

**Acara ke** : 7  
**Pokok Bahasan** : Model dan Parsing Data  
**Acara Praktikum** : Model dan Parsing Data  
**Tempat** : Laboratorium Rekayasa Perangkat Lunak  
**Alokasi Waktu** : 1 X 120 menit

### Kompetensi Dasar

1. Mahasiswa mampu memahami konsep Model pada CI.
2. Mahasiswa mampu menerapkan parsing data pada CI.

### Bahan

No.	Nama Bahan/ Alat	Kebutuhan	Keterangan
1.	Kertas A4 70 gram	5	Rim

### Dasar Teori

Model mewakili struktur data dari website yang bisa berupa basis data maupun data lain, misalnya dalam bentuk file teks atau file xml. Biasanya didalam model akan berisi class dan fungsi untuk mengambil, melakukan update dan menghapus data website. Karena sebuah website biasanya menggunakan basis data dalam menyimpan data maka bagian Model biasanya akan berhubungan dengan perintah-perintah query SQL. Model bisa dibilang khusus digunakan untuk melakukan koneksi ke basis data oleh karena itu logika-logika pemrograman yang berada didalam model juga harus yang berhubungan dengan basis data. Misalnya saja pemilihan kondisi tetapi untuk memilih melakukan query yang mana. Bentuk umum Model adalah sebagai berikut:

```
class Persegipanjang_model extends CI_model{  
    //badan class  
    // Konstruktor kelas model  
    Function __construct();  
    parent::__construct();  
}
```

*Note : nama model harus sama dengan nama class- nya*

Contoh :

```

1  <?php
2  class temp_model extends Model {
3      function temp_model()
4      {
5          parent::Model();
6      }
7
8      function getProducts()
9      {
10         return $this->db->query("SELECT * FROM AA_PRODUCTS");
11     }
12 }
13 ?>

```

## Kegiatan Praktikum

### Parsing data dari controller ke views

Buatlah file Controller dengan nama Mahasiswa.php dan file View dengan nama view\_mahasiswa.php

```

<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Mahasiswa extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->model('Mahasiswa_model');
    }
    public function index(){
        $data['nama'] = "Kim Jefry";
        $data['prodi'] = "MIF";
        $this->load->view('view_mahasiswa',$data);
    }
    public function profil(){
        echo "ini adalah method profil pada controller Mahasiswa";
    }
}

```

pada line 13 kita membuat array \$data dengan element kelas, dengan nilai “MIF”

Pada line 14 kita load view dengan nama tampil\_siswa, dengan menyertakan variabel \$data pada parameter kedua saat kita load view, variabel \$data yang berisi array yang mana didalamnya terdapat 2 nilai yaitu nama dan prodi, perintah tersebut adalah proses passing data dari controller ke bagian view

```

Nama : <?php echo $nama; ?>
<br/>
Kelas : <?php echo $kelas; ?>

```

Saat kita mempassing data dari controller, kita perlu menuliskannya dalam bentuk array, dimana elemen array didalamnya (dalam contoh ini adalah nama, dan kelas) akan menjadi variabel yang dapat dipanggil dibagian view.

Modifikasi skrip dibagian function index controller Mahasiswa, menjadi seperti berikut ini :

```
public function index(){
    $data['Mahasiswa'] = [
        ["nama"=>"Kim Jefry","prodi"=>"MIF"],
        ["nama"=>"Ahmad Kurniawan","prodi"=>"TKK"],
        ["nama"=>"Budi Sudarsono","prodi"=>"TIF"]
    ];
    $this->load->view('view_mahasiswa',$data);
}
```

Modifikasi file view : view\_mahasiswa.php menjadi seperti dibawah ini :

```
<?php
foreach($siswa as $row)
{
    echo "Nama : ".$row['nama'];
    echo "<br/>";
    echo "Prodi : ".$row['prodi'];
    echo "<hr/>";
}??>
```

## Parsing data dari Model ke Views

Buatlah file model Mahasiswa\_model.php

```
<?php
class Mahasiswa_model extends CI_Model {
    public function get_data(){
        $data_mahasiswa = [
            ["nama"=>"Kim Jefry","prodi"=>"MIF"],
            ["nama"=>"Ahmad Kurniawan","prodi"=>"TKK"],
            ["nama"=>"Budi Sudarsono","prodi"=>"TIF"]
        ];
        return $data_mahasiswa;
    }??>
```

Rubahlah bagian controller pada Mahasiswa.php

```
public function index(){
    $data['Mahasiswa'] = $this->Mahasiswa_model->get_data();
    $this->load->view('view_mahasiswa',$data);
}
```

### Latihan / Soal

1. Tambahkan 2 fungsi pada controller mahasiswa dimana masing- masing menampilkan 2 view dan data yang berbeda yaitu :

- a. view\_prodi.php

Tampilan dari halaman tersebut adalah seperti berikut :

Daftar Prodi Jurusan Teknologi Informasi

No	Prodi	Keterangan
1	MIF	Manajemen Informatika
2	TKK	Teknik Komputer
3	TIF	Teknik Informatika

- b. view\_angkatan.php

Daftar Angkatan

No	Tahun
1	2016
2	2017
3	2018

- c. Rubalah tampilan view \_mahasiswa menjadi bentuk tabel
- d. Buatlah laporan hasil kegiatan praktikum dan tugas

**Acara ke : 8**  
**Pokok Bahasan : Helper**  
**Acara Praktikum : Helper**  
**Tempat : Laboratorium Rekayasa Perangkat Lunak**  
**Alokasi Waktu : 1 X 120 menit**

### **Kompetensi Dasar**

1. Mahasiswa mampu memahami konsep Helper pada CI.
2. Mahasiswa mampu menggunakan helper pada CI.

### **Bahan**

No.	Nama Bahan/ Alat	Kebutuhan	Keterangan
1.	Kertas A4 70 gram	5	Rim

### **Dasar Teori**

Helper adalah fitur yang disediakan oleh codeigniter untuk membantu anda dalam membangun aplikasi secara lebih cepat, karena didalam helper telah disediakan fungsi – fitur yang sering digunakan saat membangun aplikasi, sehingga anda tidak perlu membuat fitur – fitur yang telah disediakan didalam helper ini.

Fungsi yang ada didalam helper ini sendiri hanyalah fitur yang melakukan satu pekerjaan yang spesifik tanpa ada ketergantungan terhadap fungsi yang lain, jika nanti anda membutuhkan fungsi yang saling terkait anda bisa menggunakan fitur library yang nantinya akan kita bahas ditutorial selanjutnya. Didalam codeigniter sebenarnya telah terdapat helper yang bisa anda gunakan, helper yang telah disediakan oleh codeigniter terdapat didalam folder application/system/helpers.

### **Kegiatan Praktikum**

Terdapat 2 cara untuk dapat mengaktifkan helper yaitu :

1. Memanggilnya didalam controller

Helper bias diload pada controller dengan meletakkan perintahnya didalam function construct agar dapat digunakan diseluruh bagian class controller tersebut.

```
$this->load->helper('html');
```

2. Mengaktifkannya didalam file autoload.php dalam folder application/config.

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

## Contoh helper pada CodeIgniter

### Helper HTML

1. Tambahkan kode dibawah ini pada file autoload.php dalam folder application/config

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

2. Buatlah file controller dengan nama Testhelper dan view : view\_helper

```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Testhelper extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->helper('html');
    }

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

```
<html>
<head>
    <title>Heading Helper HTML</title>
</head>
<body>
<?php
    echo heading("Hello CI",1);
    echo heading("Hello CI ",2);
    echo heading("Hello CI ",3);
    echo heading("Hello CI ",4);
    echo heading("Hello CI ",5);
    echo heading("Hello CI ",6);

    $gambar = array(
        'src' => https://static.cdn-cdpl.com/source/17045/codeignitor_service_provider-
image(700x350-crop)-image(700x350-crop).jpg',
        'alt' => 'Codeigniter - CI',
        'class' => 'post_images',
        'width' => '300',
        'height' => '300',
        'title' => 'Artesonraju Mountain'
    );
    echo img($gambar);
?>
</body>
</html>
```

## Tugas / Latihan

### Helper Base URL

1. Download template di <https://codeload.github.com/BlackrockDigital/startbootstrap-agency/zip/gh-pages> (optional, \* ubah file index.html bawaan template menjadi .php).

2. Gunakan helper url untuk memanfaatkan fungsi base\_url pada CI sehingga file css, js dan img dari sebuah template dapat diletakkan di folder asset ( folder asset dibuat sendiri jika tidak ada, lokasi folder asset : xampp/htdocs/<folder\_ci>/asset).
3. Contoh link sebelum menggunakan base url

```
<link href="vendor/fontawesome-free/css/all.min.css" rel="stylesheet" type="text/css">
```

Setelah menerapkan base\_url

```
<link href="<?php echo base_url('asset/template1/vendor/fontawesome-free/css/all.min.css')?>" rel="stylesheet" type="text/css">
```

**Acara ke** : 9  
**Pokok Bahasan** : Library  
**Acara Praktikum** : Library  
**Tempat** : Laboratorium Rekayasa Perangkat Lunak  
**Alokasi Waktu** : 1 X 120 menit

### Kompetensi Dasar

1. Mahasiswa mampu memahami konsep library pada CI.
2. Mahasiswa mampu menggunakan library pada CI.

### Bahan

No.	Nama Bahan/ Alat	Kebutuhan	Keterangan
1.	Kertas A4 70 gram	5	Rim

### Dasar Teori

Library merupakan sebuah tempat yang digunakan untuk menempatkan beberapa file, yang didalamnya terdapat class atau subclass (class turunan) jadi saat kita menggunakan library, sebenarnya kita menggunakan class yang berada di library tersebut atau bisa juga lebih dari satu class, sehingga memungkinkan adanya keterkaitan antara 1 class dengan class lainnya. Berbeda dengan Helper diperuntukkan fungsi yang lebih sederhana, yang hanya memerlukan function – function tanpa menggunakan class didalamnya.

### Library pada CodeIgniter

1. Cart Class, library ini berfungsi untuk membuat shopping cart (keranjang belanja) digunakan untuk keperluan membuat toko online. Library ini memiliki ketergantungan terhadap kelas session karena item chart tersebut disimpan di dalam session.
2. Benchmarking Class, library ini digunakan untuk melakukan pengukuran terhadap aplikasi yang dibuat. Seperti untuk mengetahui berapa lama waktu eksekusi dan berapa jumlah memori yang digunakan. Library ini sudah digunakan dan di-load secara otomatis oleh CodeIgniter.
3. Calendar Class, library ini berfungsi untuk menampilkan dan generate sebuah kalender.



4. Config Class, library ini berfungsi untuk mengambil data data di dalam file konfigurasi. Library ini sudah di load secara otomatis oleh CodeIgniter.
5. Email Class, library email digunakan untuk mengirimkan email. Pengiriman email tersebut bisa dilakukan dengan menggunakan protokol mail, sendmail dan smtp.sqawd.
6. Encryption Class, library encryption digunakan untuk melakukan penyandian terhadap string tertentu.
7. Database Class, library database digunakan untuk memanipulasi serta mendapatkan data dari sebuah sistem database. Secara default database yang sudah diduung oleh CodeIgniter adalah mysql, mssql, oracle, postgres. Sedangkan database yang tidak didukung secara langsung oleh CodeIgniter dapat dijembatani dengan driver odbc.
8. Form Validation Class, library form validation diguakan untuk mengecek inputan form-form yang sudah di submit oleh user.
9. File Uploading Class, library uploading digunakan untuk mengupload file. Kelas ini sudah dilengkapi dengan pengecekan jenis file, dan ukuran file.
10. FTP Class, library FTP digunakan untuk mengupload atau download file melalui ftp server.
11. HTML Table Class, library HTML table adalah sebuah kelas yang berfungsi untuk generate table dari data array.
12. Input and Security Class, librari ini berfungsi untuk menjamin bahwa inputan dari form telah bersih dari karakter-karakter “aneh”.
13. Image Manipulation Class, library ini berfungsi untuk mengolah gambar. Adapun fungsi-fungsi yang telah disediakan adalah Image Resizing, Thumbnail Creation, Image Cropping, Image Rotating, dan Image Watermarking.
14. Loader Class, library ini dapat disebut sebagai pengatur sumber daya CodeIgniter. Semua sumber daya yang ada akan dikendalikan oleh kelas ini. Library ini sudah di-load secara otomatis oleh CodeIgniter.
15. Output Class, library output ini bertujuan untuk meng-handle output dari CodeIgniter, mulai dari cache sampai ke profiling bisa dilakukan kelas ini.
16. Language Class, library ini digunakan untuk mengatur bahasa apa yang akan dipakai oleh CodeIgniter.
17. Pagination Class, library ini untuk membuat paging hasil data dari database untuk performance dan usability, kita bisa mengontrol berapa banyak record untuk ditampilkan di setiap halaman website, berapa banyak record untuk ditarik dari database dan tampilan dari bagian pagination.

18. **Trackback Class**, library **trackback** ini digunakan untuk mengirim dan menerima data **trackback**.
19. **Session Class**, library **session** ini dapat digunakan untuk memelihara informasi status tentang user (seperti layaknya **session** di **PHP**). Tetapi library ini tidak menggunakan **session built-in** dari **PHP**, library **session** men-generate **session** datanya sendiri yang disimpan di dalam **Cookies**.
20. **Unit Testing Class**, library ini digunakan untuk unit test function dalam aplikasi yang sedang dibuat **CodeIgniter** menyediakan fungsi evaluasi dan dua fungsi hasil dalam library ini.
21. **Template Parser Class**, library **template parser** digunakan untuk membuat template yang berisi parsable pseudo templates.
22. **URL Class**, library **URL** digunakan untuk memarsing **URL**, lalu memecahnya ke dalam beberapa segmen dan kemudian di-passing ke controller atau disimpan sebagai variable.
23. **XML-RPC Class**, library ini digunakan untuk men-setup klien **XML-RPC** dan server.
24. **ZIP Encoding Class**, library ini digunakan untuk membuat file **ZIP** baik yang berjenis teks maupun data binary.
25. **User Agent Class**, library **user agent** digunakan untuk mengidentifikasi browser, mobile device atau robot yang mengunjungi website. Kita juga bisa menggunakannya untuk mendeteksi dukungan bahasa, sekumpulan karakter, dan referrer.

#### Load Library

1. Load didalam autoload  
file **autoload.php**, (**application/config/autoload.php**)  
*\$autoload['libraries'] = array('session');*
2. Load didalam Controller  
Pada fungsi konstruktor  
*function \_\_construct(){  
    parent::\_\_construct();  
    \$this->load->library('session');*  
}

#### Kegiatan Praktikum

1. Buatlah Controller dengan nama **Teslib**

```

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

class Teslib extends CI_Controller {
    function __construct(){
        parent::__construct();
        $this->load->library('table');
        $this->load->model('Mahasiswa_model');
    }

    public function index()
    {
        $data['Mahasiswa'] = $this->Mahasiswa_model->get_data();
        $this->load->view('view_teslib',$data);
    }
}

```

## 2. Buatlah file view : view\_teslib.php

```

<?php

$template = array(
    'table_open'=> '<table border="1 cellpadding="4 cellspacing="0">',
    'table_close'=> '</table>'
);

$this->table->set_template($template);
$this->table->set_heading(array('No', 'Nama', 'Prodi'));
$num = 1;

foreach($Mahasiswa as $row)
{

    $this->table->add_row(array($num,$row['nama'],$row['prodi']));
    $num++;
}

echo $this->table->generate();
?>

```

## Soal / Latihan

1. Cobalah Latihan berikut dan buat laporannya !
2. Buatlah file Lingkaran di folder application/libraries

```

<?php
class Lingkaran{

    function keliling($jari){
        $keliling = 2*3.14*$jari;
        echo "Luas Lingkaran dengan jari-jari : ".$jari." ,Adalah = ".$keliling;
    }

    function luas($jari){
        $luas = 3.14*$jari*$jari;
        echo "Luas Lingkaran dengan jari-jari : ".$jari." ,Adalah = ".$luas;
    }

}

```

### 3. Buat Controller Hitunglingkaran

```

defined('BASEPATH') OR exit('No direct script access allowed');
class Hitunglingkaran extends CI_Controller {

    function __construct(){
        parent::__construct();
        $this->load->library('lingkaran');
    }

    function index(){
        $this->lingkaran->keliling('21');
        echo "<hr/>";
        $this->lingkaran->luas('14');
    }

}

```

### 4. Cobalah Benchmarkingmark pada CI.

**Acara ke** : 10  
**Pokok Bahasan** : Template  
**Acara Praktikum** : Template  
**Tempat** : Laboratorium Rekayasa Perangkat Lunak  
**Alokasi Waktu** : 1 X 120 menit

### Kompetensi Dasar

1. Mahasiswa mampu memahami konsep partial pada CI.
2. Mahasiswa mampu memecah template pada CI.

### Bahan

No.	Nama Bahan/ Alat	Kebutuhan	Keterangan
1.	Kertas A4 70 gram	5	Rim

### Dasar Teori

Partial merupakan teknik untuk membagi template menjadi bagian-bagian kecil agar mudah digunakan. Perhatikan gambar dibawah ini secara umum bagian-bagian dari sebuah halaman tampak seperti gambar ini:

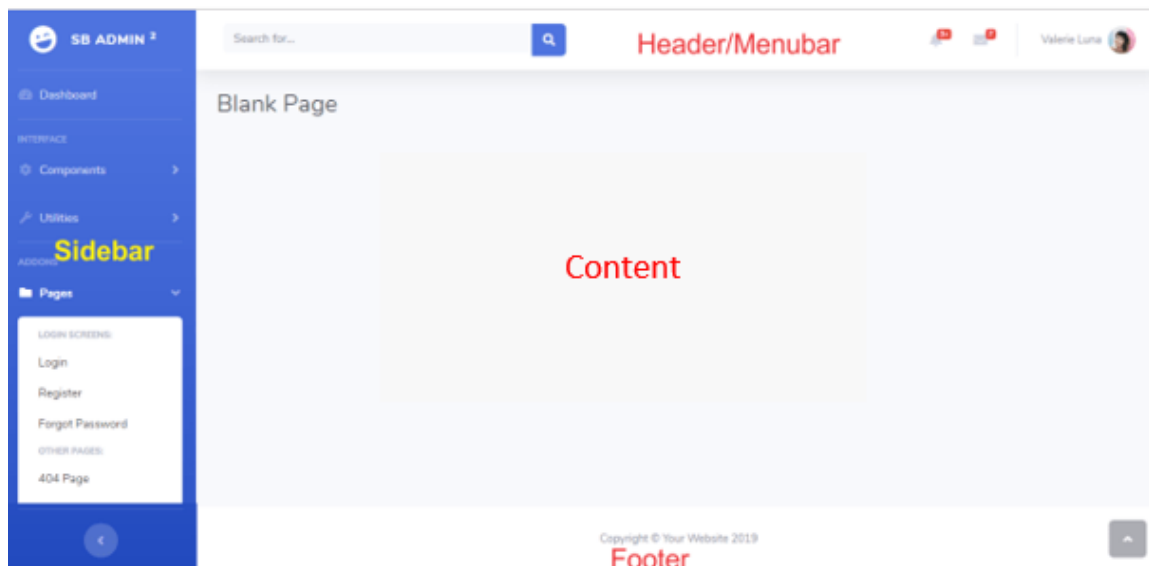


Secara garis besar template di bagi menjadi 3 bagian : Header, Content dan Footer

Untuk bagian header dan footer akan bersifat statis, dan untuk bagian content akan bersifat dinamis dimana akan menyesuaikan dari fitur yang sedang diakses.

### Kegiatan Praktikum

1. Download template SBAdmin di : <https://startbootstrap.com/themes/sb-admin-2/>
2. [Extrak folder tersebut.](#)
3. Kita akan membagi sbadmin menjadi beberapa partial



Berdasarkan gambar di atas, berikut ini partial yang bisa kita buat:

- a. head.php untuk menyimpan isi dari tag <head>;
  - b. sidebar.php untuk menyimpan kode menu bagian samping (sidebar);
  - c. topbar.php untuk menyimpan menu bagian atas(Head/menubar);
  - d. footer.php untuk menyimpan kode footer;
  - e. js.php untuk meload javascript.
  - f. content.php untuk meload javascript.
4. Buka file blank.html pada hasil ekstrak sbadmin2.
  5. Pindahkan semua folder css, js,img, scss dan vendor ke folder asset (pastikan helper url sudah ada di autoload).
  6. Rubah semua link yang mengarah ke folder-folder berikut ke folder asset dengan helper base\_url.
  7. Buat folder di Application/View dengan nama Admin dan didalamnya kita create folder \_Template
  8. Buat file head.php didalam folder \_template, copy bagian head dari blank.html kedalam file head.php ( line 4-21).

9. Buat file sidebar.php di view yang berisi line 28 – 140 dari blank.html.

```
27
28     <!-- Sidebar -->
29     <ul class="navbar-nav bg-gradient-primary sidebar sidebar-c
140     <!-- End of Sidebar -->
141
```

10. Buat file sidebar.php di view yang berisi line 148 – 329 dari blank.html.

```
148     <!-- Topbar -->
149     <nav class="navbar navbar-expand navbar-light bg-white topbar
329     <!-- End of Topbar -->
330
```

11. Buat file content.php di view yang berisi :

```
330
331     <!-- Begin Page Content -->
332     <div class="container-fluid">
333
334         <!-- Page Heading -->
335         <!--<h1 class="h3 mb-4 text-gray-8
336         <?php echo @$konten ;?>
337
338     </div>
339     <!-- /.container-fluid -->
340
```

12. Buat file sidebar.php di view yang berisi line 148 – 329 dari blank.html.

```
343
344     <!-- Footer -->
345     <footer class="sticky-footer bg-white">
346         <div class="container my-auto">
347             <div class="copyright text-center my-auto">
348                 <span>Copyright &copy; Your Website 2019</span>
349             </div>
350         </div>
351     </footer>
352     <!-- End of Footer -->
```

13. Buat file content.php di view yang berisi :

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <?php echo @$head; ?>
4
5 <body id="page-top">
6
7     <!-- Page Wrapper -->
8     <div id="wrapper">
9
10         <!-- Sidebar -->
11         <?php echo @$sidebar; ?>
12         <!-- End of Sidebar -->
13
14         <!-- Content Wrapper -->
15         <div id="content-wrapper" class="d-flex flex-column">
```

```

16
17     <!-- Main Content -->
18     <div id="content">
19
20         <!-- Topbar -->
21         <?php echo @$topbar; ?>
22         <!-- End of Topbar -->
23
24         <!-- Begin Page Content -->
25         <?php echo @$content; ?>
26         <!-- /.container-fluid -->
27
28     </div>
29     <!-- End of Main Content -->
30
31     <!-- Footer -->
32     <?php echo @$footer; ?>
33     <!-- End of Footer -->
34
35 </div>
36 <!-- End of Content Wrapper -->

```

```

37 </div>
38 <!-- End of Page Wrapper -->
39
40
41 <!-- Scroll to Top Button-->
42 <a class="scroll-to-top rounded" href="#page-top">
43     <i class="fas fa-angle-up"></i>
44 </a>
45
46 <!-- Logout Modal-->
47 <div class="modal fade" id="logoutModal" tabindex="-1" role="dialog"
48     aria-labelledby="exampleModalLabel" aria-hidden="true">
49     <div class="modal-dialog" role="document">
50         <div class="modal-content">
51             <div class="modal-header">
52                 <h5 class="modal-title" id="exampleModalLabel">Ready to Leave?</h5>
53                 <button class="close" type="button" data-dismiss="modal" aria-label="Close">
54                     <span aria-hidden="true">x</span>
55                 </button>
56             </div>
57             <div class="modal-body">Select "Logout" below if you are ready to end your current session.</div>
58             <div class="modal-footer">
59                 <button class="btn btn-secondary" type="button" data-dismiss="modal">Cancel</button>
60                 <a class="btn btn-primary" href="login.html">Logout</a>
61             </div>
62         </div>
63     </div>
64 </div>

```

```

65 <!-- Bootstrap core JavaScript-->
66 <script src="<?php echo base_url('asset/vendor/jquery/jquery.min.js') ?>"></script>
67 <script src="<?php echo base_url('asset/vendor/bootstrap/js/bootstrap.bundle.min.js') ?>"
68 ></script>
69
70 <!-- Core plugin JavaScript-->
71 <script src="<?php echo base_url('asset/vendor/jquery-easing/jquery.easing.min.js') ?>"
72 ></script>
73
74 <!-- Custom scripts for all pages-->
75 <script src="<?php echo base_url('asset/js/sb-admin-2.min.js') ?>"></script>
</body>
</html>

```

14. Buat file template.php di folder library



```

<?php
class Template {
    protected $_ci;
    function __construct() {
        $this->_ci = &get_instance();
    }
    function views($template = NULL, $data = NULL) {
        if ($template != NULL) {
            // Bagian $data['head'] untuk memanggil file head.php dari folder
Admin/_Template/
            // ['head'] data yang kita panggil dari file template.php dari folder
Admin/_Template/
            $data['head'] = $this->_ci->load->view('Admin/_Template/head', $data,
TRUE);

            // Bagian $data['topbar'] untuk memanggil file topbar.php dari folder
Admin/_Template/
            // ['topbar'] data yang kita panggil dari file template.php dari folder
Admin/_Template/
            $data['topbar'] = $this->_ci->load->view('Admin/_Template/topbar', $data,
TRUE);

            // Bagian $data['sidebar'] untuk memanggil file sidebar.php dari folder
Admin/_Template/
            // ['sidebae'] data yang kita panggil dari file template.php dari folder
Admin/_Template/
            $data['sidebar'] = $this->_ci->load->view('Admin/_Template/sidebar',
$data, TRUE);

            // ['isi'] data yang kita panggil dari file content.php dari folder
Admin/_Template/
            $data['isi'] = $this->_ci->load->view($template, $data, TRUE);

            // Bagian $data['content'] untuk memanggil file content.php dari folder
Admin/_Template/
            // ['content'] data yang kita panggil dari file template.php dari folder
Admin/_Template/
            $data['content'] = $this->_ci->load->view('Admin/_Template/content', $data,
TRUE);

            // Bagian $data['footer'] untuk memanggil file footer.php dari folder
Admin/_Template/
            // ['footer'] data yang kita panggil dari file template.php dari folder
Admin/_Template/
            $data['footer'] = $this->_ci->load->view('Admin/_Template/footer', $data, TRUE);

            // Bagian $data['template'] untuk menampilkan file template.php dari folder
Admin/_Template/
            // view('Admin/_Template/Template', $data, TRUE); untuk memanggil $data diatas
seperti $data['head'], dll
            echo $data['Template']= $this->_ci->load->view('Admin/_Template/Template',
$data, TRUE); }}}>

```

