

**TUGAS PEMROGRAMAN WEB FRAMEWORK**  
**BKPM MINGGU 7**



Nama : Agna Rizky Putra A  
NIM / Golongan : E31191434 /C

**MANAJEMEN INFORMATIKA**  
**TEKNOLOGI INFORMASI**  
**POLITEKNIK NEGERI JEMBER 2020**

## BAB I

### DASAR TEORI

Router di CodeigniterRouter pada framework codeigniter, memiliki tugas untuk menentukan controller serta method/fungsi yang akan dijalankan ketika pengguna aplikasi mengakses alamat/url tertentu. Ketika kita mengakses <http://localhost/cimaka> akan muncul tampilan welcome to ci, hal tersebut karena default controller mengarah pada file routers.php, anda bisa mengakses file routers.php didalam direktori application/config/routers.php.

Perhatikan pada line : 52 –54

```
1 $route['default_controller'] = 'welcome';
2 $route['404_override'] = '';
3 $route['translate_uri_dashes'] = FALSE;
```

Keterangan :

- `$route['default_controller'] = 'welcome'` ini merupakan pengaturan default controller yang otomatis akan dipanggil ketika halaman base\_url web diakses, base url disini adalah alamat utama dari web, disitu kita menulis welcome artinya akan mengakses controller welcome, controller welcome adalah controller default yang merupakan bawaan codeigniter, untuk file controller berada di application\controllers, nah pada controller welcome, yang akan dijalankan awal adalah function index, pada function index tersebut menjalankan view welcome\_message, dimana file view ini berada pada direktori application\views, anda bisa mengganti nilai pada nilai default\_controller, untuk mengarahkan ke controller tertentu saat base\_url diakses
- `$route['404_override'] = ''` merupakan pengaturan default controller yang akan diakses apabila halaman default controller tidak ditemukan, ataupun sebuah controller lainnya tidak ditemukan.
- `$route['translate_uri_dashes'] = FALSE`, ini adalah pengaturan yang memperbolehkan anda menggunakan tanda dash (-) pada bagian url, anda bisa menggantinya dengan nilai TRUE, sebagai controller semisal anda memiliki controller dengan nama produk\_makanan maka kita dapat mengakses pada urlnya menjadi produk-makanan

Tampilan dari link : <http://localhost/ci/index.php/welcome/index> akan sama dengan tampilan dari link : <http://localhost/ci>. Hal tersebut dikarenakan saat kita mengakses : <http://localhost/ci/index.php/welcome/index>, kita sedang mengakses function index

didalam controller welcome. Jadi untuk mengakses function / method didalam controller, kita perlu menuliskan :

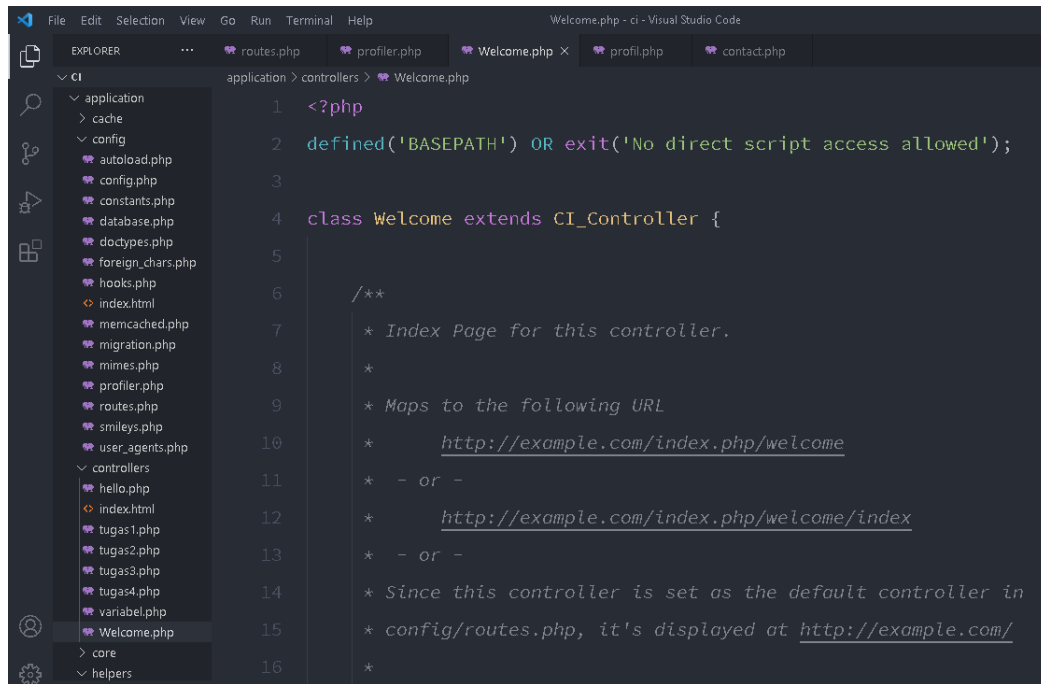
- Base\_url : localhost/belajarcodigniter
- Lalu tambahkan index.php
- Lalu tuliskan nama controller
- Lalu tuliskan nama function

## BAB II

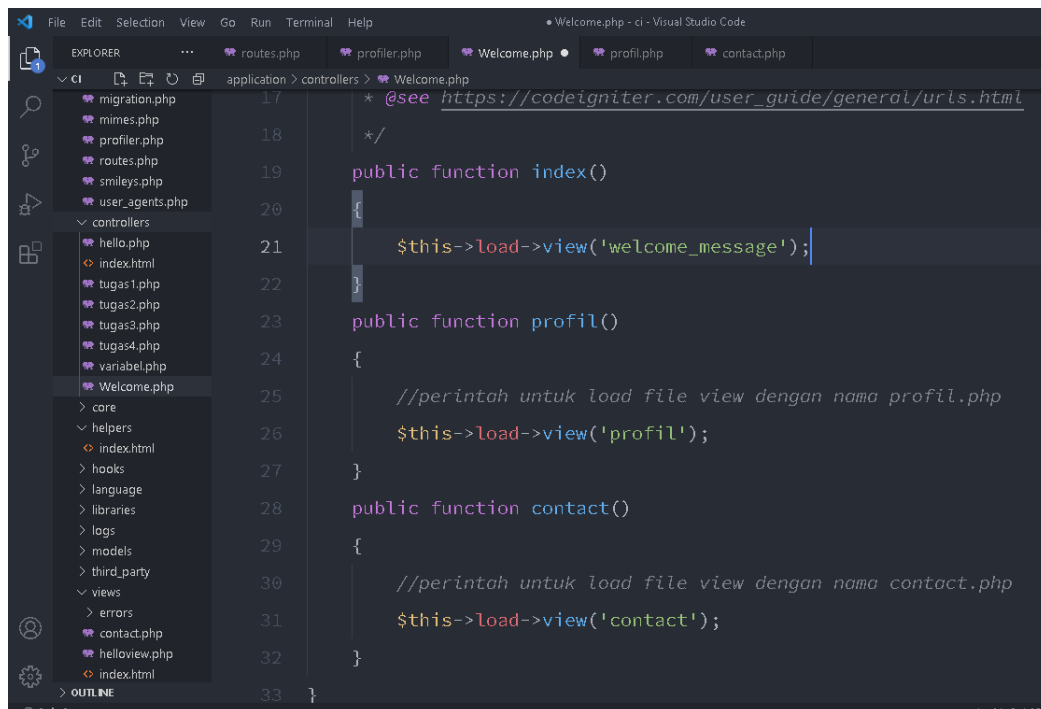
### HASIL TUGAS PRAKTIKUM

- Membuat Beberapa Router

Buat 2 function dengan kode dibawah ini, tuliskan didalam controller Welcome, dibawah function index.



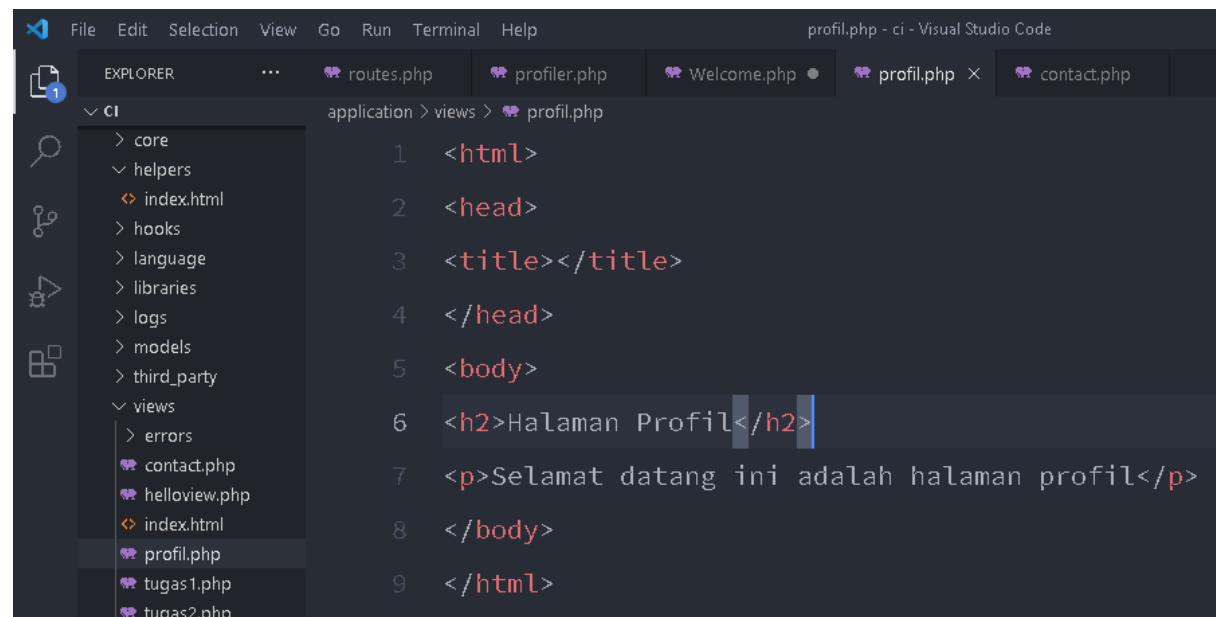
```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Welcome extends CI_Controller {
5
6     /**
7      * Index Page for this controller.
8      *
9      * Maps to the following URL
10      *      http://example.com/index.php/welcome
11      * - or -
12      *      http://example.com/index.php/welcome/index
13      * - or -
14      * Since this controller is set as the default controller in
15      * config/routes.php, it's displayed at http://example.com/
16      *
17      */
```



```
17 * @see https://codeigniter.com/user\_guide/general/urls.html
18 */
19 public function index()
20 {
21     $this->load->view('welcome_message');
22 }
23 public function profil()
24 {
25     //perintah untuk load file view dengan nama profil.php
26     $this->load->view('profil');
27 }
28 public function contact()
29 {
30     //perintah untuk load file view dengan nama contact.php
31     $this->load->view('contact');
32 }
33 }
```

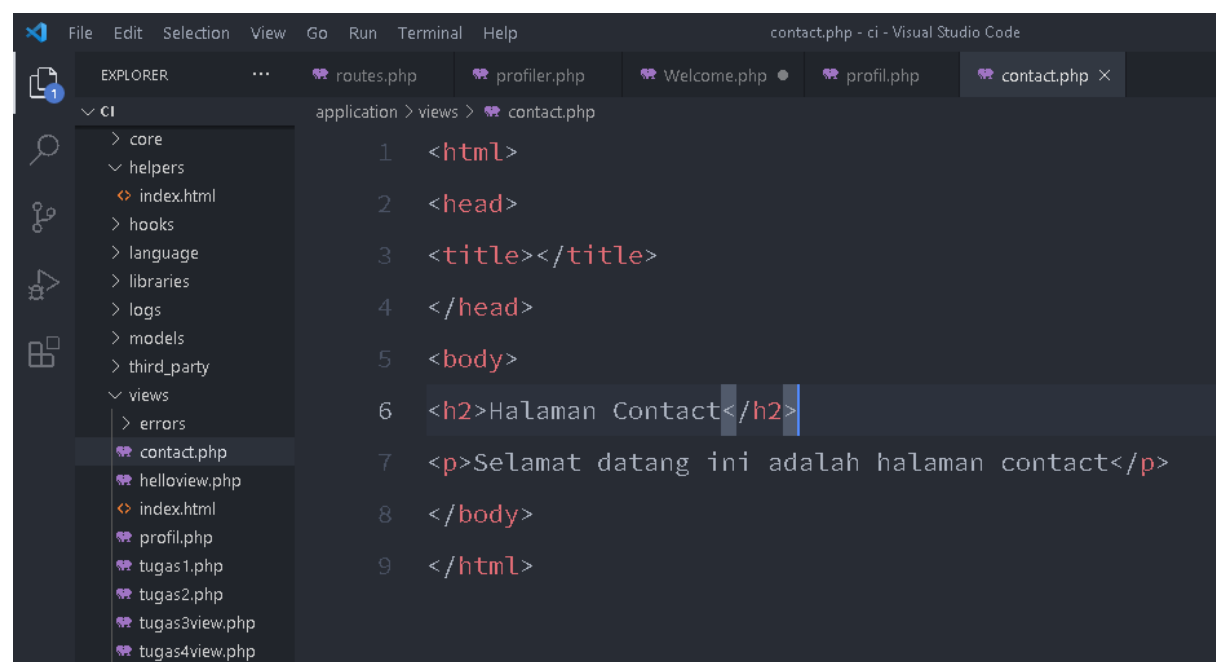
- Berikutnya silahkan buat 2 file view

dengan nama profil.php, dan contact.php didalam direktori application/views.



The screenshot shows the Visual Studio Code interface with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the 'views' directory expanded, listing files like errors, contact.php, helloview.php, index.html, profil.php, tugas1.php, and tugas2.php. The code editor shows the content of 'profil.php' with the following HTML code:

```
1 <html>
2 <head>
3 <title></title>
4 </head>
5 <body>
6 <h2>Halaman Profil</h2>
7 <p>Selamat datang ini adalah halaman profil</p>
8 </body>
9 </html>
```



The screenshot shows the Visual Studio Code interface with the file explorer on the left and the code editor on the right. The file explorer shows the project structure with the 'views' directory expanded, listing files like errors, contact.php, helloview.php, index.html, profil.php, tugas1.php, tugas2.php, tugas3view.php, and tugas4view.php. The code editor shows the content of 'contact.php' with the following HTML code:

```
1 <html>
2 <head>
3 <title></title>
4 </head>
5 <body>
6 <h2>Halaman Contact</h2>
7 <p>Selamat datang ini adalah halaman contact</p>
8 </body>
9 </html>
```

- Membuat perintah agar bisa memanggil profil.php dan contact.php

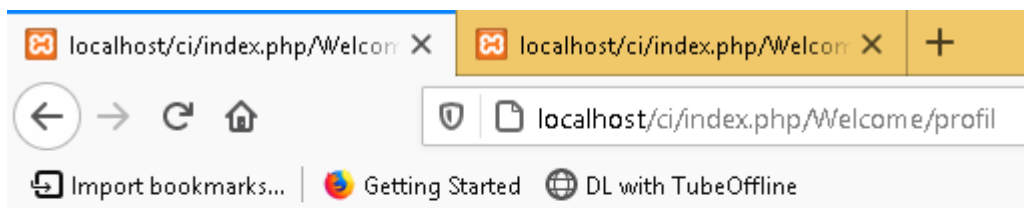
```

44 | controller and method names that contain dashes. '-' isn't
45 | class or method name character, so it requires translation
46 | When you set this option to TRUE, it will replace ALL dashes
47 | controller and method URI segments.
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
56 $route['profil'] = 'welcome/profil';
57 $route['contact'] = 'welcome/contact';

```

- Hasil

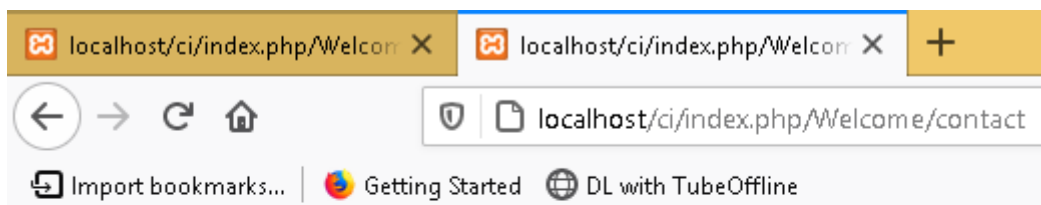
File profil



## Halaman Profil

Selamat datang ini adalah halaman profil

File contact



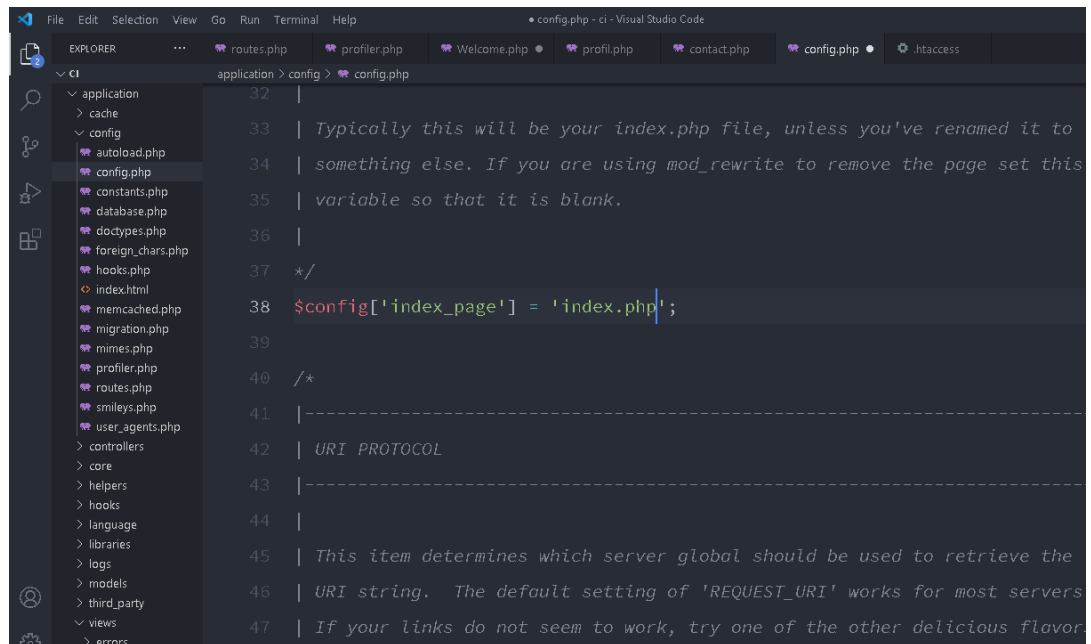
## Halaman Contact

Selamat datang ini adalah halaman contact

- **TUGAS / SOAL**

1. Untuk mengakses file pada CI normalnya link adalah : localhost/nama\_folder/index.php/controller. Rubahlah agar aksesnya menjadi localhost/ci/controller.

1.1. Cara pertama buka file config yang berada di application/config/config.php dan cari pengaturan index\_page

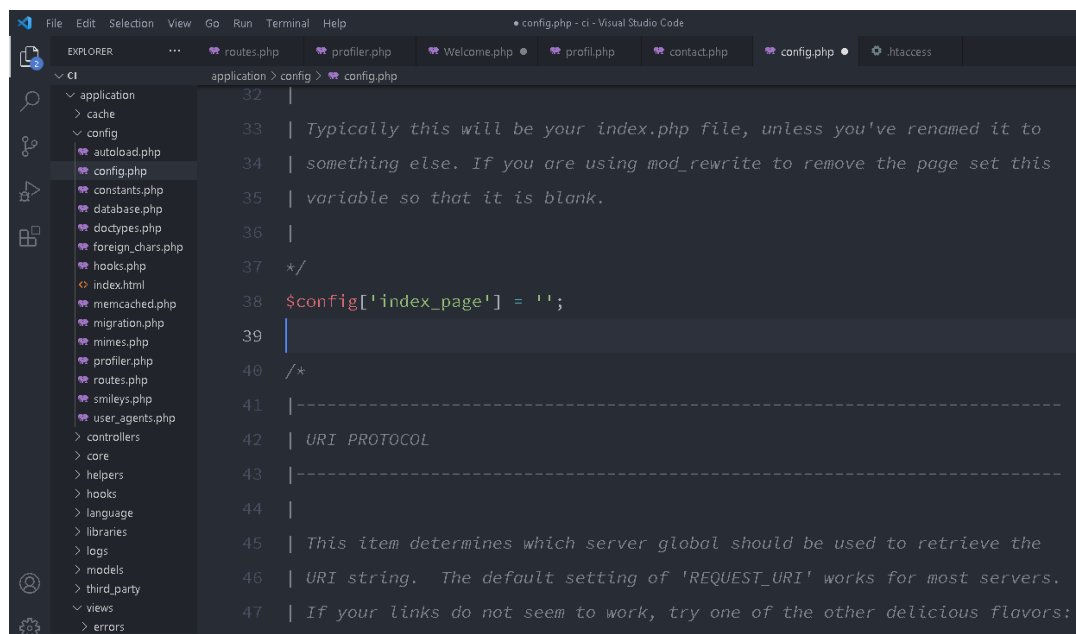


```
File Edit Selection View Go Run Terminal Help
• config.php - ci - Visual Studio Code

EXPLORER
  application
    cache
    config
      config.php
      constants.php
      database.php
      doctypes.php
      foreign_chars.php
      hooks.php
      index.html
      memcached.php
      migration.php
      mimes.php
      profiler.php
      routes.php
      smileys.php
      user_agents.php
    controllers
    core
    helpers
    hooks
    language
    libraries
    logs
    models
    third_party
    views
    errors

32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 | */
38 | $config['index_page'] = 'index.php';
39 |
40 | /*
41 | -----
42 | URI PROTOCOL
43 | -----
44 |
45 | This item determines which server global should be used to retrieve the
46 | URI string. The default setting of 'REQUEST_URI' works for most servers.
47 | If your links do not seem to work, try one of the other delicious flavors:
```

1.2. Setelah itu hapus pada bagian index.php sehingga hasil nya akan seperti ini

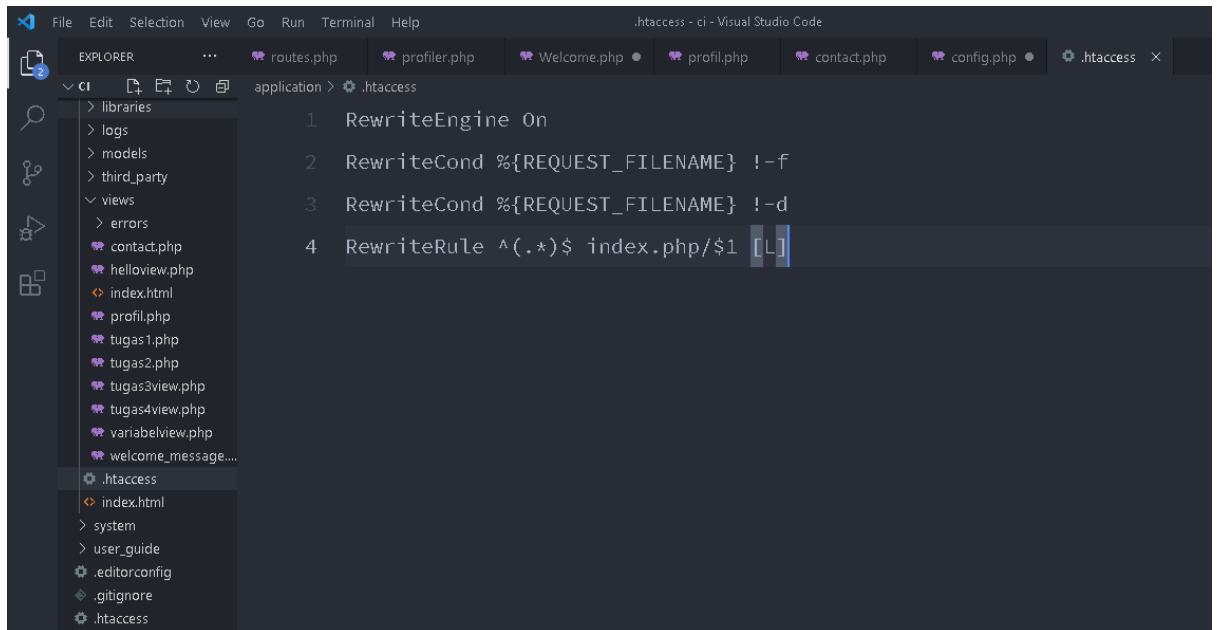


```
File Edit Selection View Go Run Terminal Help
• config.php - ci - Visual Studio Code

EXPLORER
  application
    cache
    config
      config.php
      constants.php
      database.php
      doctypes.php
      foreign_chars.php
      hooks.php
      index.html
      memcached.php
      migration.php
      mimes.php
      profiler.php
      routes.php
      smileys.php
      user_agents.php
    controllers
    core
    helpers
    hooks
    language
    libraries
    logs
    models
    third_party
    views
    errors

32 |
33 | Typically this will be your index.php file, unless you've renamed it to
34 | something else. If you are using mod_rewrite to remove the page set this
35 | variable so that it is blank.
36 |
37 | */
38 | $config['index_page'] = '';
39 |
40 | /*
41 | -----
42 | URI PROTOCOL
43 | -----
44 |
45 | This item determines which server global should be used to retrieve the
46 | URI string. The default setting of 'REQUEST_URI' works for most servers.
47 | If your links do not seem to work, try one of the other delicious flavors:
```

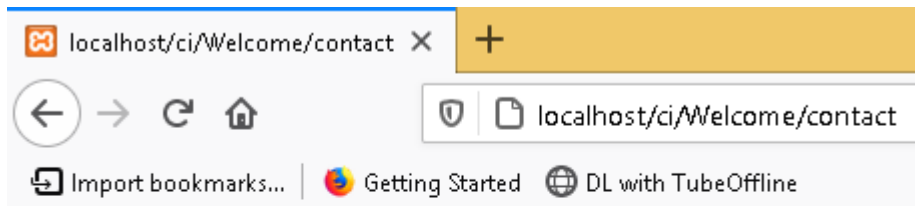
- 1.3. Setelah itu membuat file dengan nama '.htaccess' pada direktori file codeigniter dan disini file '.htaccess' tidak ber-ekstensi karena diawali dengan tanda titik '.', pada file '.htaccess' isi perintah sebagai berikut.



The screenshot shows the Visual Studio Code editor with the '.htaccess' file open. The Explorer sidebar on the left shows the project structure, including folders like 'libraries', 'logs', 'models', 'third\_party', 'views', and 'errors', and files like 'contact.php', 'helloview.php', 'index.html', 'profil.php', 'tugas1.php', 'tugas2.php', 'tugas3view.php', 'tugas4view.php', 'variabelview.php', 'welcome\_message...', '.htaccess', 'index.html', 'system', 'user\_guide', '.editorconfig', '.gitignore', and '.htaccess'. The main editor area shows the following content:

```
1 RewriteEngine On
2 RewriteCond %{REQUEST_FILENAME} !-f
3 RewriteCond %{REQUEST_FILENAME} !-d
4 RewriteRule ^(.*)$ index.php/$1 [L]
```

- 1.4. Jika sudah di save, maka hasil akan seperti ini



## Halaman Contact

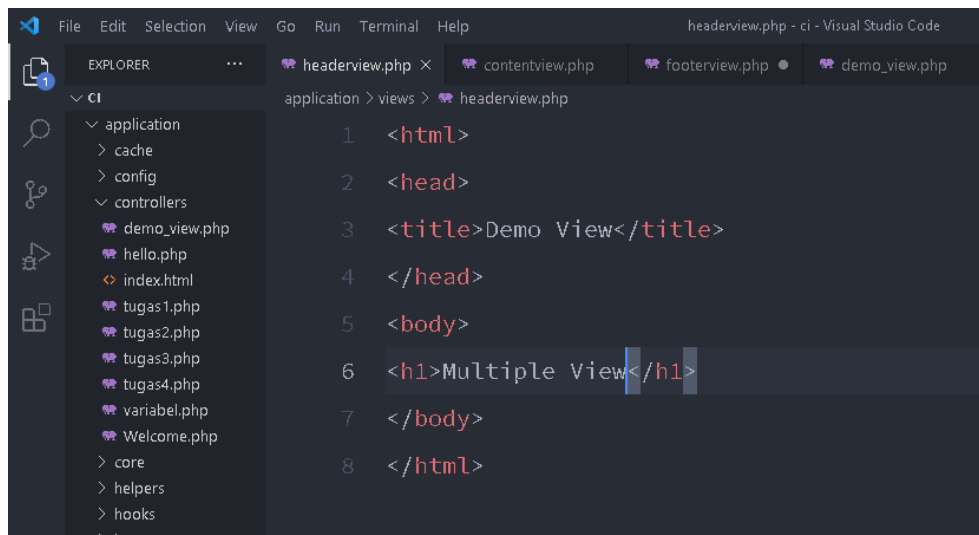
Selamat datang ini adalah halaman contact



2. Cobalah kode dibawah ini dan buat laporannya :

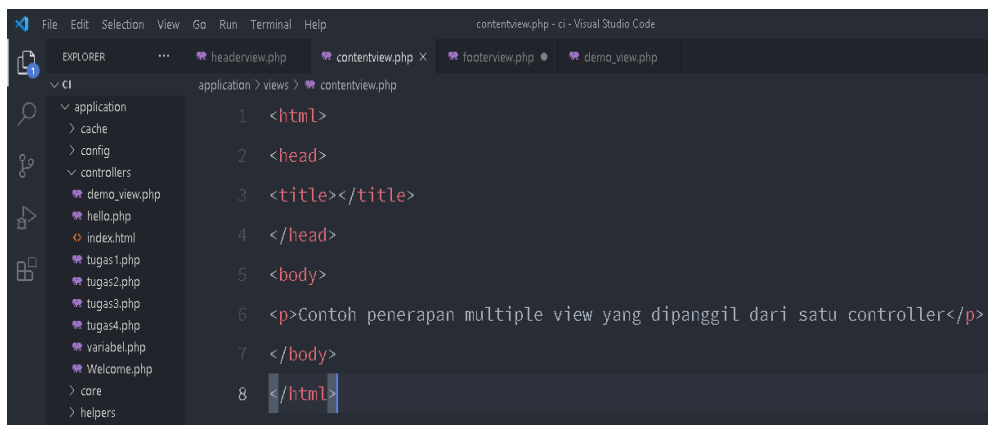
### 2.1. Kode 1 : Load multiple view

#### 2.1.1. Membuat headerview.php



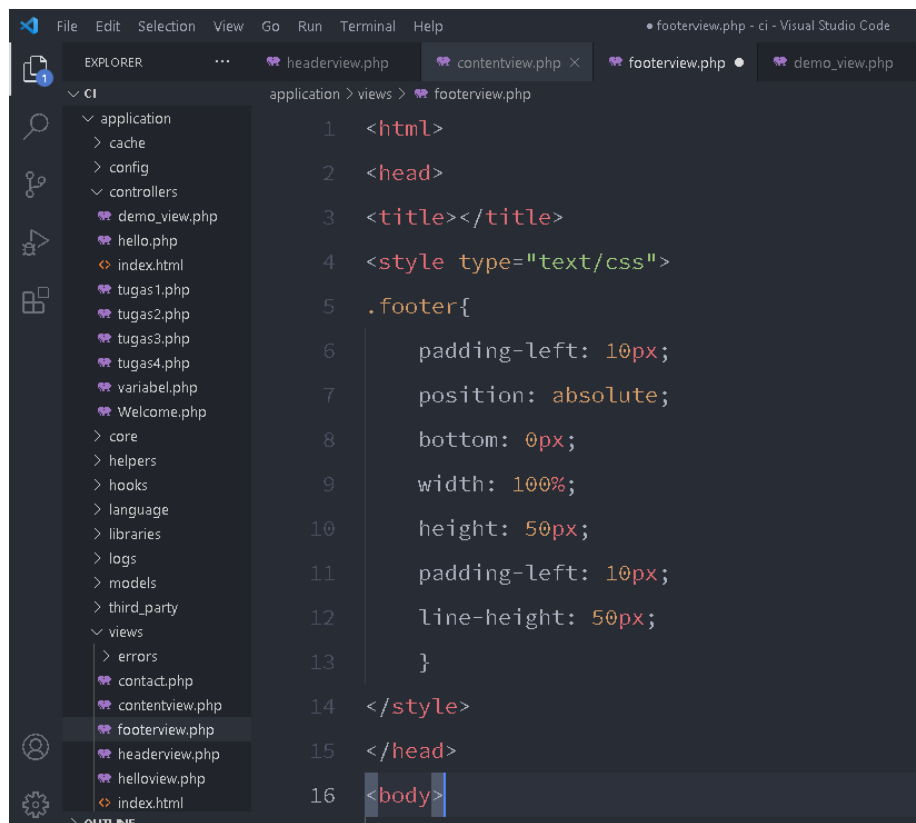
```
1 <html>
2 <head>
3 <title>Demo View</title>
4 </head>
5 <body>
6 <h1>Multiple View</h1>
7 </body>
8 </html>
```

#### 2.1.2. Membuat contentview.php



```
1 <html>
2 <head>
3 <title></title>
4 </head>
5 <body>
6 <p>Contoh penerapan multiple view yang dipanggil dari satu controller</p>
7 </body>
8 </html>
```

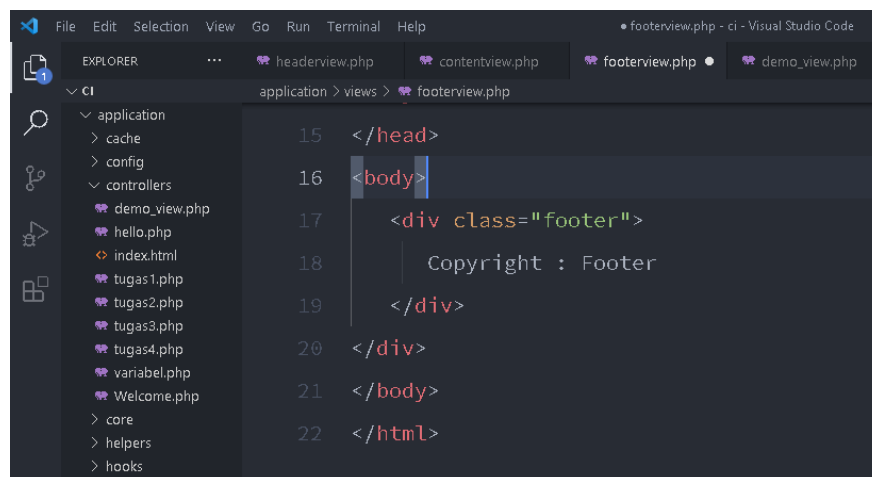
### 2.1.3. Membuat footerview.php



```
File Edit Selection View Go Run Terminal Help
• footerview.php - ci - Visual Studio Code

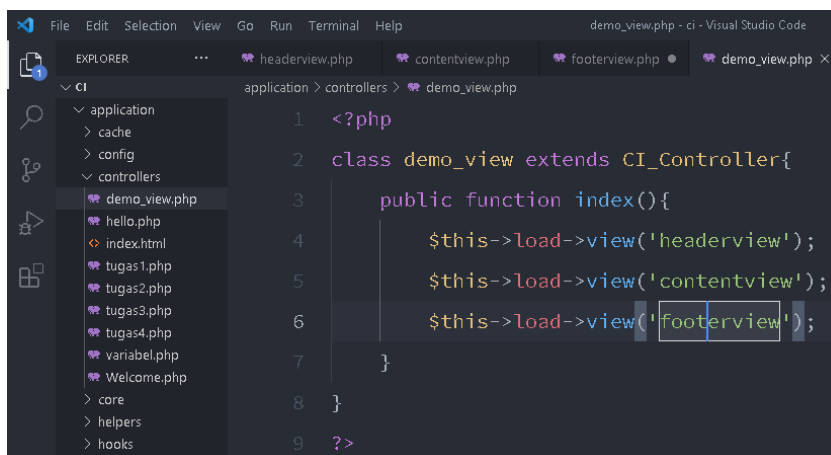
EXPLORER
CI
  application
    > cache
    > config
    > controllers
      demo_view.php
      hello.php
      index.html
      tugas1.php
      tugas2.php
      tugas3.php
      tugas4.php
      variabel.php
      Welcome.php
    > core
    > helpers
    > hooks
    > language
    > libraries
    > logs
    > models
    > third_party
    > views
      errors
      contact.php
      contentview.php
      footerview.php
      headerview.php
      helloview.php
      index.html

1  <html>
2  <head>
3  <title></title>
4  <style type="text/css">
5  .footer{
6      padding-left: 10px;
7      position: absolute;
8      bottom: 0px;
9      width: 100%;
10     height: 50px;
11     padding-left: 10px;
12     line-height: 50px;
13 }
14 </style>
15 </head>
16 <body>
```



```
15 </head>
16 <body>
17     <div class="footer">
18         Copyright : Footer
19     </div>
20 </div>
21 </body>
22 </html>
```

### 2.1.4. Membuat sebuah controller dengan nama demo\_view.php

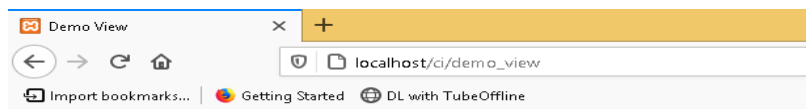


```
File Edit Selection View Go Run Terminal Help
demo_view.php - ci - Visual Studio Code

EXPLORER
CI
  application
    > cache
    > config
    > controllers
      demo_view.php
      hello.php
      index.html
      tugas1.php
      tugas2.php
      tugas3.php
      tugas4.php
      variabel.php
      Welcome.php
    > core
    > helpers
    > hooks
    > language
    > libraries
    > logs
    > models
    > third_party
    > views

1  <?php
2  class demo_view extends CI_Controller{
3      public function index(){
4          $this->load->view('headerview');
5          $this->load->view('contentview');
6          $this->load->view('footerview');
7      }
8  }
9  ?>
```

### 2.1.5. Hasil



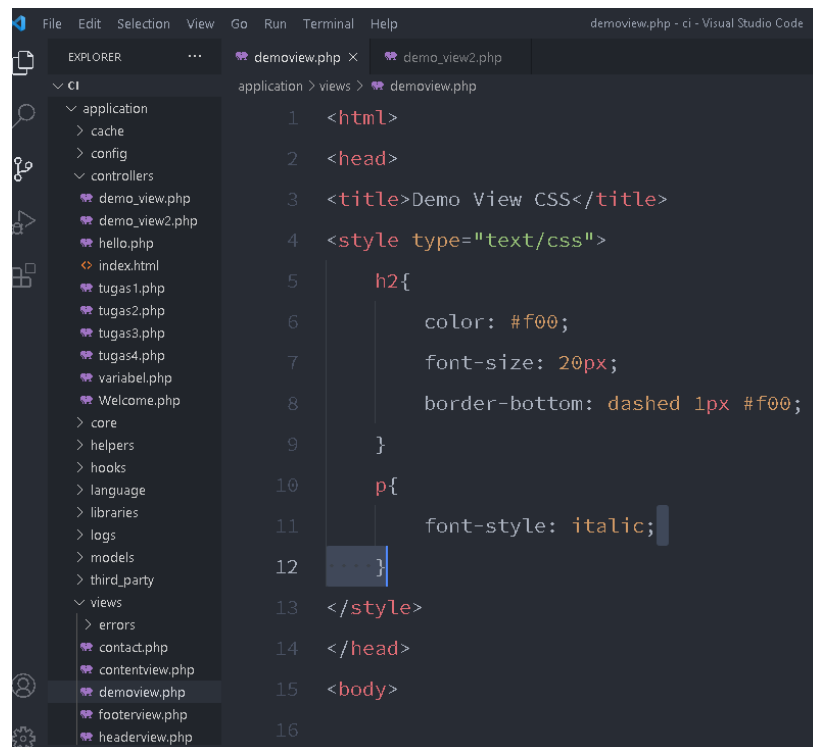
## Multiple View

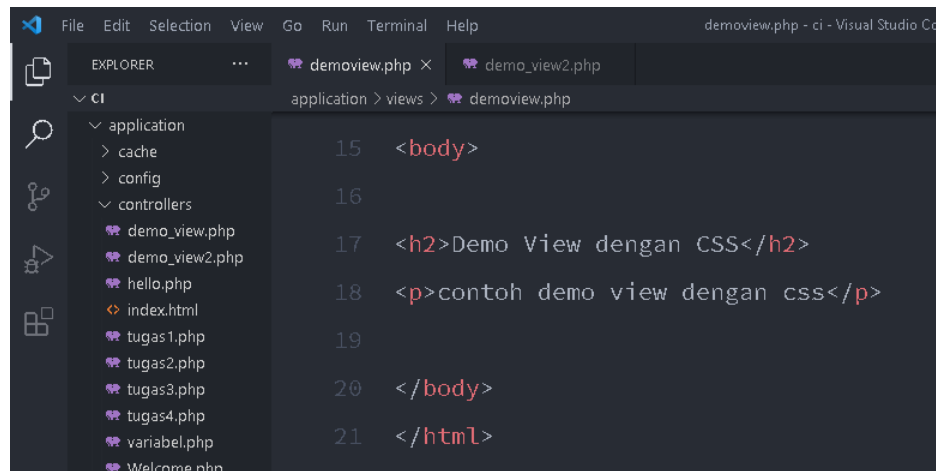
Contoh penerapan multiple view yang dipanggil dari satu controller

Copyright : Footer

## 2.2. Kode 2 : Menyisipkan CSS ke view 1

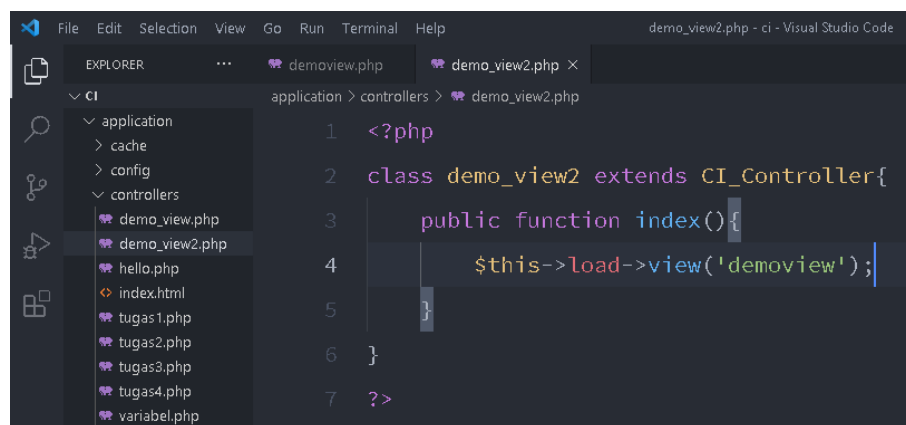
### 2.2.1. Membuat demoview di file direktori view





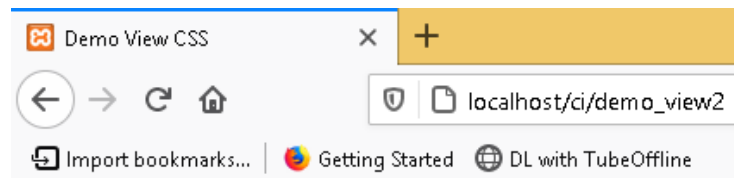
```
15 <body>
16
17 <h2>Demo View dengan CSS</h2>
18 <p>contoh demo view dengan css</p>
19
20 </body>
21 </html>
```

### 2.2.2. Membuat controller dengan nama demo\_view2



```
1 <?php
2 class demo_view2 extends CI_Controller{
3     public function index(){
4         $this->load->view('demoview');
5     }
6 }
7 ?>
```

### 2.2.3. Hasil

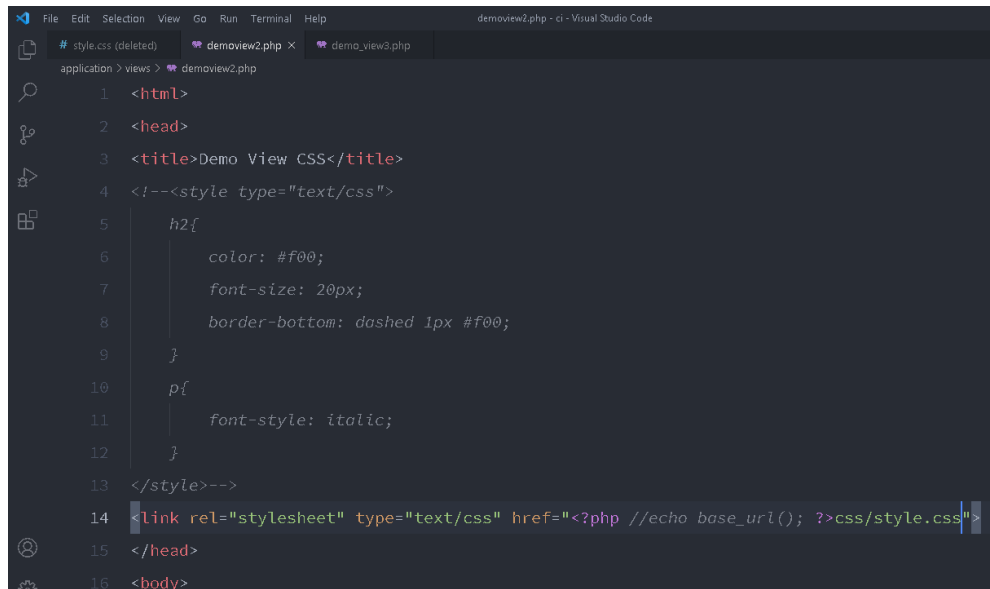


## Demo View dengan CSS

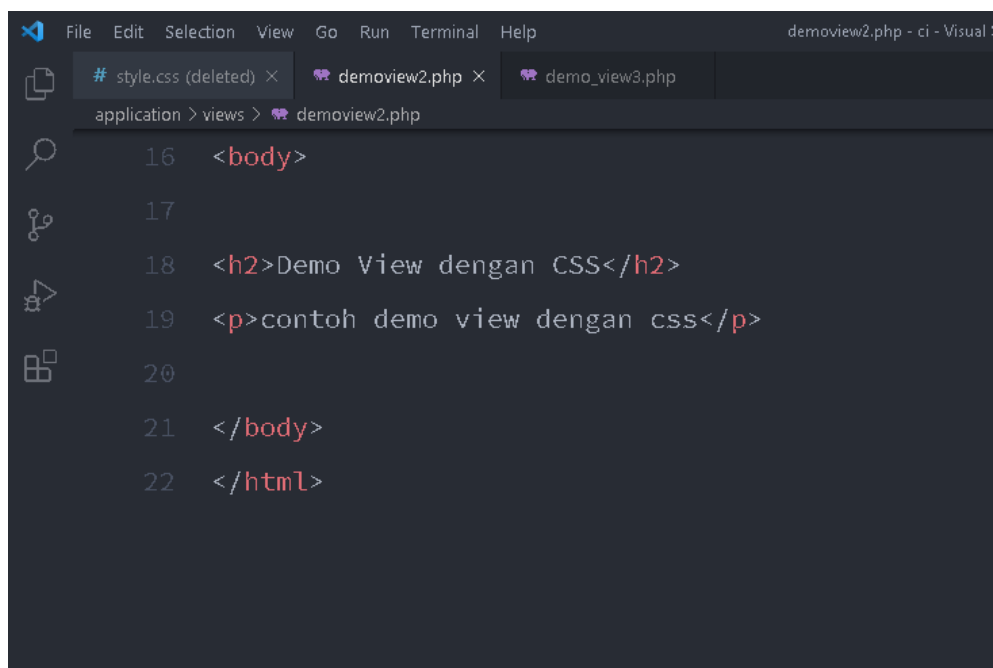
*contoh demo view dengan css*

## 2.3. Kode 3 : Menyisipkan CSS ke View 2

### 2.3.1. Membuat demoview di file direktori view

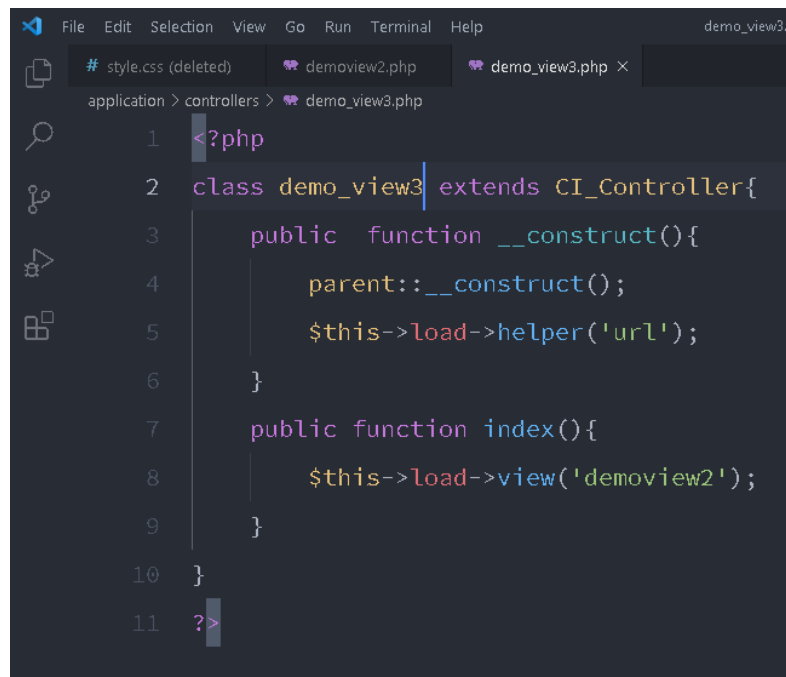


```
1 <html>
2 <head>
3 <title>Demo View CSS</title>
4 <!--<style type="text/css">
5     h2{
6         color: #f00;
7         font-size: 20px;
8         border-bottom: dashed 1px #f00;
9     }
10    p{
11        font-style: italic;
12    }
13 </style>-->
14 <link rel="stylesheet" type="text/css" href="<?php //echo base_url(); ?>css/style.css">
15 </head>
16 <body>
```



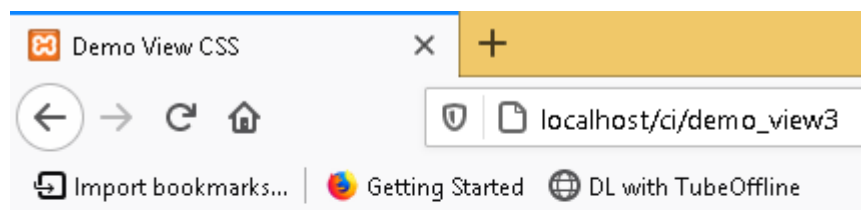
```
16 <body>
17
18 <h2>Demo View dengan CSS</h2>
19 <p>contoh demo view dengan css</p>
20
21 </body>
22 </html>
```

### 2.3.2. Membuat controller demo\_view



```
1 <?php
2 class demo_view3 extends CI_Controller{
3     public function __construct(){
4         parent::__construct();
5         $this->load->helper('url');
6     }
7     public function index(){
8         $this->load->view('demoview2');
9     }
10 }
11 ?>
```

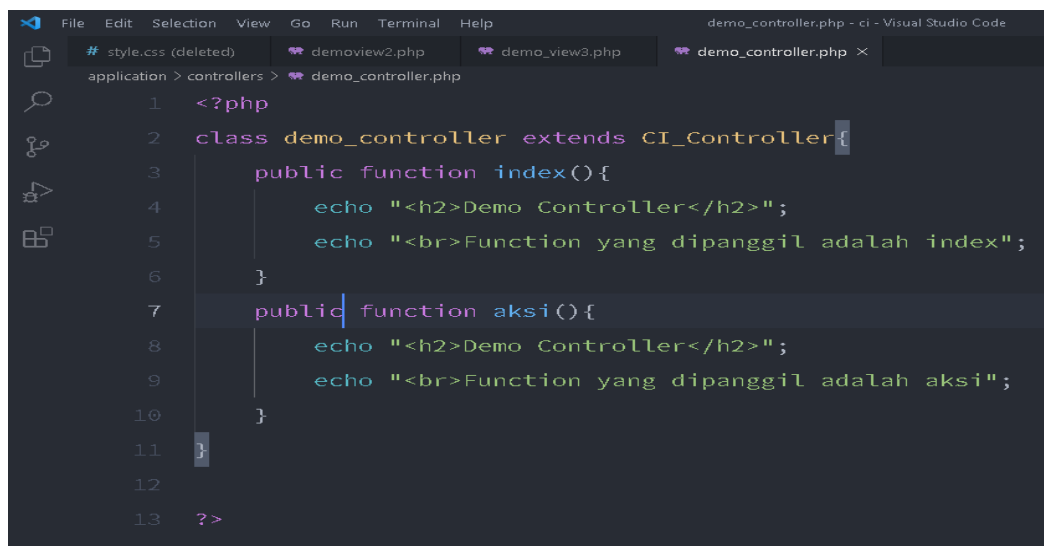
### 2.3.3. Hasil



## Demo View dengan CSS

*contoh demo view dengan css*

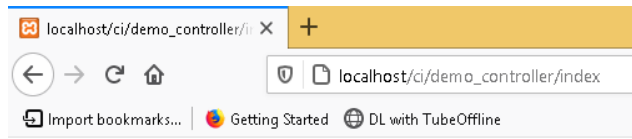
### 2.4. Controller : Demo\_controller.php



```
1 <?php
2 class demo_controller extends CI_Controller{
3     public function index(){
4         echo "<h2>Demo Controller</h2>";
5         echo "<br>Function yang dipanggil adalah index";
6     }
7     public function aksi(){
8         echo "<h2>Demo Controller</h2>";
9         echo "<br>Function yang dipanggil adalah aksi";
10    }
11 }
12
13 ?>
```

### 2.4.1 Hasil

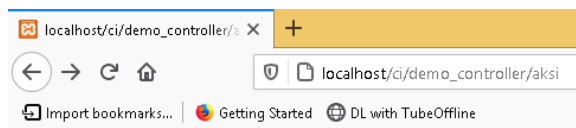
## Fungsi Index



## Demo Controller

Function yang dipanggil adalah index

### Fungsi aksi

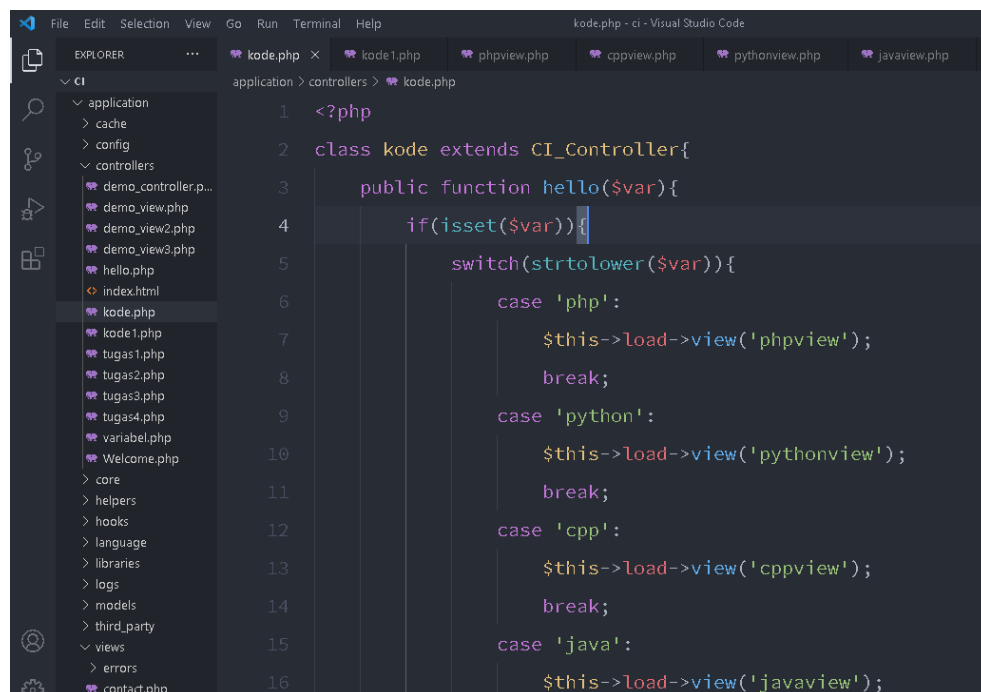


## Demo Controller

Function yang dipanggil adalah aksi

## 2.5. Melewatkan segment URI kedalam metode

### 2.5.1. Membuat controller kode



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the file structure of a project. The file `kode.php` is selected under the `application > controllers` path. The main editor window shows the PHP code for `kode.php`, which includes a `load->view('javaview');` call, a `break;` statement, a `default:` case with an `echo 'Input URL Salah';` statement, and an `else{}` block with another `echo 'Input URL Salah';` statement.

```
16 $this->load->view('javaview');
17
18 break;
19
20 default:
21     echo 'Input URL Salah';
22 }
23 }else{
24     echo 'Input URL Salah';
25 }
26 }
27 ?>
```

## 2.5.2. Membuat phpview di file views

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the file structure. The file `phpview.php` is selected under the `application > views` path. The main editor window shows the HTML code for `phpview.php`, which includes a `<html>` tag, a `<head>` section with a `<title>Hello View</title>` tag, and a `<body>` section with a `<h2>Hello World! PHP</h2>` tag.

```
1 <html>
2 <head>
3 <title>Hello View</title>
4 </head>
5 <body>
6 <h2>Hello World! PHP</h2>
7 </body>
8 </html>
```

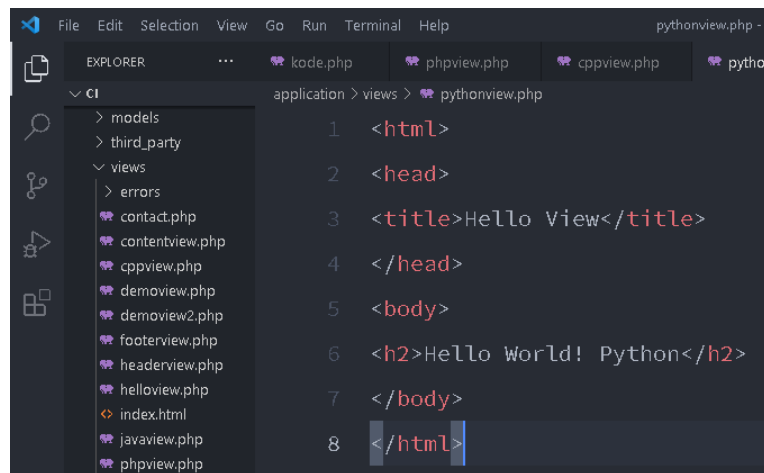
## 2.5.3. Membuat cppview di file views

The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left displaying the file structure. The file `cppview.php` is selected under the `application > views` path. The main editor window shows the HTML code for `cppview.php`, which is identical to `phpview.php` but with the `<h2>Hello World! CPP</h2>` tag in the body section.

```
1 <html>
2 <head>
3 <title>Hello View</title>
4 </head>
5 <body>
6 <h2>Hello World! CPP</h2>
7 </body>
8 </html>
```



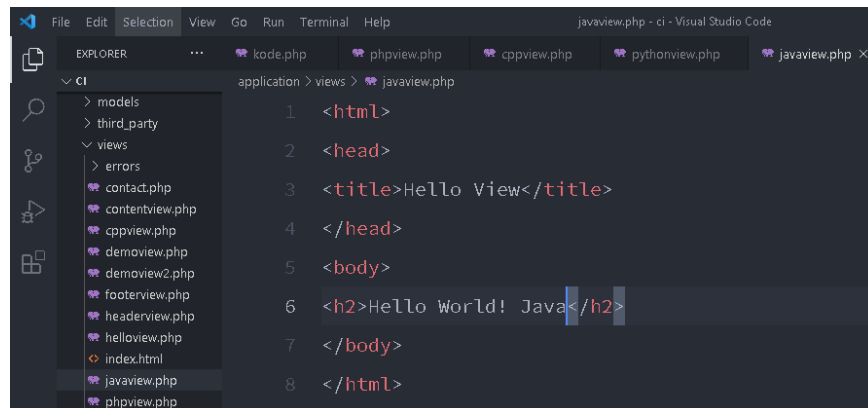
#### 2.5.4. Membuat pythonview di file views



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer shows a project structure with folders 'models', 'third\_party', and 'views'. The 'views' folder is expanded, showing files like 'contact.php', 'contentview.php', 'cppview.php', 'demoview.php', 'demoview2.php', 'footerview.php', 'headerview.php', 'helloview.php', 'index.html', 'javaview.php', and 'phpview.php'. The main editor area shows the content of 'pythonview.php' with the following HTML code:

```
1 <html>
2 <head>
3 <title>Hello View</title>
4 </head>
5 <body>
6 <h2>Hello World! Python</h2>
7 </body>
8 </html>
```

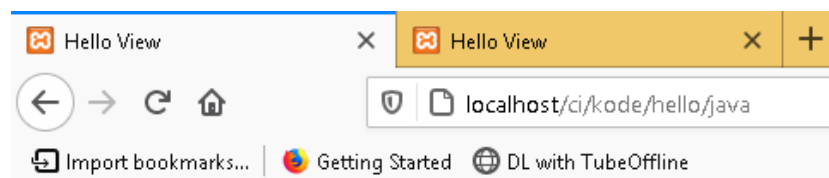
#### 2.5.5. Membuat javaview di file views



The screenshot shows the Visual Studio Code editor with the Explorer sidebar on the left. The Explorer shows the same project structure as the previous screenshot. The main editor area shows the content of 'javaview.php' with the following HTML code:

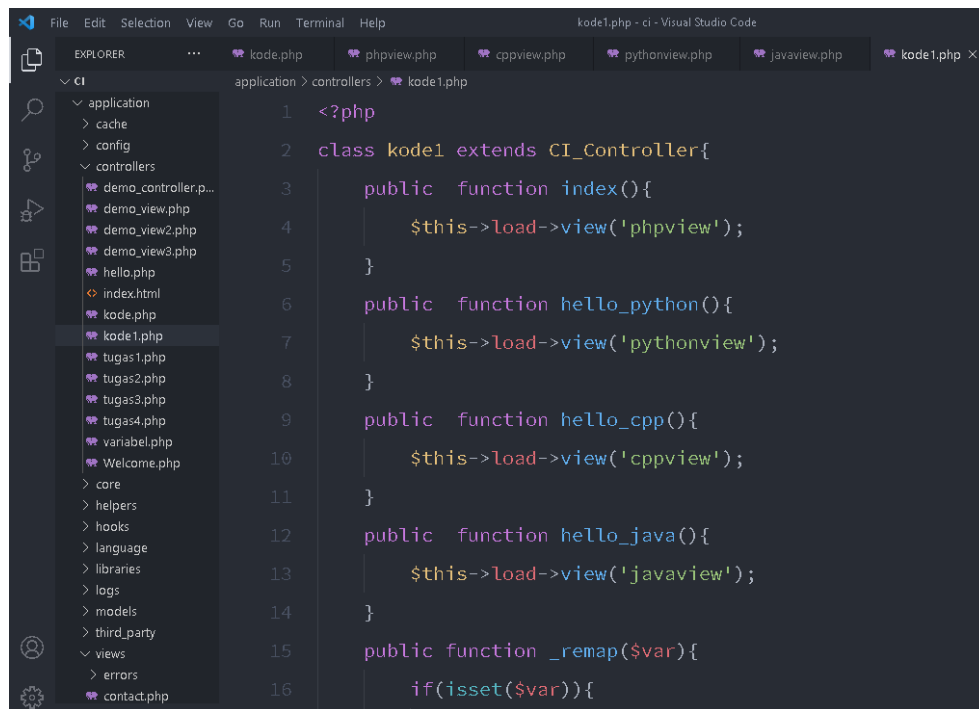
```
1 <html>
2 <head>
3 <title>Hello View</title>
4 </head>
5 <body>
6 <h2>Hello World! Java</h2>
7 </body>
8 </html>
```

#### 2.5.6. Hasil

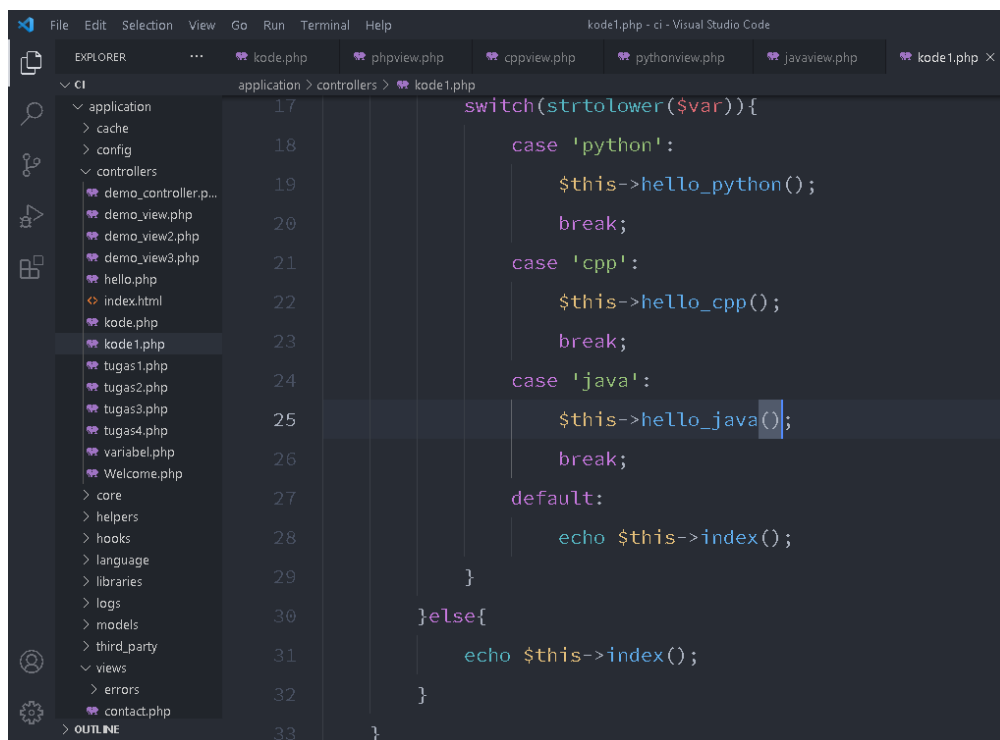


# Hello World! Java

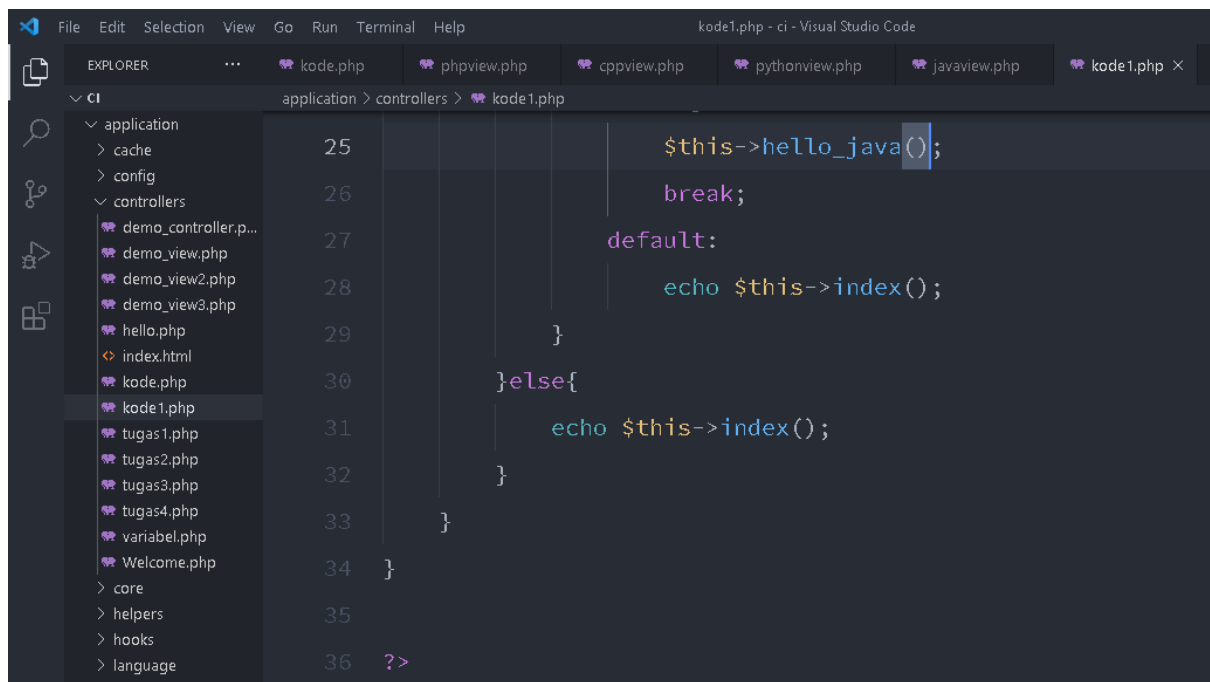
## 2.6. Memetakan nama metode yang akan dipanggil



```
1 <?php
2 class kode1 extends CI_Controller{
3     public function index(){
4         $this->load->view('phpview');
5     }
6     public function hello_python(){
7         $this->load->view('pythonview');
8     }
9     public function hello_cpp(){
10        $this->load->view('cppview');
11    }
12    public function hello_java(){
13        $this->load->view('javaview');
14    }
15    public function _remap($var){
16        if(isset($var)){
```

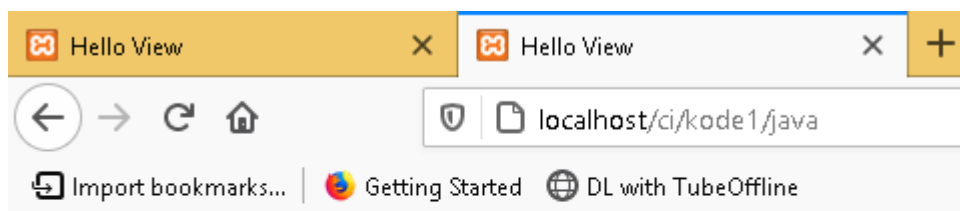


```
17 switch(strtolower($var)){
18     case 'python':
19         $this->hello_python();
20         break;
21     case 'cpp':
22         $this->hello_cpp();
23         break;
24     case 'java':
25         $this->hello_java();
26         break;
27     default:
28         echo $this->index();
29 }
30 }else{
31     echo $this->index();
32 }
33 }
```



```
25     $this->hello_java();
26     break;
27     default:
28         echo $this->index();
29     }
30 }else{
31     echo $this->index();
32 }
33 }
34 }
35
36 ?>
```

- **Hasil**



# Hello World! Java

### **BAB III**

### **KESIMPULAN**

1. Router pada framework codeigniter, memiliki tugas untuk menentukan controller serta method/fungsi yang akan dijalankan ketika pengguna aplikasi mengakses alamat/url tertentu.
2. Menyisipkan css terdapat 2 cara yaitu pertama, dapat langsung dituliskan di dalam HTML di bagian head dengan member kode 'style = text/css', dan kedua dapat menggunakan cara tag link dengan kode '<link rel="stylesheet" type="text/css" href="<?php //echo base\_url(); ?>css/style.css">'. Jika kita menggunakan cara kedua kita harus merubah kode di kontroler.
3. URI atau singkatan dari Uniform Resource Identifier, memiliki fungsi membantu kamu untuk mengambil data melalui url codeIgniter. Perlu kamu ketahui dulu segment 1 pada codeIgniter adalah class atau controller yang sedang dijalankan. Segment 2 adalah nama method atau function, sedangkan segment 3 adalah data berbentuk id yang dikirim.