

# **LAPORAN PEMROGRAMAN WEB FRAMEWORK**

**“Helper”**



**DISUSUN OLEH :**

Ratih Cahyani Apriliya E31191660

**GOLONGAN C**

**MANAJEMEN INFORMATIKA**

**TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI JEMBER**

**2021**

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam pembangunan aplikasi perlu adanya alat bantu untuk mendukung keberhasilan dalam pembangunan. Tanpa adanya alat bantu yang mendukung maka waktu dan energi yang dibutuhkan akan semakin banyak, dan pekerjaan tidak cepat terselesaikan sesuai jadwal yang sudah ditentukan. Faktor kesulitan dalam pembangunan aplikasi berada pada pengulangan kode program yang sama, salah satu pengulangan kode program yaitu dalam pembuatan formulir dan laporan. Untuk mendukung keberhasilan maka dibutuhkan alat bantu yaitu framework PHP. Framework PHP akan mendukung kelancaran dalam pembangunan aplikasi sehingga pekerjaan cepat terselesaikan tepat waktu sesuai jadwal yang ditentukan. Framework adalah kerangka kerja, sedangkan PHP adalah bahasa script yang dapat ditanamkan atau disisipkan ke dalam HTML untuk membuat website, maka framework PHP dalam kasus ini bisa diartikan sebagai aplikasi yang didalamnya terdapat beberapa kode program serta fungsi – fungsi yang tergabung menjadi satu yang di standartkan guna untuk mempermudah dalam pembuatan program aplikasi yang dapat di modifikasi secara berkala. Penggunaan Framework PHP bertujuan untuk memudahkan dalam membuat aplikasi dibandingkan apabila kita membuat semuanya mulai dari awal

### **1.2 Rumusan Masalah**

- 1.2.1 Bagaimana konsep helper pada CI?
- 1.2.2 Bagaimana menerapkan helper data pada CI?

### **1.3 Tujuan**

- 1.3.1 Mahasiswa dapat memahami konsep helper pada CI.
- 1.3.2 Mahasiswa dapat menerapkan helper data pada CI.

## **BAB II**

### **DASAR TEORI**

#### **2.1 Helper**

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././badan class.

#### **2.2 Cara mengaktifkan Helper di codeigniter**

Jika anda ingin menggunakan helper di codeigniter, anda perlu mengaktifkan terlebih dahulu helper tersebut agar dapat digunakan.

Terdapat 2 cara yang bisa anda gunakan untuk dapat mengaktifkan helper yaitu :

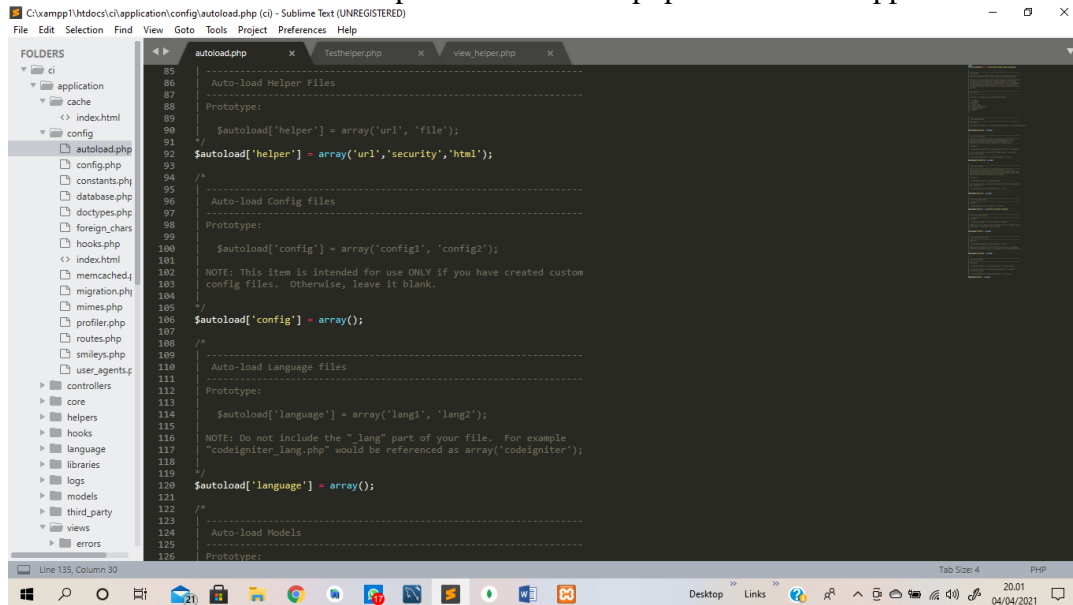
1. Memanggilnya didalam controller
2. Mengaktifkannya didalam file autoload.php dalam folder application/config

## BAB III

### HASIL DAN PEMBAHASAN

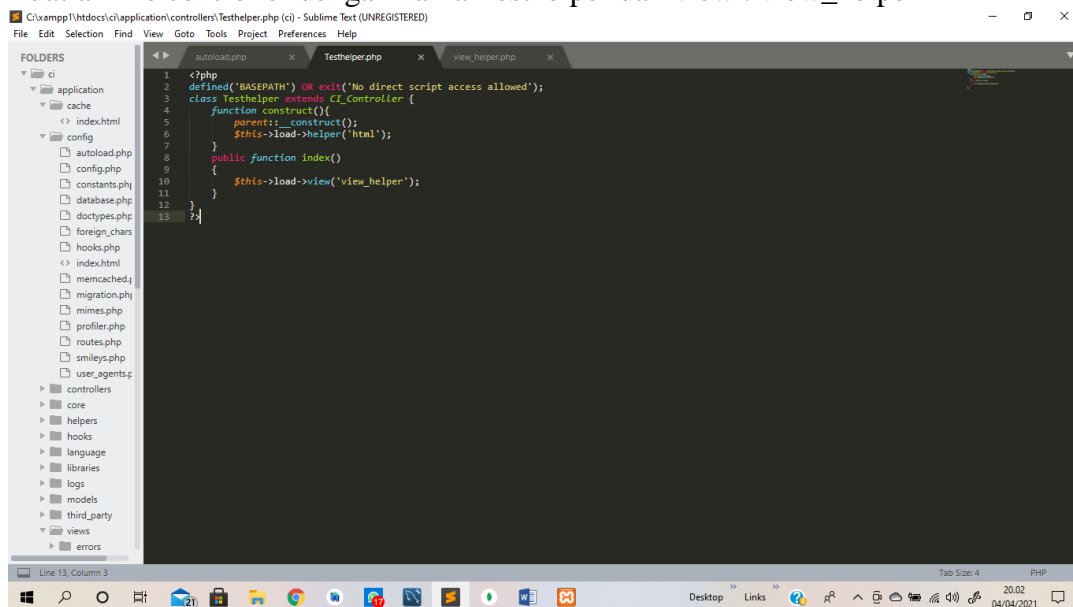
#### 3.1 Kegiatan Praktikum

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

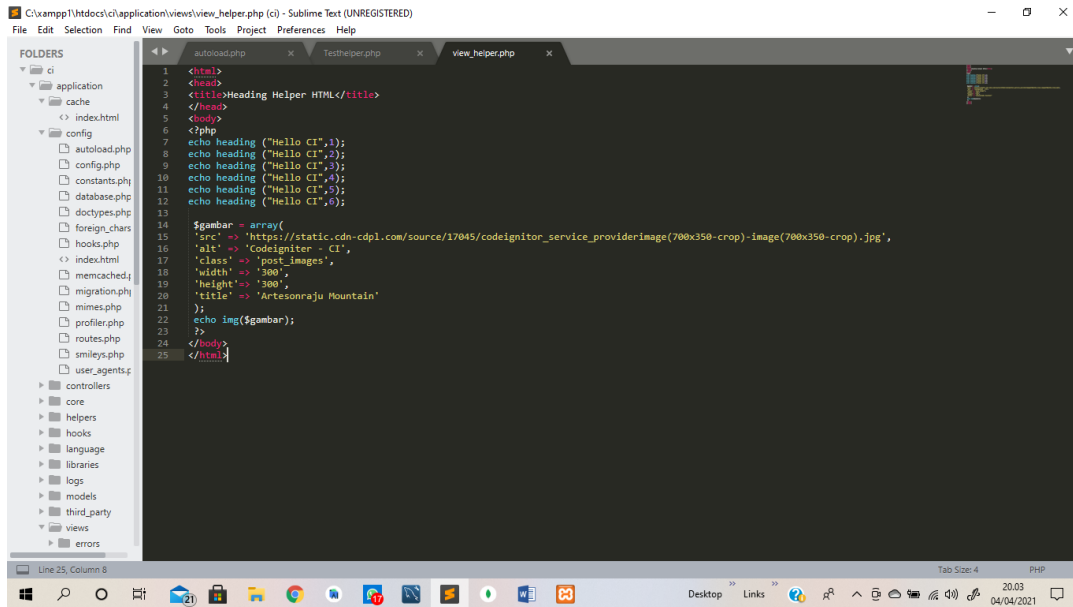


```
85
86 Auto-load Helper Files
87 -----
88 Prototype:
89 $autoload['helper'] = array('url', 'file');
90
91
92 $autoload['helper'] = array('url', 'security', 'html');
93
94
95 Auto-load Config files
96 -----
97 Prototype:
98 $autoload['config'] = array('config1', 'config2');
99
100
101 NOTE: This item is intended for use ONLY if you have created custom
102 config files. Otherwise, leave it blank.
103
104
105 $autoload['config'] = array();
106
107
108
109 Auto-load Language files
110 -----
111 Prototype:
112 $autoload['language'] = array('lang1', 'lang2');
113
114
115 NOTE: Do not include the ".lang" part of your file. For example
116 "codeigniter_lang.php" would be referenced as array('codeigniter');
117
118
119 $autoload['language'] = array();
120
121
122
123 Auto-load Models
124 -----
125 Prototype:
126
```

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

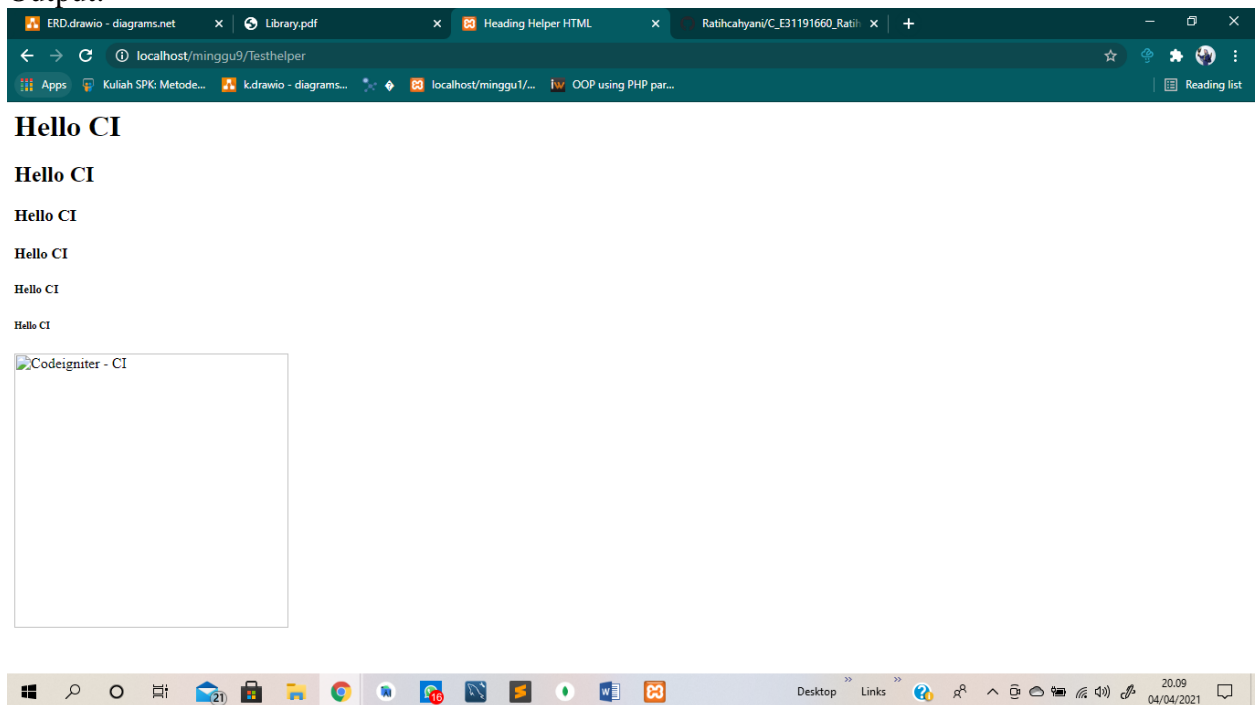


```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3 class Testhelper extends CI_Controller {
4 function construct() {
5 parent::__construct();
6 $this->load->helper('html');
7 }
8 public function index()
9 {
10 $this->load->view('view_helper');
11 }
12 }
13 ?
```



```
1 <?php
2 <html>
3 <head>
4 <title>Heading Helper HTML</title>
5 </head>
6 <body>
7 <?php
8 echo heading ("Hello CI",1);
9 echo heading ("Hello CI",2);
10 echo heading ("Hello CI",3);
11 echo heading ("Hello CI",4);
12 echo heading ("Hello CI",5);
13
14 $gambar = array(
15 'src' => 'https://static.cdn-cdp1.com/source/17045/codeignitor_service_providerimage(700x350-crop)-image(700x350-crop)-.jpg',
16 'alt' => 'Codeigniter - CI',
17 'class' => 'post_images',
18 'width' => '300',
19 'height' => '300',
20 'title' => 'Artesonraju Mountain'
21 );
22 echo img($gambar);
23 ?>
24 </body>
25 </html>
```

Output:



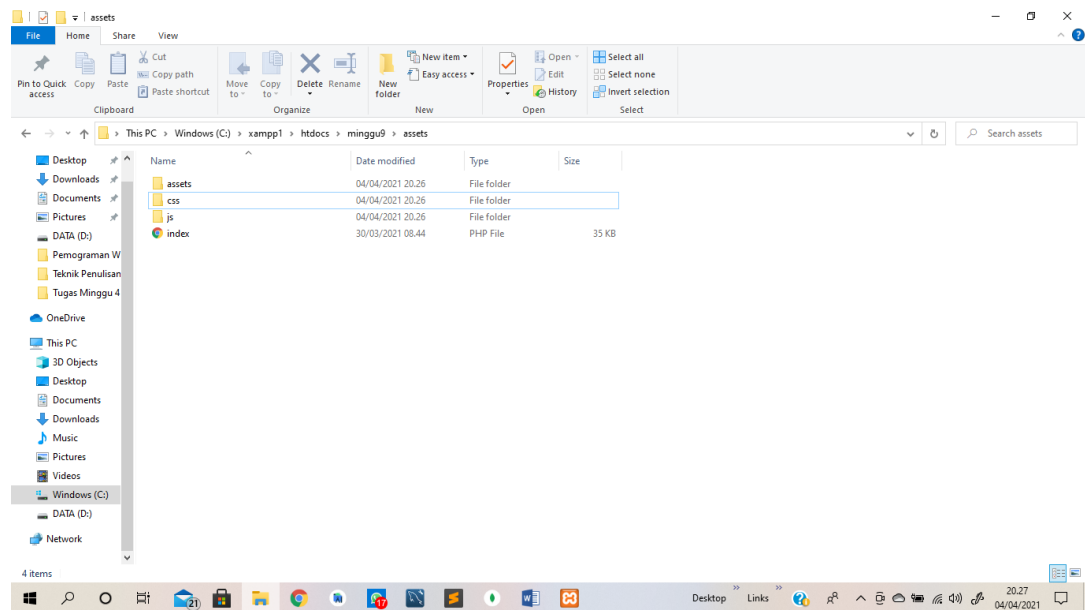
### 3.2 Tugas / Latihan Helper Base URL

1. Download template di

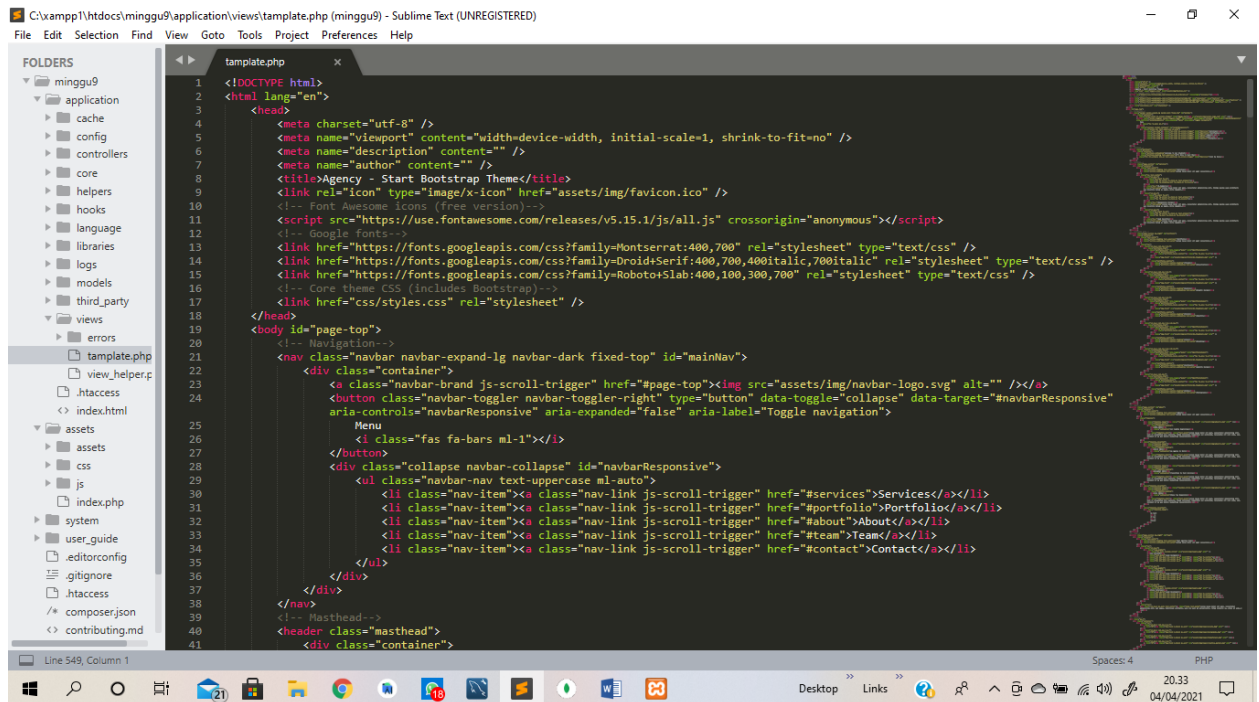
<https://codeload.github.com/BlackrockDigital/startbootstrapagency/zip/gh-pages> (optional, \*

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//asset).

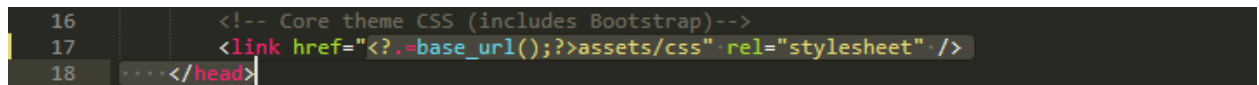


3. Buat file didalam folder view dengan nama template.php (nama sesuai keinginan), dan setelah itu copy code index.html ke dalam template .php



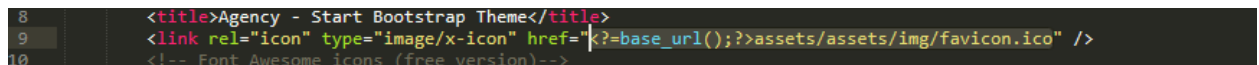
```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4 <meta charset="utf-8" />
5 <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no" />
6 <meta name="description" content="" />
7 <meta name="author" content="" />
8 <title>Agency - Start Bootstrap Theme</title>
9 <link rel="icon" type="image/x-icon" href="assets/img/favicon.ico" />
10 <!-- Font Awesome icons (free version) -->
11 <script src="https://use.fontawesome.com/releases/v5.15.1/js/all.js" crossorigin="anonymous"></script>
12 <!-- Google Fonts -->
13 <link href="https://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css" />
14 <link href="https://fonts.googleapis.com/css?family=Droid+Serif:400,700,400italic,700italic" rel="stylesheet" type="text/css" />
15 <link href="https://fonts.googleapis.com/css?family=Roboto+Slab:400,100,300,700" rel="stylesheet" type="text/css" />
16 <!-- Core theme CSS (includes Bootstrap) -->
17 <link href="css/styles.css" rel="stylesheet" />
18 </head>
19 <body id="page-top">
20 <!-- Navigation -->
21 <nav class="navbar navbar-expand-lg navbar-dark fixed-top" id="mainNav">
22 <div class="container">
23 <a class="navbar-brand js-scroll-trigger" href="#page-top"></a>
24 <button class="navbar-toggler navbar-toggler-right" type="button" data-toggle="collapse" data-target="#navbarResponsive"
25   aria-controls="navbarResponsive" aria-expanded="false" aria-label="Toggle navigation">
26   Menu
27   </button>
28 <div class="collapse navbar-collapse" id="navbarResponsive">
29 <ul class="navbar-nav text-uppercase ml-auto">
30 <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#services">Services</a></li>
31 <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#portfolio">Portfolio</a></li>
32 <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#about">About</a></li>
33 <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#team">Team</a></li>
34 <li class="nav-item"><a class="nav-link js-scroll-trigger" href="#contact">Contact</a></li>
35 </ul>
36 </div>
37 </div>
38 </nav>
39 <!-- Masthead -->
40 <header class="masthead">
41 <div class="container">
```

4. Cari code link href, samakan dengan code dibawah ini.



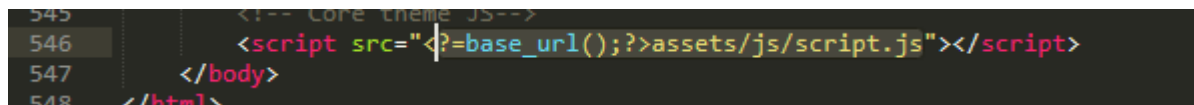
```
16 <!-- Core theme CSS (includes Bootstrap) -->
17 <link href="<?=base_url();?>assets/css" rel="stylesheet" />
18 <!--></head>
```

5. Cari code link rel icon, samakan dengan code dibawah ini.



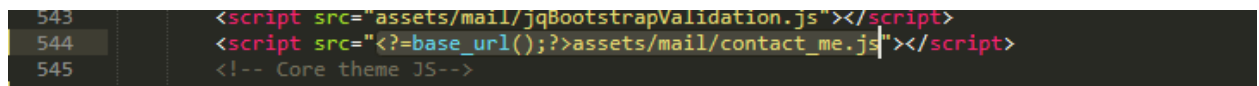
```
8 <title>Agency - Start Bootstrap Theme</title>
9 <link rel="icon" type="image/x-icon" href="<?=base_url();?>assets/assets/img/favicon.ico" />
10 <!-- Font Awesome icons (free version) -->
```

6. Cari code script src js, samakan dengan code dibawah ini.



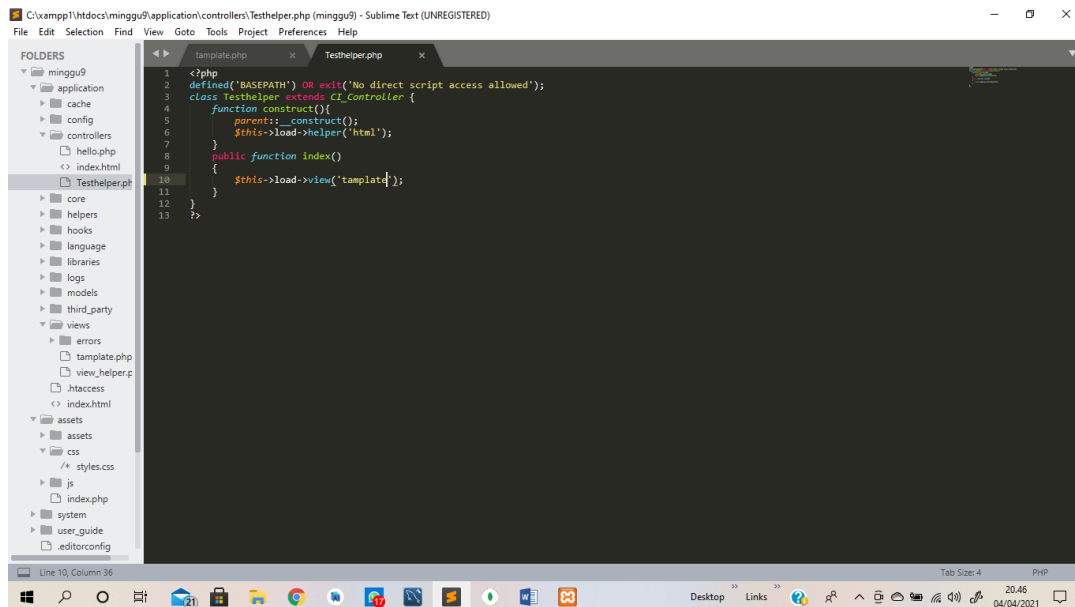
```
545 <!-- Core theme JS -->
546 <script src="<?=base_url();?>assets/js/script.js"></script>
547 </body>
548 </html>
```

7. Cari code script src mail, samakan dengan code dibawah ini.



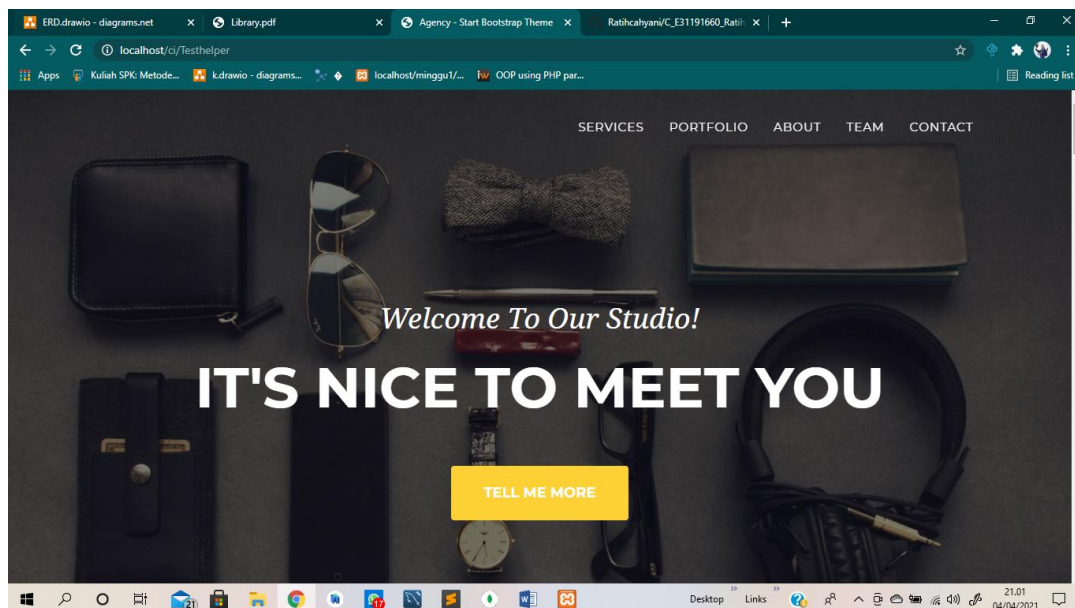
```
543 <script src="assets/mail/jqbootstrapValidation.js"></script>
544 <script src="<?=base_url();?>assets/mail/contact_me.js"></script>
545 <!-- Core theme JS -->
```

8. Jika sudah, deklarasikan view di controller dan panggil di browser.



```
<?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('template');
    }
}
```

9. Hasil :







## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Helper adalah kumpulan fungsi yang diletakkan di dalam folder `system/helpers` atau `applications/helpers`. Biasanya helper sering digunakan dalam view untuk membantu prosesproses yang berulang, seperti generate html, url, security, dan lain-lain.