiswa</h2>
 <button>Tambah</button>
 <button>Cetak</button>
<?php
echo form_open('kampus/mahasiswa');
?>
<input type="text" name="nama" value="<?php echo $cari;?>">
<input type="submit" value="Cari">
</form>
<table border="1">
<tr>
<th>Nim</th><th>Nama</th><th>Alamat</th><th>Beasiswa</th><th>Aksi</th>
</tr>
<?php foreach ($dt as $d) { ?>
<tr>
<td><?php echo $d['nim']; ?></td>
<td><?php echo $d['nama']; ?></td>
<td><?php echo $d['alamat']; ?></td>
<td><?php echo $d['nama_beasiswa']; ?></td>
<td>

```

## 10.2.4. Hasil



## *Daftar Pustaka*

- Abdul Kadir, 2002, *Dasar Pemrograman WEB dinamis menggunakan PHP*, Andi Yogyakarta
- Anhar, 2018, *Trik SEO Security Codeigniter*, Lokomedia Yogyakarta
- Anton Subagia. 2018, *Kolaborasi Codeigniter dan Ajax dalam Perancangan CMS*, Elexmedia Komputindo Jakarta
- Awan Pribadi Basuki, Lukmanul Hakim, 2010, *Membangun Web Berbasis PHP dengan Codeigniter*. Lokomedia Yogyakarta
- Gugun Septian. 2011, *Trik Pintar Menguasai Codeigniter*. Elexmedia Komputindo Jakarta
- Komang Wismakarma, 2010, *9 Langkah menjadi Master Framework Codigniter*, Elexmedia Komputindo Jakarta

# PEMROGRAMAN WEB LANJUT

( ARRAY, FUNGSI DAN CRUD DENGAN CODEIGNITER )

## AGUS SETIYO BUDI N

Alhamdulillah, akhirnya buku pemrograman *web* lanjut yang membahas Framework Codeigniter (CI) pada mata kuliah pemrograman *web* pada kampus POLIBAN (Politeknik Negeri Banjarmasin) Prodi TI dapat diselesaikan.

Sebelum membahas CI, buku ini memulai pembahasan tentang *array, fungsi, prosedur* dan *file*. Hal ini dikarenakan, teramat banyak *array* dan *fungsi* yang dipakai dalam CI, sehingga pemahaman tentang hal tersebut mutlak diperlukan.

Metode buku ini adalah, membaca, memahami dan mencoba. Jika tidak mencoba, kecil kemungkinan akan paham dan mengerti, mengingat banyak faktor X yang muncul saat kita melakukan coding. Akhir kata, semoga buku ini bermanfaat kepada para pembaca semuanya dalam memahami cara kerja Framework Codeigniter untuk membangun sistem informasi sederhana CRUD untuk dua tabel yang berelasi.



Penerbit Poliban Press

Redaksi :

Politeknik Negeri Banjarmasin, Jl. Brigjen H. Hasan Basry,  
Pangeran, Komp. Kampus ULM, Banjarmasin Utara

Telp : (0511)3305052

Email : [press@poliban.ac.id](mailto:press@poliban.ac.id)

ISBN 978-623-91786-5-9

