

**LAPORAN**  
**BENGKEL PEMOGRAMAN FRAMEWORK**  
**“P2 :Konfigurasi Dasar Codeigniter, Konsep MVC,**  
**Pemanggilan library, Pengenalan Function”**



**NAMA : Ikhsan Damasraya**  
**NIM : 2255301076**  
**KELAS : 2 TI D**  
**DOSEN : Mutia Sari Zulvi, S.S.T., M.M.S.I**  
**ILB : Ika Suhasmi**

**PROGRAM STUDI TEKNIK INFORMATIKA**  
**POLITEKNIK CALTEX RIAU**  
**TA 2023 / 2024**

## 1. Pengenalan Dasar

### a. Konfigurasi Dasar

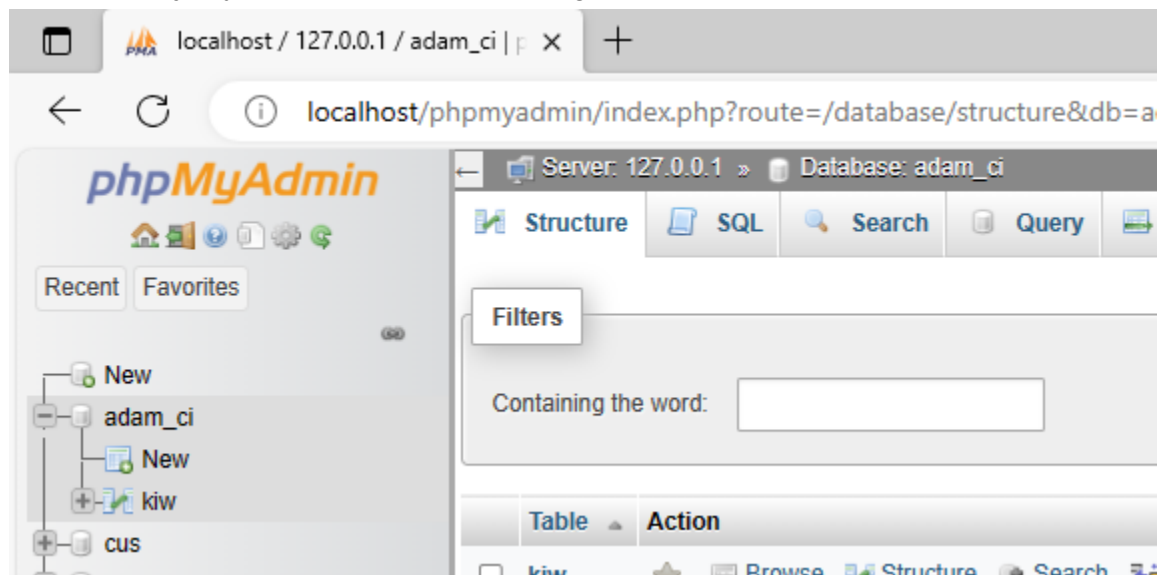
Membuka file **autoload.php** yang berada pada folder **application/config**  
Lalu menambahkan codingan dibawah ini pada masing-masing **\$autoload**

```
| $autoload[ 'libraries' ] = array( 'user_agent' => 'ua' );  
*  
$autoload[ 'libraries' ] = array( 'database', 'session', 'email', 'table' );  
*  
*/  
$autoload[ 'helper' ] = array( 'url', 'file' );
```

Buka file **config.php** yang ada di folder **application/config**, lalu  
tambahkan codingan dari folder yang dimiliki, berguna untuk menggunakan  
domain dengan lebih mudah

```
|  
*/  
$config[ 'base_url' ] = 'http://localhost/IkhsanDamasraya_2TID_CI/';  
*  
*/
```

Selanjutnya buat database baru dengan format **nama\_ci**



Setelah itu buka file **database.php** pada folder **application/config** dan Tambahkan username dan database sesuai dengan database yang dibuat diatas tadi.

```
72  */
73  $active_group = 'default';
74  $query_builder = TRUE;
75
76  $db['default'] = array(
77      'dsn' => '',
78      'hostname' => 'localhost',
79      'username' => 'root',
80      'password' => '',
81      'database' => 'adam_ci',
82      'dbdriver' => 'mysqli',
83      'dbprefix' => '',
84      'pconnect' => FALSE,
85      'db_debug' => (ENVIRONMENT !== 'production'),
86      'cache_on' => FALSE,
87      'cachedir' => '',
88      'char_set' => 'utf8',
89      'dbcollat' => 'utf8_general_ci',
90      'swap_pre' => '',
91      'encrypt' => FALSE,
92      'compress' => FALSE,
93      'stricton' => FALSE,
94      'failover' => array(),
95      'save_queries' => TRUE
96  );
97
```

File **routes.php** ini berada pada folder **application/config**, route-route ini mempunyai peran masing-masing yaitu

<code>\$route['default_controller'] = 'welcome';</code>	Mengatur controller mana yang akan digunakan sebagai controller default pada website
<code>\$route['404_override'] = '';</code>	Berguna untuk menampilkan halaman yang tidak ditemukan
<code>\$route['translate_uri_dashes'] = FALSE;</code>	Otomatis mereplace hal yang false

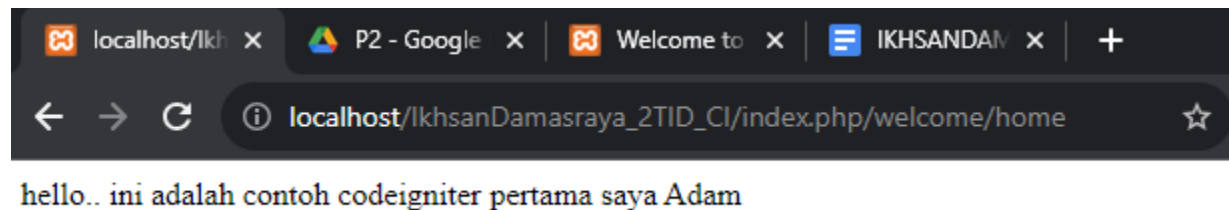
```
application > config > routes.php
48 |
49 | Examples: my-controller/index -> my_controller/index
50 |           my-controller/my-method -> my_controller/my_method
51 */
52 $route['default_controller'] = 'welcome';
53 $route['404_override'] = '';
54 $route['translate_uri_dashes'] = FALSE;
55
```

## 2. Struktur URL

- localhost/ : Sebuah induk/host dari website/web server yang dibuat
- adam\_ci : nama folder projek di web server atau htdocs
- Index.php : file yang merupakan wajib untuk menjalankan CI namun file ini bisa ditiadakan menggunakan .htaccess
- Welcome : Nama class pada gambar dibawah
- Index : Nama function pada gambar dibawah
- Membuka file **welcome.php** yang berada pada **application/controller**
- Controller berguna sebagai penengah/perantara antara Model, View, Core dan berbagai resources yang dibutuhkan untuk membuat sebuah web

```
0 references | 0 overrides
21 public function index()
22 {
23     $this->load->view('welcome_message');
24 }
0 references | 0 overrides
25 public function home ($name = "")
26 {
27     echo "hello.. ini adalah contoh codeigniter pertama saya Adam";
28 }
0 references | 0 overrides
29 public function komentar()
30 {
31     echo "ini adalah function komentar";
32 }
33 }
34
```

Output:



### **Tugas Dadakan**

- **home** : Ini adalah salah satu fungsi yang ada dalam controller. Biasanya, fungsi ini akan merespons permintaan ketika URL tanpa segmen tambahan ([http://localhost/nama\\_proyek/controller](http://localhost/nama_proyek/controller)) diakses.
- **komentar()**: Ini adalah fungsi lain dalam controller. Fungsi ini mungkin merespons permintaan yang terkait dengan menampilkan komentar atau melakukan operasi terkait komentar dalam aplikasi.
- **\$nama=""**; Ini adalah deklarasi variabel \$nama dengan nilai awal string kosong. Variabel ini mungkin digunakan dalam fungsi-fungsi tersebut untuk menyimpan atau memanipulasi data.
- **Jika \$nama=""** dihilangkan, apakah terjadi error? Jika deklarasi variabel \$nama dihilangkan dari kode, maka jika fungsi-fungsi tersebut mencoba mengakses atau memanipulasi variabel \$nama, akan terjadi kesalahan (error) karena variabel tersebut tidak ada. Ini akan tergantung pada bagaimana variabel tersebut digunakan dalam fungsi-fungsi tersebut.

### 3. Konsep MVC

MVC adalah pola arsitektur yang membagi aplikasi menjadi tiga komponen utama:

1. Model: Menangani logika bisnis dan interaksi dengan database.
2. Controller: Mengkoordinasikan aktivitas antara Model dan View.
3. View: Bertugas untuk merepresentasikan data kepada pengguna.

Keuntungan penggunaan pola MVC meliputi komponen yang berfungsi secara independen, fleksibilitas dalam mengubah komponen individual, dan meningkatkan produktivitas tim pengembangan.

#### Contoh penerapan Controller dan View

- Dalam CodeIgniter, controller adalah komponen yang mengatur logika aplikasi dan merespons permintaan dari pengguna.
- Contoh kode controller "Blog.php" yang merespons permintaan dengan memuat tampilan (View).
- Kode dalam tampilan "hello\_codeigniter.php" digunakan sebagai contoh penerapan View. View bertugas untuk merepresentasikan data yang diberikan oleh Controller kepada pengguna.

application/controller/Blog.php

```
1  <?php
2  defined('BASEPATH') or exit ('No direct script acces allowed');
   0 references | 0 implementations
3  class Blog extends CI_Controller
4  {
   1 reference | 0 overrides | prototype
5      public function __construct()
6      {
7          parent::__construct();
8      }
   0 references | 0 overrides
9      function index()
10     {
11         $this->view("hello_codeigniter")
12     }
13 }
14 ?>
15
```

application/views/hello\_codeigniter.php

```
application > views > hello_codeigniter.php
1  <h1>Hello saya adalah halaman view</h1>
2
```

Output:



Fungsi view memiliki 3 Parameter

**A. Nama file view**

Nama file view yang hendak di-load terletak didalam folder **application/view**

**B. Data Parameter**

Data parameter yaitu parameter yang digunakan untuk melewati data dari controller kedalam view contoh nya:

application/controllers/Blog2.php

```
application > controllers > Blog2.php > Blog2 > index()
1  <?php
2  defined('BASEPATH') or exit ('No direct script access allowed');
3  class Blog2 extends CI_Controller
4  {
5      public function __construct()
6      {
7          parent::__construct();
8      }
9      function index()
10     {
11         $data['judul'] = "Judul blog";
12         $data['isi'] = "Isi blog";
13         $this->load->view("blog_view, $data");
14     }
15 }
16
```



## application/views/blog\_view.php

```
hello_codeigniter.php  Blog2.php  blog_view.php x
application > views > blog_view.php
1  <h1><?php echo $judul; ?> </h1>
2  <p><?php echo $isi; ?></p>
3  <p><br> />Page rendered in {elapsed_time} seconds
4  </p>
5
```

## Output:

```
01. Materi Konfigurasi  localhost/lkhsanDamasraya_2TID
localhost/lkhsanDamasraya_2TID_CI/index.php/Blog2

Judul blog

Isi blog

/>Page rendered in 0.0651 seconds
```

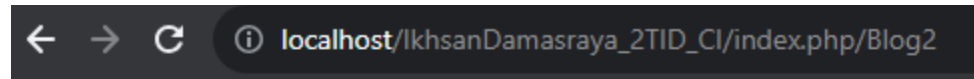
## C. Output Parameter

Parameter ini akan set true jika kita ingin menyimpan hasil view ke dalam sebuah variabel. Contohnya:

## application/controllers/Blog2.php

```
application > controllers > Blog2.php > Blog2 > index()
1  <?php
2  defined('BASEPATH') or exit ('No direct script access allowed');
3  class Blog2 extends CI_Controller
4  {
5      public function __construct()
6      {
7          parent::__construct();
8      }
9      function index()
10     {
11         $data['judul'] = "Judul blog";
12         $data['isi'] = "Isi blog";
13         $out = $this->load->view("blog_view", $data , true);
14         echo $out;
15     }
16 }
17
```

Output:



# Judul blog

Isi blog

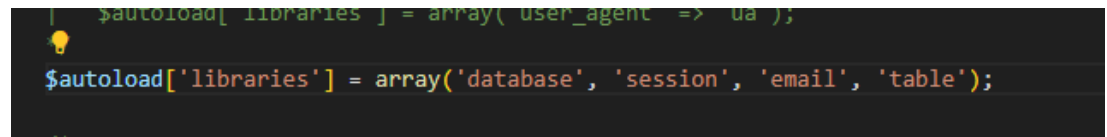
/>Page rendered in {elapsed\_time} seconds

## Library dalam CodeIgniter

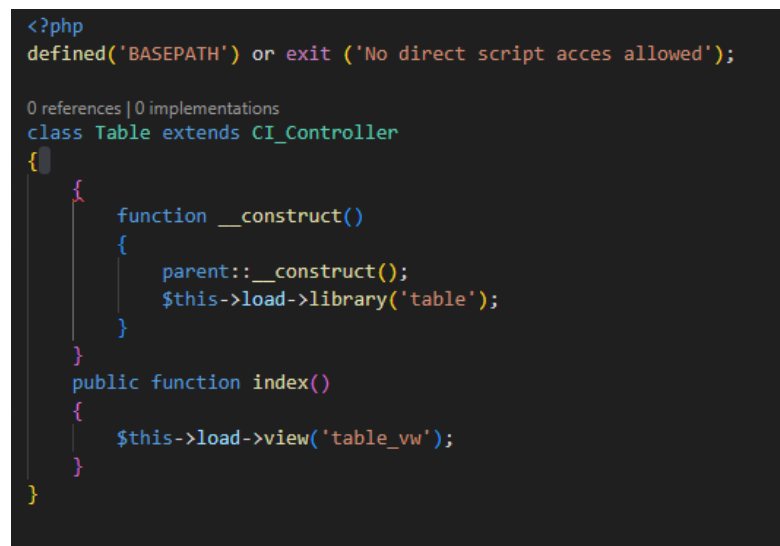
- CodeIgniter menyediakan banyak library yang dapat digunakan untuk tugas-tugas umum dalam pengembangan aplikasi web.
- Contoh beberapa library yang tersedia meliputi database, form validation, email, session, dan lainnya.
- Kita dapat mengimpor dan menggunakan library ini dalam aplikasi Anda untuk mempercepat pengembangan.

## Ada dua cara untuk menambahkan library pada codeigniter

1. Menambahkan file autoupload.php pada bagian libraries



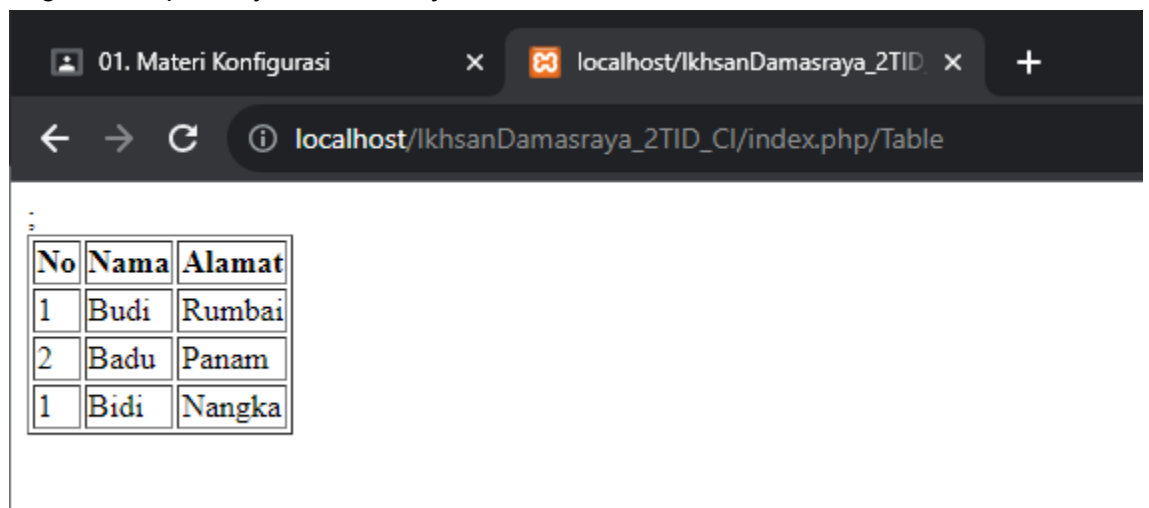
2. Menambahkan pada bagian function **\_\_construct** yang berada pada file controller, contoh:



Lalu, langkah berikutnya yaitu membuat halaman view dengan nama `application/views/table_vw.php`

```
application > views > table_vw.php
1  <?php
2
3  $template = array([
4      'table_open' => '<table border="1 cellpadding"4" cellspacing"0">;',
5      'table_close' => '</table>'
6  ]);
7
8  $this->table->set_template($template);
9  $this->table->set_heading(array('No', 'Nama', 'Alamat'));
10
11 $this->table->add_row(array('1', 'Budi', 'Rumbai'));
12 $this->table->add_row(array('2', 'Badu', 'Panam'));
13 $this->table->add_row(array('1', 'Bidi', 'Nangka'));
14
15 echo $this->table->generate();
16
17 ?>
18
```

Beginilah tampilan nya sesudah dijalankan.



## Membuat Library Sendiri

- CodeIgniter memungkinkan untuk membuat library khusus sesuai kebutuhan aplikasi.
- Dapat menyimpan library buatan sendiri dalam folder **"application/libraries"** dan mengaksesnya dari Controller.

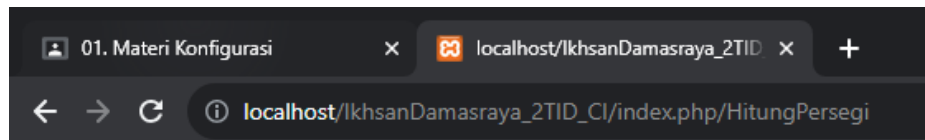
## Membuat file **persegi.php** dan simpan didalam folder **application/libraries**

```
application > libraries > persegi.php > ...
1  <?php
2
3  class persegi
4  {
5      0 references | 0 overrides
6      function keliling($sisi)
7      {
8          $keliling = $sisi * 4;
9          echo "Keliling Persegi dengan sisi " . $sisi . " adalah " . $keliling;
10     }
11     0 references | 0 overrides
12     function luas($sisi)
13     {
14         $luas = $sisi * $sisi;
15         echo "Luas Persegi dengan sisi " . $sisi . " adalah " . $luas;
16     }
17 }
18
```

## Berikut nya buat file **Hitungpersegi.php** yang disimpan didalam **application/controlles**

```
application > controllers > HitungPersegi.php > HitungPersegi
1  <?php
2  defined('BASEPATH') or exit('No direct script acces allowed');
3
4  class HitungPersegi extends CI_Controller
5  {
6      4 references | 0 overrides | prototype
7      function __construct()
8      {
9          parent::__construct();
10         $this->load->library("persegi");
11     }
12     0 references | 0 overrides
13     function index()
14     {
15         $this->persegi->keliling('10');
16         echo "<br/>";
17         $this->persegi->luas('10');
18     }
19 }
20
```

## Output:



Keliling Persegi dengan sisi 10 adalah 40  
Luas Persegi dengan sisi 10 adalah 100

#### 4. Tugas

##### a. Soal

Silahkan buat library sendiri untuk perhitungan berikut.

Volume Tabung (V)	$V = \pi \times r^2 \times t$
Luas Permukaan Tabung (L)	$L = 2 \times \pi \times r \times (r + t)$
Luas Selimut Ls	$L_s = 2 \times \pi \times r \times t$

```
← → ↻ localhost/ardi_ci/index.php/hitung3
Phi : 3.14
Jari-jari 5
Tinggi 3
Volume Tabung adalah :235.5

Luas Permukaan Tabung adalah :471

Luas Selimut Tabung adalah :94.2
```

##### application/libaries/tabung.php

Membuat library menggunakan rumus menghitung volume tabung, luas permukaan tabung dan luas selimut tabung.

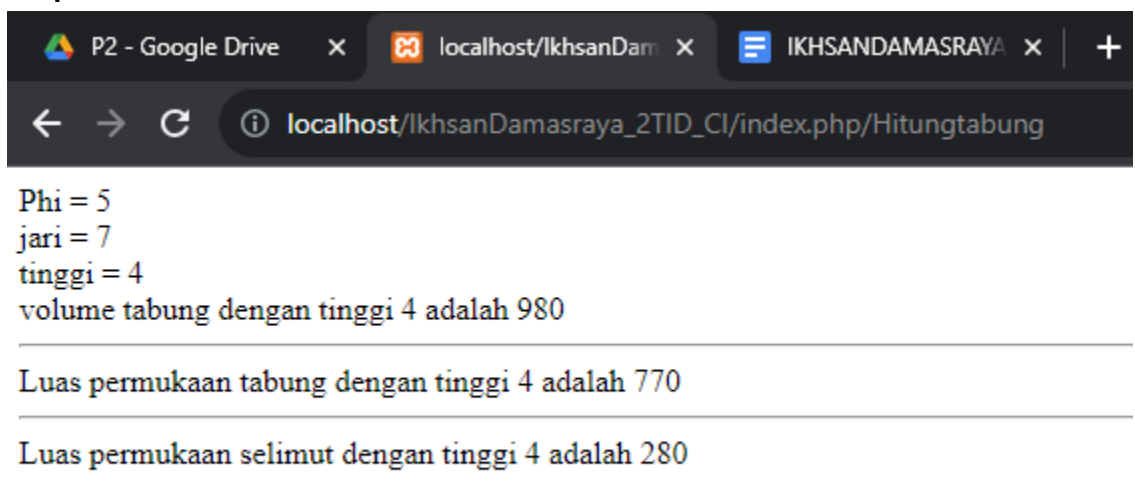
```
application > libraries > tabung.php > Tabung > volume()
1  <?php
   0 references | 0 implementations
2  class Tabung
3  {
   0 references | 0 overrides
4  function volume($pi, $r, $t)
5  {
6      echo "Phi = " . $pi . "<br>";
7      echo "jari = " . $r . "<br>";
8      echo "tinggi = " . $t . "<br>";
9      $volume = $pi * $r * $r * $t;
10     echo "volume tabung dengan tinggi " . $t . " adalah " . $volume;
11 }
12 }
   0 references | 0 overrides
13 function luas($pi, $r, $t)
14 {
15     $luas = 2 * $pi * $r * ($r + $t);
16     echo "Luas permukaan tabung dengan tinggi " . $t . " adalah " . $luas;
17 }
   0 references | 0 overrides
18 function luasS($pi, $r, $t)
19 {
20     $luassel = 2 * $pi * $r * $t;
21     echo "Luas permukaan selimut dengan tinggi " . $t . " adalah " . $luassel;
22 }
23 }
24 ?>
25
```

## application/controllers/Hitungtabung.php

Mencetak output menggunakan php ini.

```
application > controllers > Hitungtabung.php > Hitungtabung > __construct()
1  <?php
2  defined('BASEPATH') or exit('No direct script access allowed');
3
4  class Hitungtabung extends CI_Controller
5  {
6      function __construct()
7      {
8          parent::__construct();
9          $this->load->library('tabung');
10     }
11
12     function index()
13     {
14         $this->tabung->volume('5', '7', '4');
15         echo "<hr/>";
16         $this->tabung->luas('5', '7', '4');
17         echo "<hr/>";
18         $this->tabung->luass('5', '7', '4');
19         echo "<hr/>";
20     }
21 }
22
```

Output:



Phi = 5  
jari = 7  
tinggi = 4  
volume tabung dengan tinggi 4 adalah 980

---

Luas permukaan tabung dengan tinggi 4 adalah 770

---

Luas permukaan selimut dengan tinggi 4 adalah 280

---