

LAPORAN PEMOGRAMAN WEB FRAMEWORK
“Konsep Router pada Code Igniter”



DISUSUN OLEH :

Ratih Cahyani Apriliya (E31191660 / Golongan C)

MANAJEMEN INFORMATIKA
TEKNOLOGI INFORMASI
POLITEKNIK NEGERI JEMBER

BAB I

PENDAHULUAN

1.1 Latar Belakang

PHP adalah sebuah bahasa pemrograman web yang populer, tangguh dan dapat diperoleh secara gratis. Bahasa pemrograman ini tergolong mudah untuk dipelajari. Untuk mempermudah dan mempercepat pengembangan aplikasi dengan PHP, banyak bermunculan framework PHP. Framework adalah suatu struktur yang nyata atau konseptual dimaksudkan untuk melayani sebagai dukungan atau panduan untuk membangun sesuatu yang memperluas struktur menjadi sesuatu yang bermanfaat. Pada Perangkat komputer, Framework biasanya berstruktur berlapis - lapis yang menunjukkan jenis program yang dapat dibangun dan program yang saling berhubungan. Beberapa framework sistem komputer juga mencakup program-program yang sebenarnya, menentukan antarmuka pemrograman, atau menawarkan tool pemrograman untuk menggunakan kerangka kerja. Sebuah kerangka mungkin berupa satu set fungsi dalam sistem dan bagaimana mereka saling berhubungan, lapisan sistem operasi, lapisan subsistem aplikasi, bagaimana komunikasi harus distandardisasi pada beberapa tingkat jaringan, dan sebagainya. Sebuah framework umumnya lebih komprehensif daripada protokol dan lebih preskriptif daripada struktur. Framework secara sederhana dapat diartikan kumpulan dari fungsi atau prosedur dan class-class untuk tujuan tertentu yang sudah siap digunakan sehingga bisa lebih mempermudah dan mempercepat pekerjaan seorang programmer, tanpa harus membuat fungsi atau class dari awal.

Framework adalah sebuah kerangka kerja yang digunakan untuk mempermudah para developer software dalam membuat dan mengembangkan aplikasi. Secara sederhana, framework disebut sebagai kerangka kerja. Framework berisikan perintah dan fungsi dasar yang umum digunakan untuk membangun sebuah software aplikasi sehingga diharapkan aplikasi dapat dibangun dengan lebih cepat serta tersusun dan terstruktur dengan cukup rapi. Framework juga bisa diartikan sebagai komponen – komponen pemrograman yang sudah jadi dan siap

untuk digunakan kapan saja, sehingga pengembang aplikasi tidak perlu lagi membuat script yang sama untuk tugas – tugas yang sama

1.2 Rumusan Masalah

1.2.1 Bagaimana konsep dasar Router pada CI?

1.2.2 Bagaimana membuat beberapa router pada CI?

1.3 Tujuan

1.3.1 Mahasiswa mampu memahami konsep dasar Router pada CI.

1.3.2 Mahasiswa mampu membuat beberapa router pada CI.

BAB II

DASAR TEORI

2.1 Router di Codeigniter

Router 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/ci> maka 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

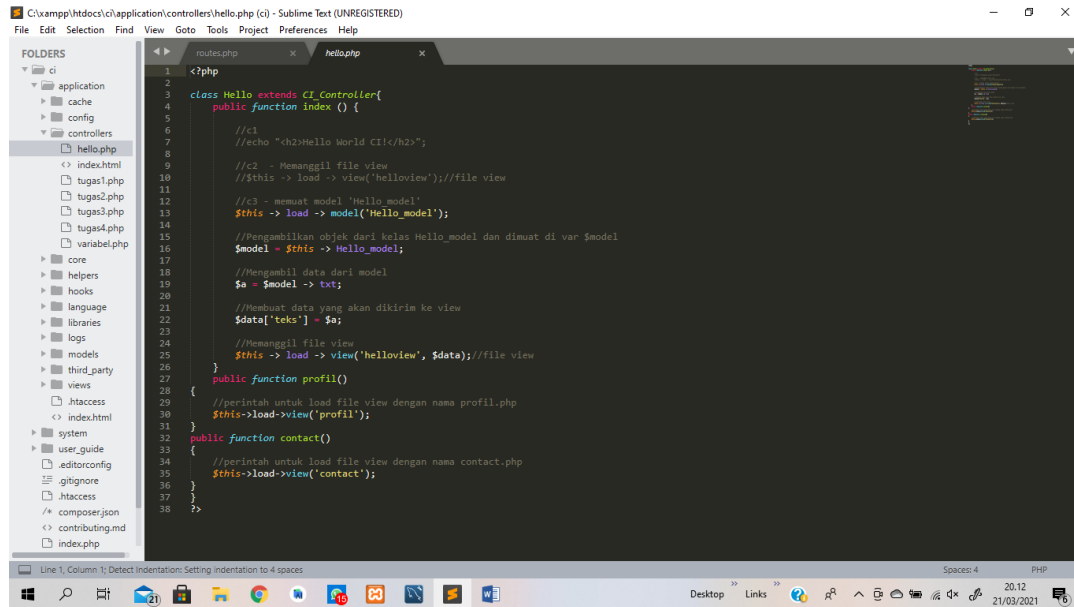
BAB III

HASIL DAN PEMBAHASAN

3.1 Kegiatan Praktikum

3.1.1 Membuat Beberapa Router

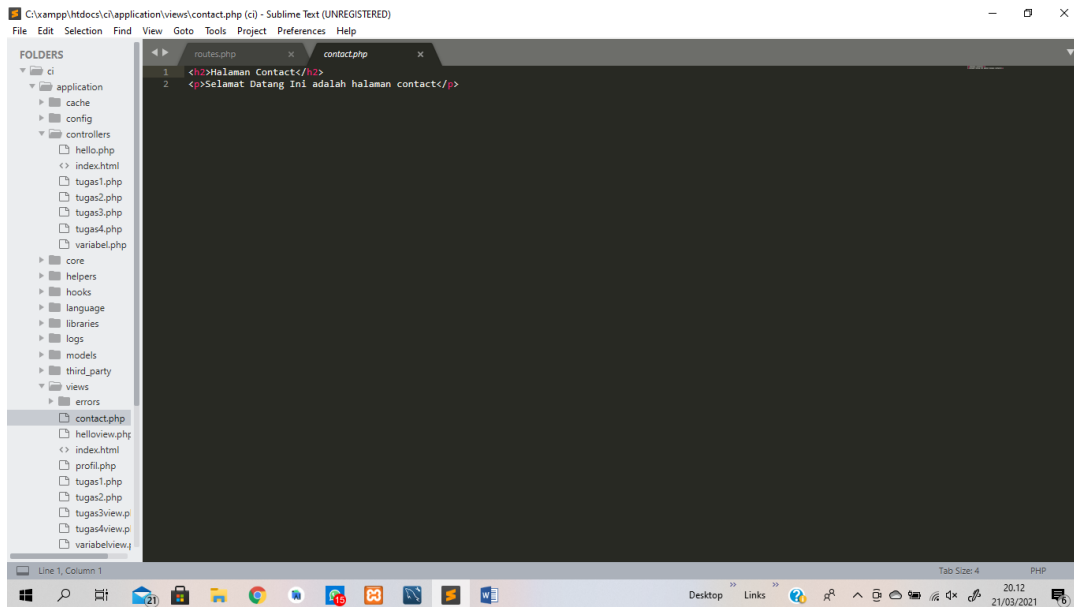
Membuat 2 function didalam controller Welcome dibawah function index.



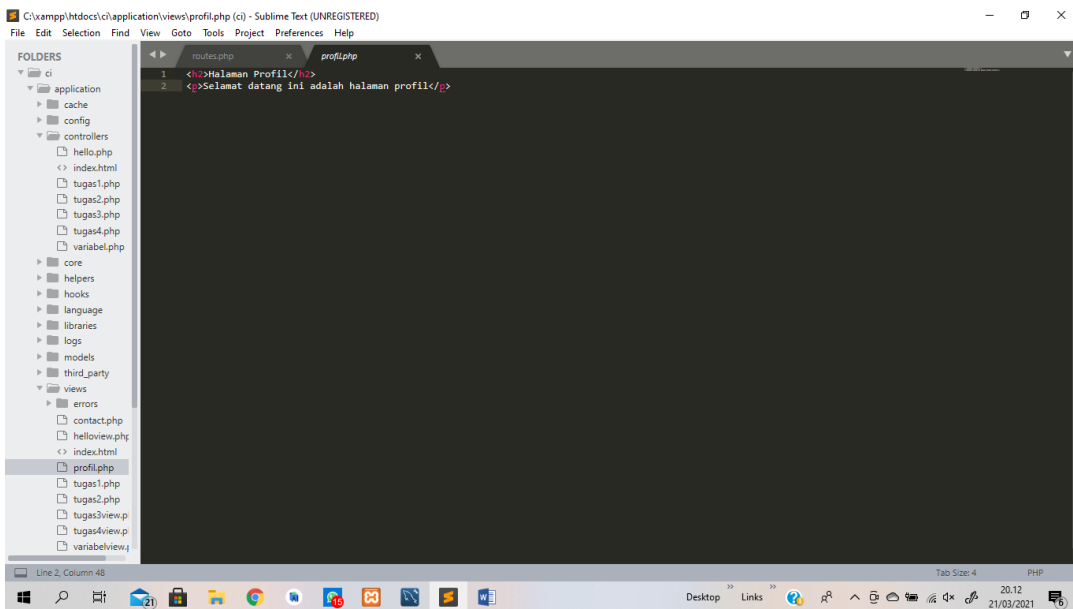
```
1 <?php
2
3 class Hello extends CI_Controller{
4     public function index () {
5
6         //c1
7         //echo "ch2>Hello World CI</h2>";
8
9         //c2 - Memanggil file view
10        //$this->load->view('helloworld'); //file view
11
12        //c3 - memuat model 'Hello_model'
13        //$this->load->model('Hello_model');
14
15        //Pengambilan objek dari kelas Hello_model dan dimuat di var $model
16        $model = $this->Hello_model;
17
18        //Membuat data dari model
19        $a = $model->txt;
20
21        //Membuat data yang akan dikirim ke view
22        $data['teks'] = $a;
23
24        //Memanggil file view
25        //$this->load->view('helloworld', $data); //file view
26    }
27    public function profil()
28    {
29        //perintah untuk load file view dengan nama profil.php
30        //$this->load->view('profil');
31    }
32    public function contact()
33    {
34        //perintah untuk load file view dengan nama contact.php
35        //$this->load->view('contact');
36    }
37    }
38    ?>
```

Berikutnya membuat 2 file view, dengan nama profil.php, dan contact.php didalam direktori application/views.

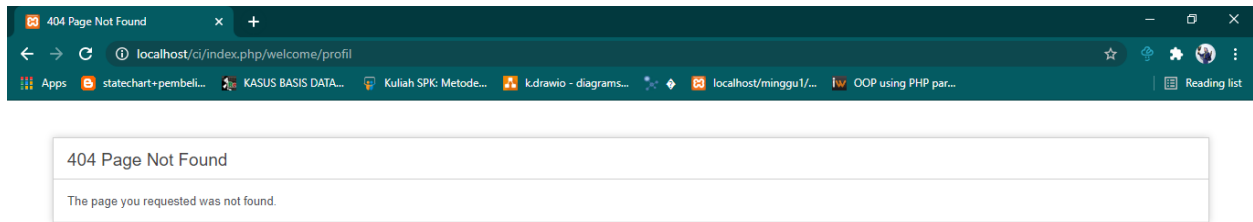
contact.php



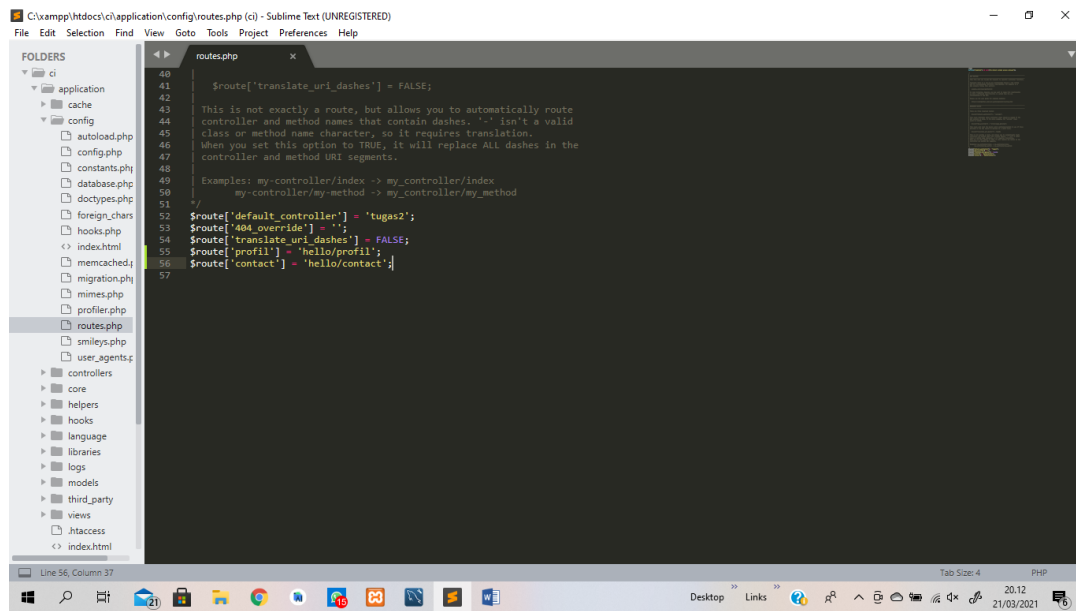
profil.php



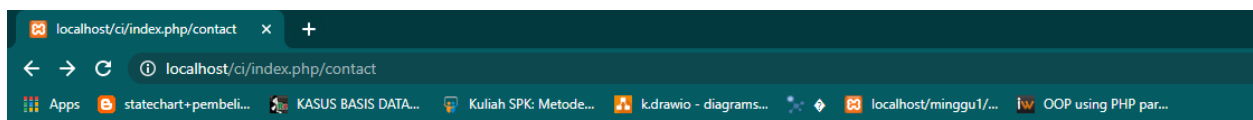
Halaman akan tampil, karena memang kita langsung mengakses function dalam controller welcome dibagian URL. Tetapi apakah bisa kita mengaksesnya dengan alamat :
localhost/ci/index.php/contact dan localhost/ci/index.php/profil



maka akan muncul tampilan 404 not found. Hal ini dikarenakan dibagian router belum diset, sehingga perlu ditambahkan perintah dibagian di router.



Output setelah menambahkan perintah dibagian router :



Halaman Contact

Selamat Datang Ini adalah halaman contact

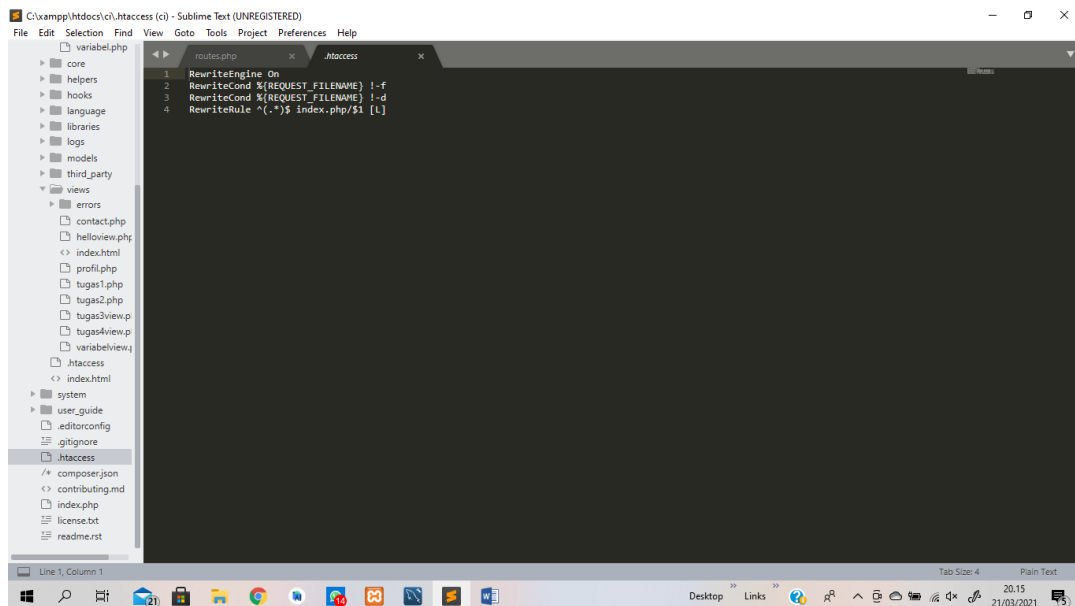
Kita tidak harus menambahkan route di file routers.php, setiap kali anda membuat route baru, karena Codeigniter otomatis mendeteksi route berdasarkan nama controller dan function/method yang dibuat.

3.2 Tugas / Soal

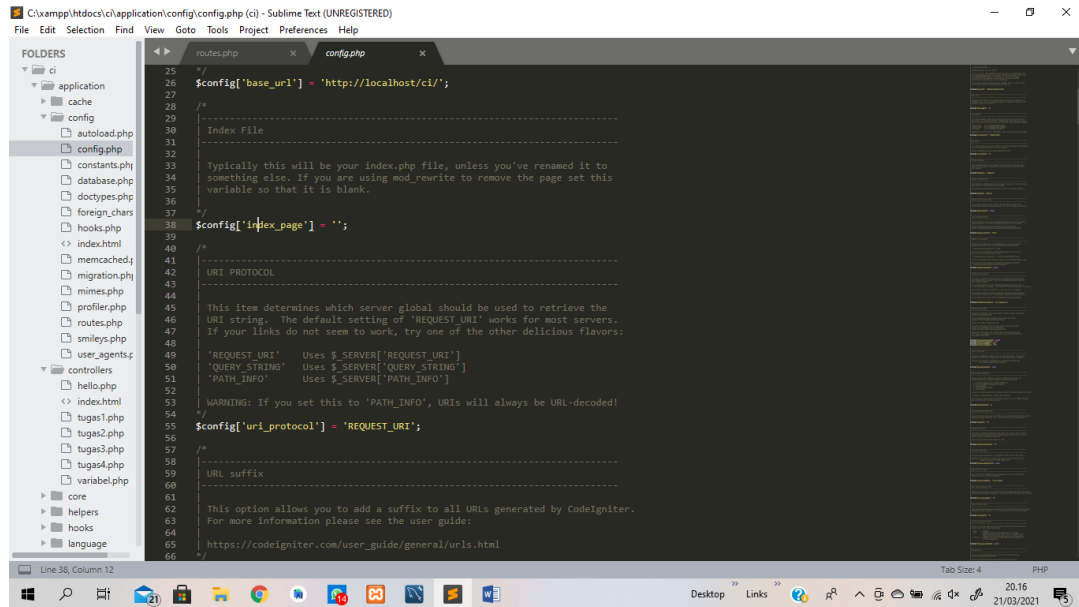
3.2.1 Untuk mengakses file pada CI normalnya link adalah :

localhost/nama_folder/index.php/controller. Rubahlah agar aksesnya menjadi localhost/ci/controller.

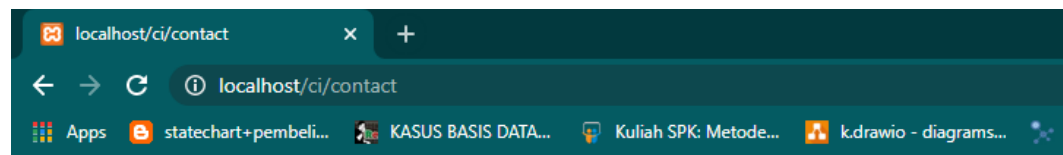
Secara default, file index.php akan dimasukkan ke dalam URL kita akan tetapi kita dapat dengan mudah menghapus file ini menggunakan file .htaccess dengan beberapa aturan sederhana. Berikut adalah contoh file semacam itu, menggunakan metode "negatif" di mana semuanya dialihkan kecuali item yang ditentukan :



Ketika kita telah mengganti namanya menjadi sesuatu yang lain atau menggunakan mod_rewrite untuk menghapus halaman, Selanjutnya kita atur `$config['index_page'] = 'index.php';` menjadi kosong `$config['index_page'] = '';`



Setelah kita menambahkan file .htaccess maka saat kita menuliskan pada URL untuk menjalankan programnya cukup dengan `http://localhost/ci/contact`



Halaman Contact

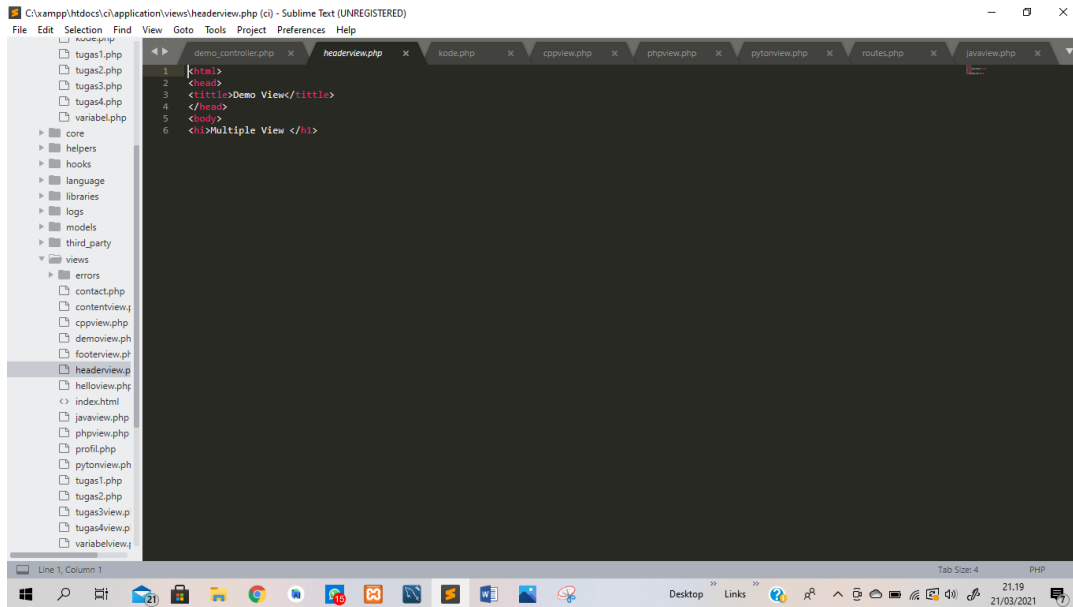
Selamat Datang Ini adalah halaman contact

3.2.2 Cobalah kode dibawah ini :

Kode 1 :

View

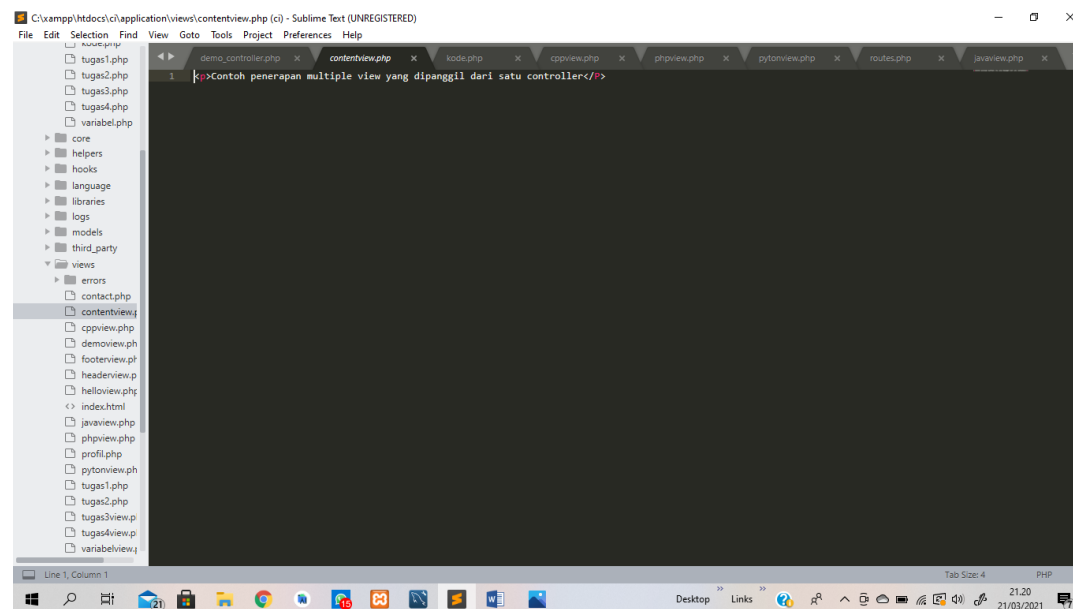
Load multiple view headerview.php



C:\xampp\htdocs\c\application\views\headview.php (c) - Sublime Text (UNREGISTERED)

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

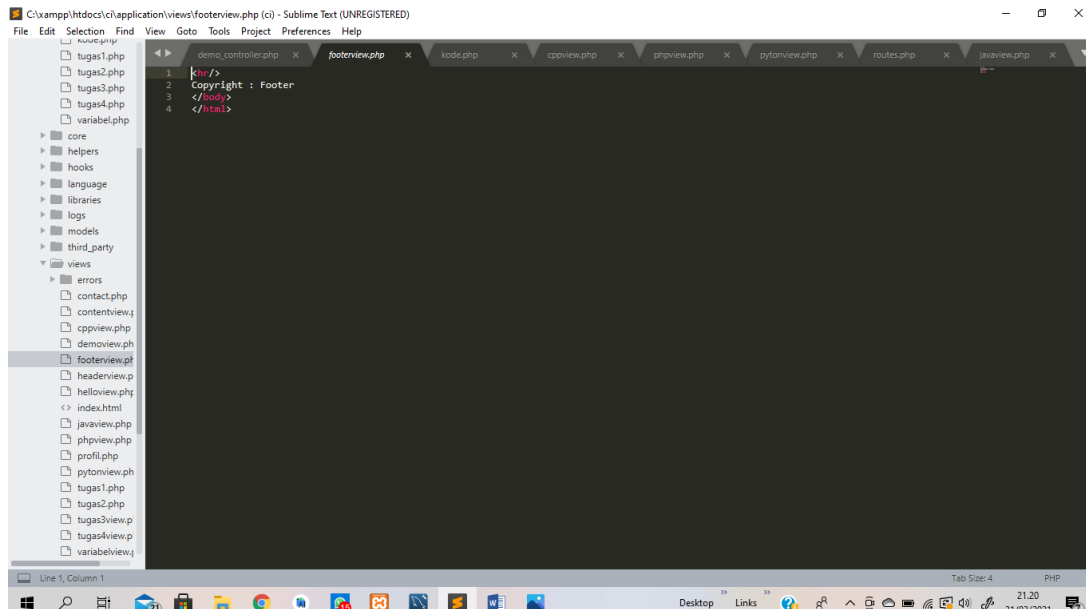
contentview.php



C:\xampp\htdocs\c\application\views\contentview.php (c) - Sublime Text (UNREGISTERED)

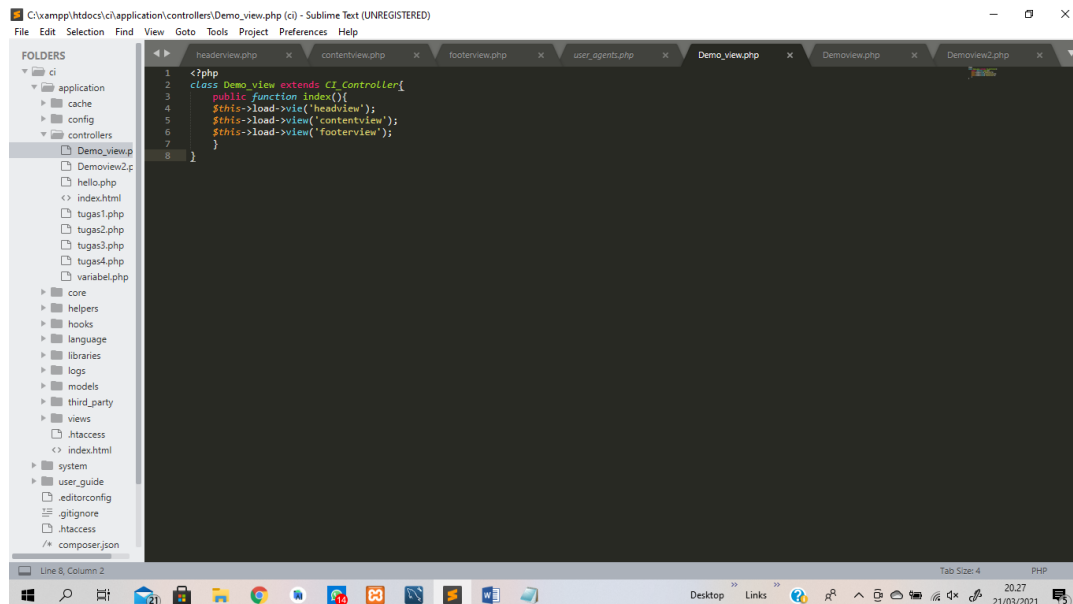
```
1 <?php>Contoh penerapan multiple view yang dipanggil dari satu controller</?php>
```

footerview.php



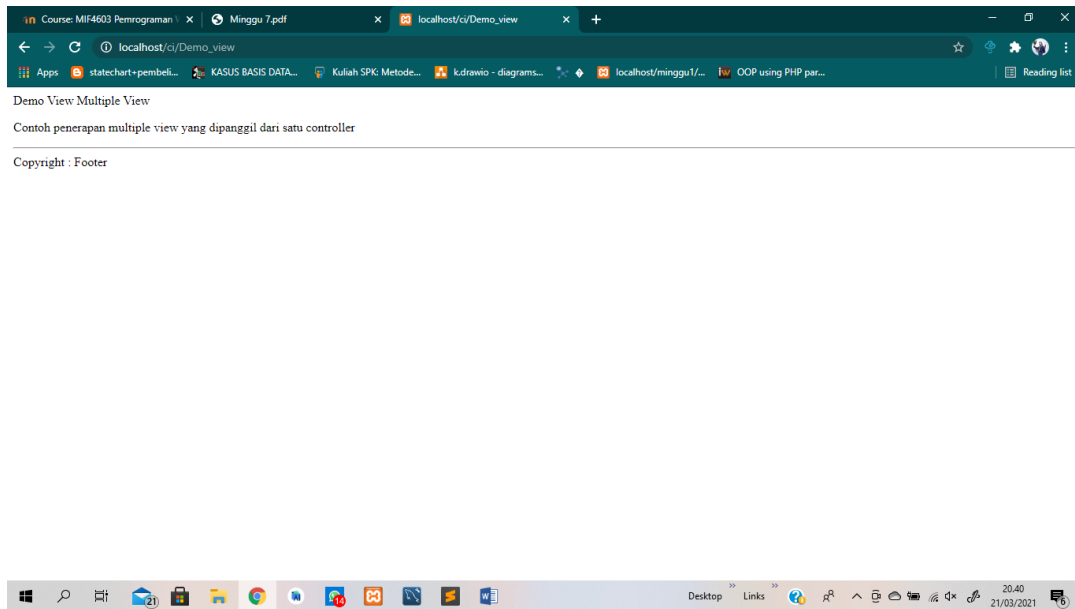
```
1 <!--
2 Copyright : Footer
3 </body>
4 </html>
```

Membuat file Demo_view.php pada controller :



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

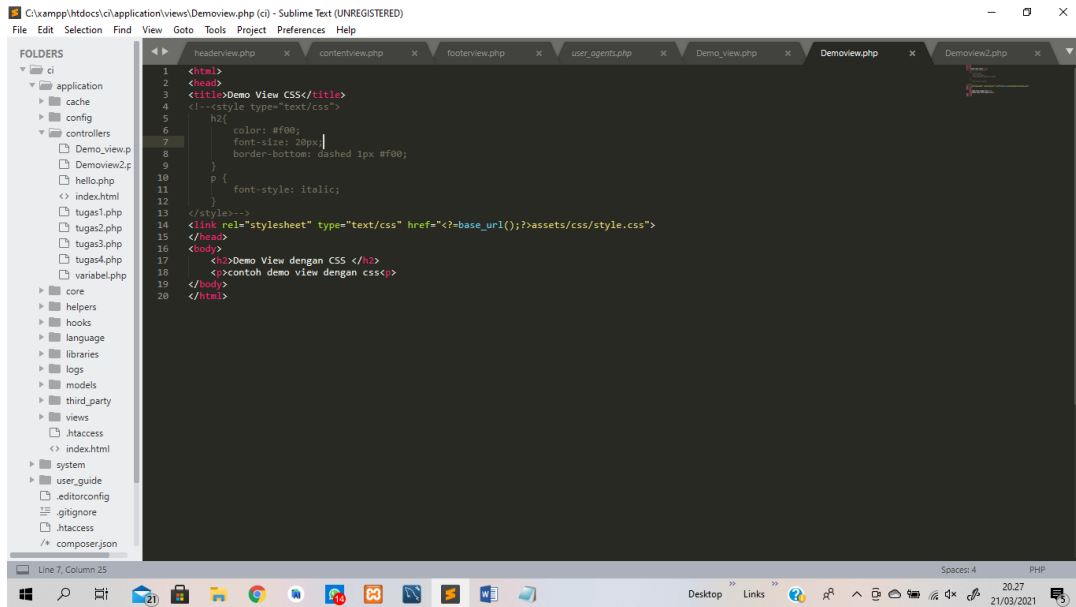
Output :



Sama halnya dengan tugas nomor 1 , setelah menambahkan file .htaccess maka pada saat akan menjalankan program kita hanya mengetikkan pada URL `http://localhost/ci/controller` tidak perlu menggunakan `index.php` lagi karena kita sudah hilangkan pada tugas sebelumnya dan berlaku juga pada tugas berikutnya selama berada pada folder codeigniter yang sama.

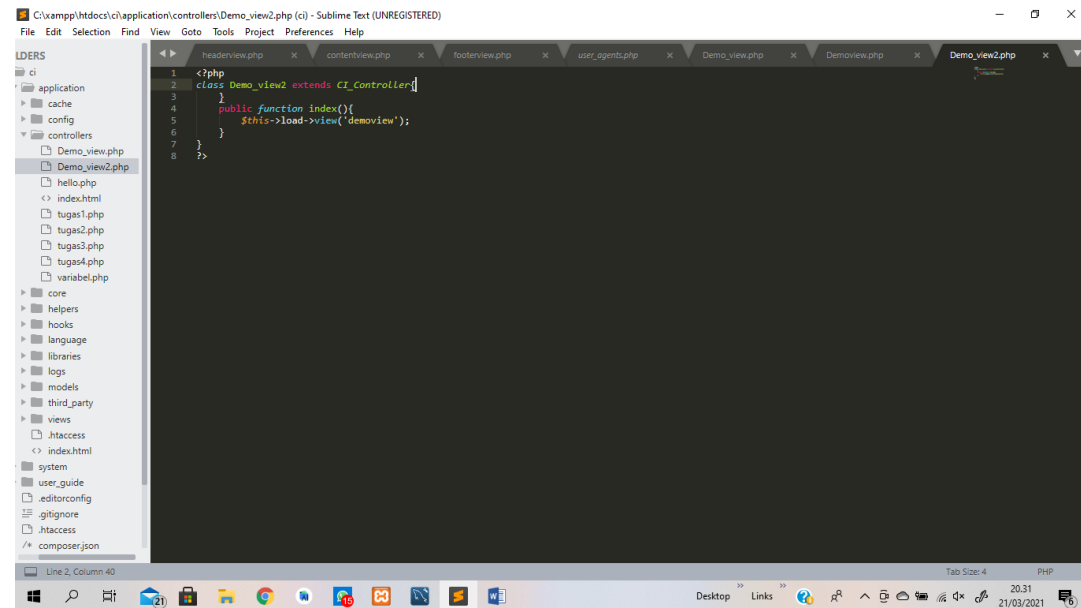
Kode 2 :

Menyisipkan CSS ke View 1
Demoview.php



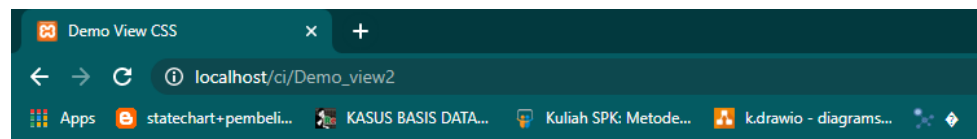
```
1 <html>
2 <head>
3 <title>Demo View CSS</title>
4 <!--style type="text/css">
5 h2{
6     color: #f00;
7     font-size: 20px;
8     border-bottom: dashed 1px #f00;
9 }
10 p {
11     font-style: italic;
12 }
13 </style-->
14 <link rel="stylesheet" type="text/css" href="<?base_url();>assets/css/style.css">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

Demo_view2.php



```
1 <?php
2 class Demo_view2 extends CI_Controller{
3 }
4 public function index(){
5     $this->load->view('demo_view');
6 }
7 }
8 >>
```

Output :

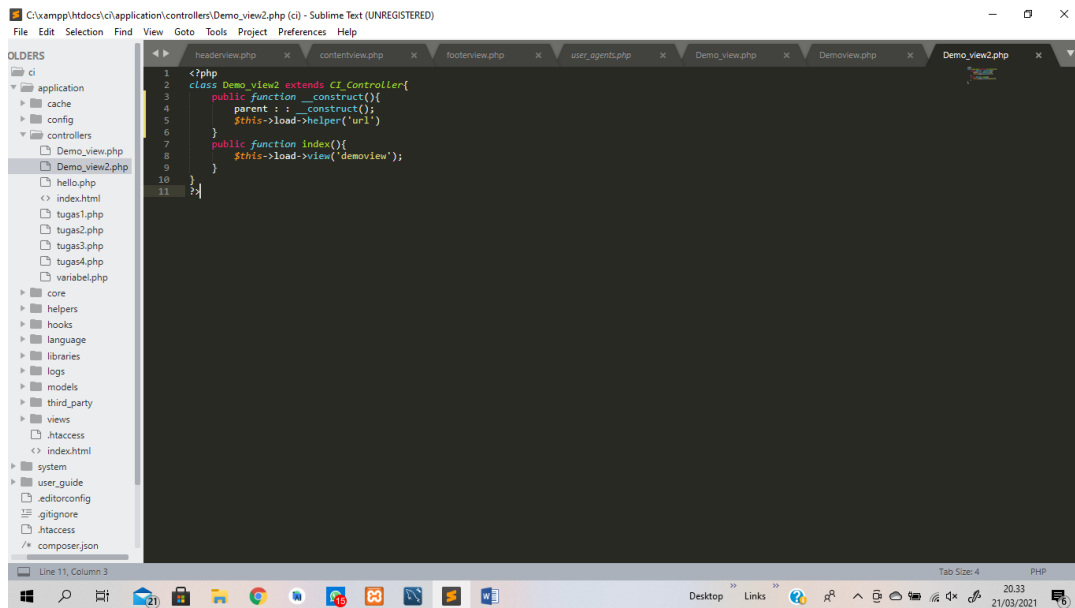


Demo View dengan CSS

contoh demo view dengan css

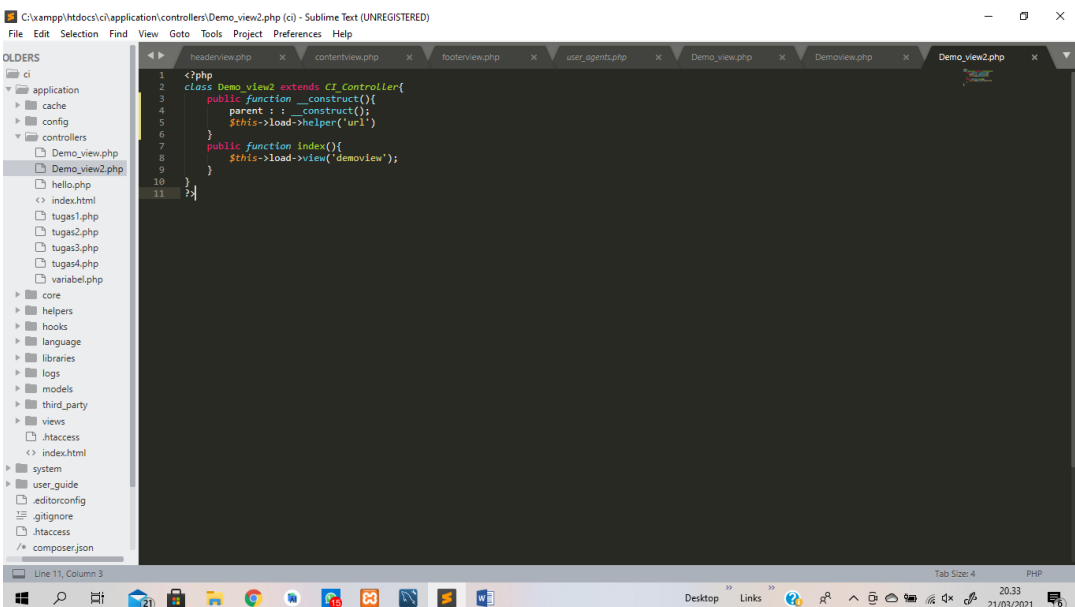
Kode 3 :

Menyisipkan CSS ke View 2 demoview.php



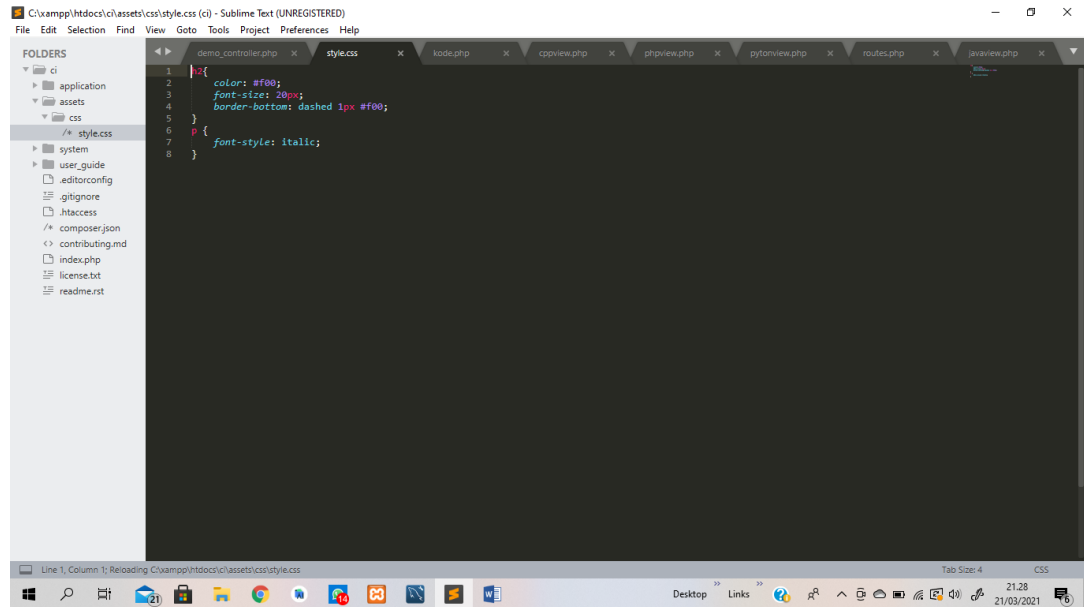
```
1 <?php
2 class Demo_view2 extends CI_Controller{
3     public function __construct(){
4         parent::__construct();
5         $this->load->helper('url');
6     }
7     public function index(){
8         $this->load->view('demoview');
9     }
10 }
11 ?>
```

Menambahkan function `__construct` di `Demo_view2.php` untuk menangkap perintah pada `Demoview.php` memanggil `assets/css/style.css`

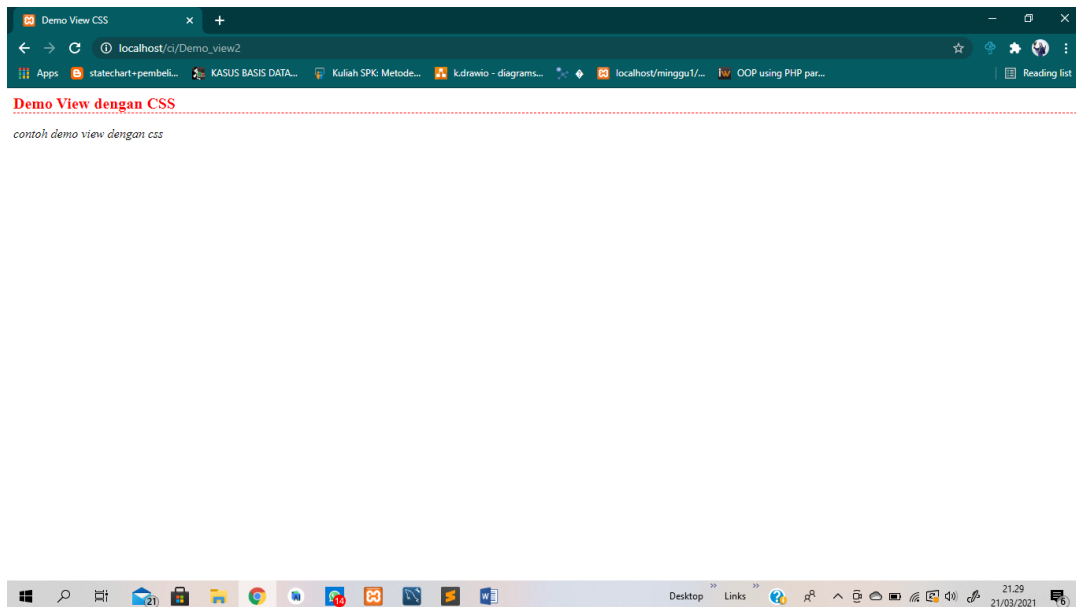


```
1 <?php
2 class Demo_view2 extends CI_Controller{
3     public function __construct(){
4         parent::__construct();
5         $this->load->helper('url');
6     }
7     public function index(){
8         $this->load->view('demoview');
9     }
10 }
11 ?>
```

Sebelumnya saya telah membuat `style.css` yang diletakkan pada folder `assets/css` dengan menambahkan beberapa style dengan perintah sebagai berikut :

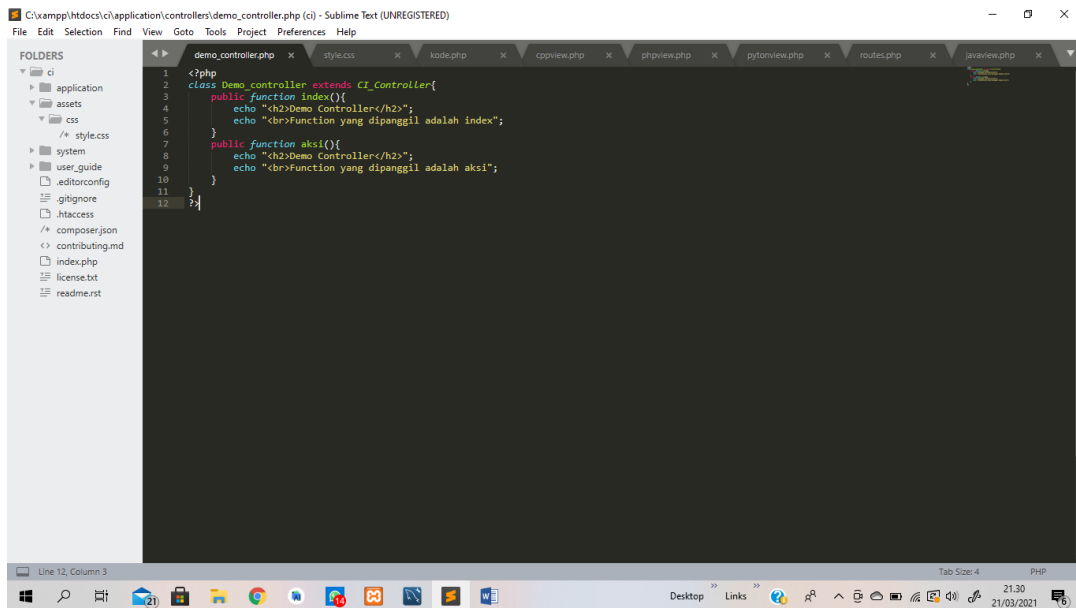


Untuk output sama dengan style yang sebelumnya dikarenakan saya memasukkan perintah yang sama pada program , namun yang membedakan hanya pada kode program ini memanggil style.css pada folder assets/css



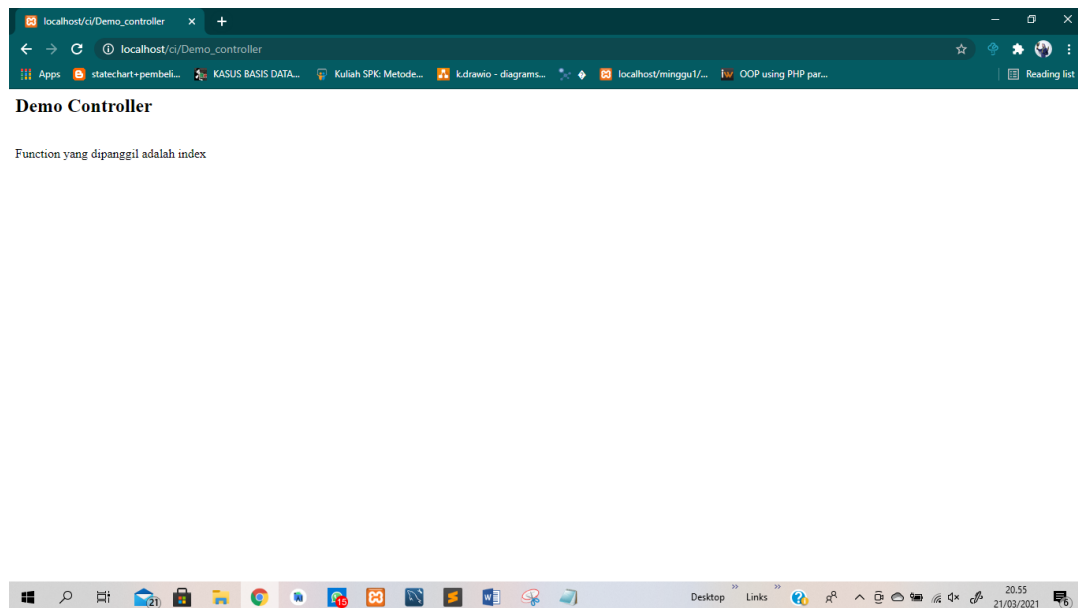
Controller

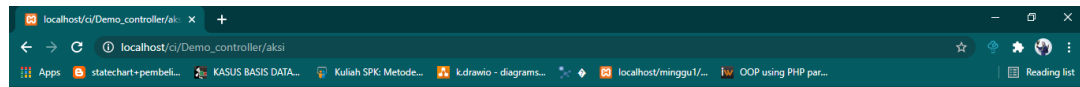
Demo_controller.php



```
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 ?>
```

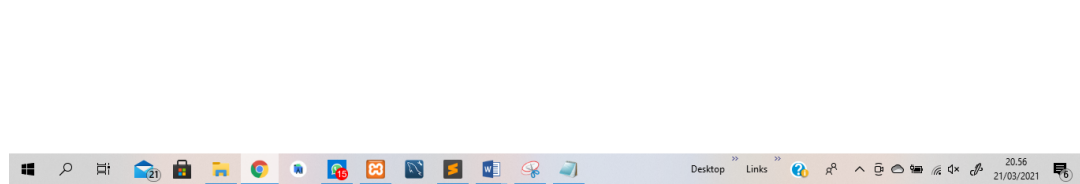
Pada method Demo_controller terdapat 2 function yakni index dan aksi yang dapat kita run dengan mengetikkan URL http://localhost/ci/demo_controller/index ataupun juga bisa dengan http://localhost/ci/demo_controller untuk menampilkan function index dan http://localhost/ci/demo_controller/aksi untuk menampilkan



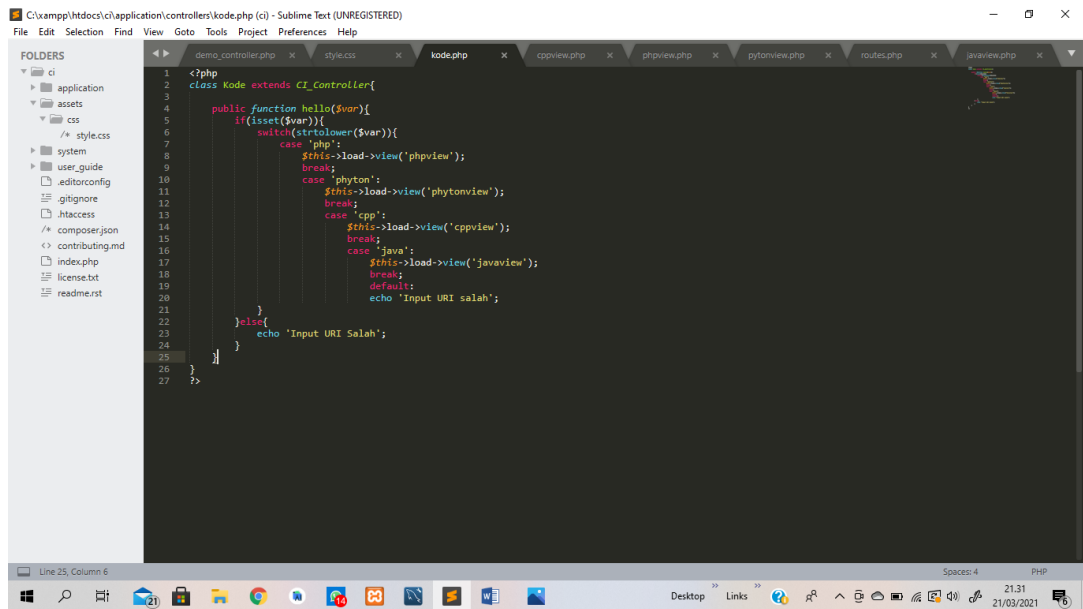


Demo Controller

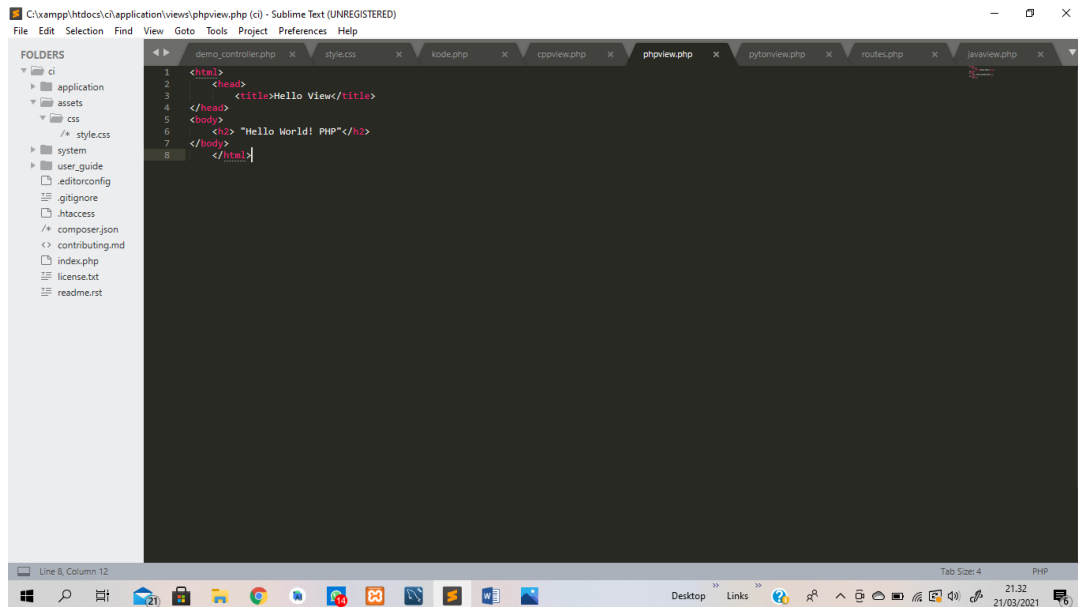
Function yang dipanggil adalah aksi



Kode.php



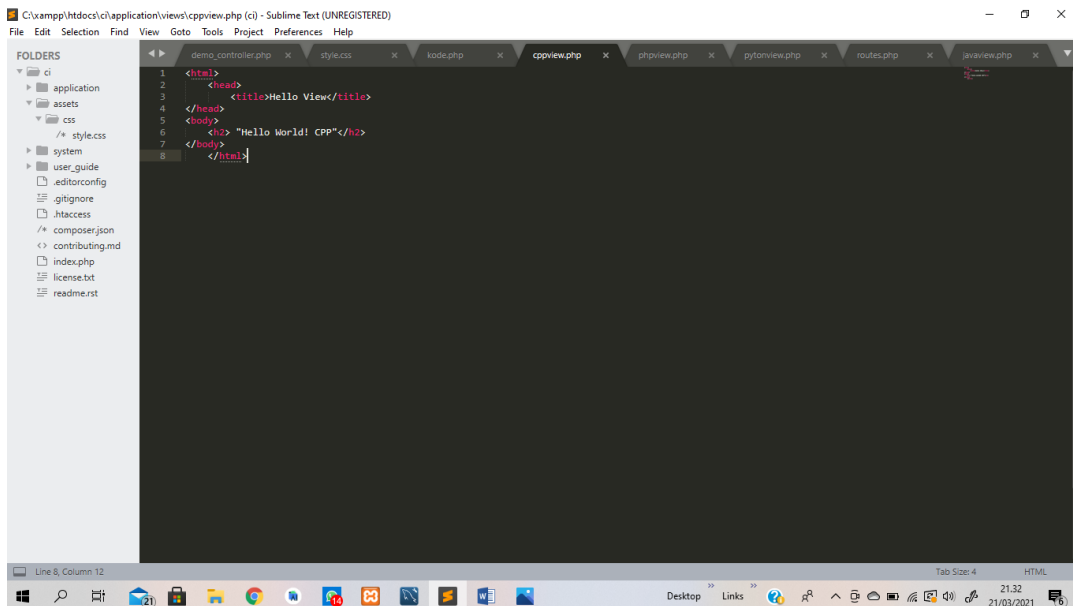
phpview.php



A screenshot of the Sublime Text editor interface. The title bar indicates the file path is C:\xampp\htdocs\application\views\phpview.php (ci) - Sublime Text (UNREGISTERED). The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. On the left, a 'FOLDERS' sidebar shows a project structure with folders like application, assets, css, system, user_guide, and files like .editorconfig, .gitignore, .htaccess, composer.json, contributing.md, index.php, license.txt, and readme.rst. The main editor area shows the content of phpview.php with line numbers 1 through 8. The code is an HTML template with a title 'Hello View' and a body containing 'Hello World! PHP'. The status bar at the bottom shows 'Line 8, Column 12', 'Tab Size: 4', and the file type 'PHP'. The Windows taskbar at the very bottom shows various application icons and the system clock at 21:32 on 21/03/2021.

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

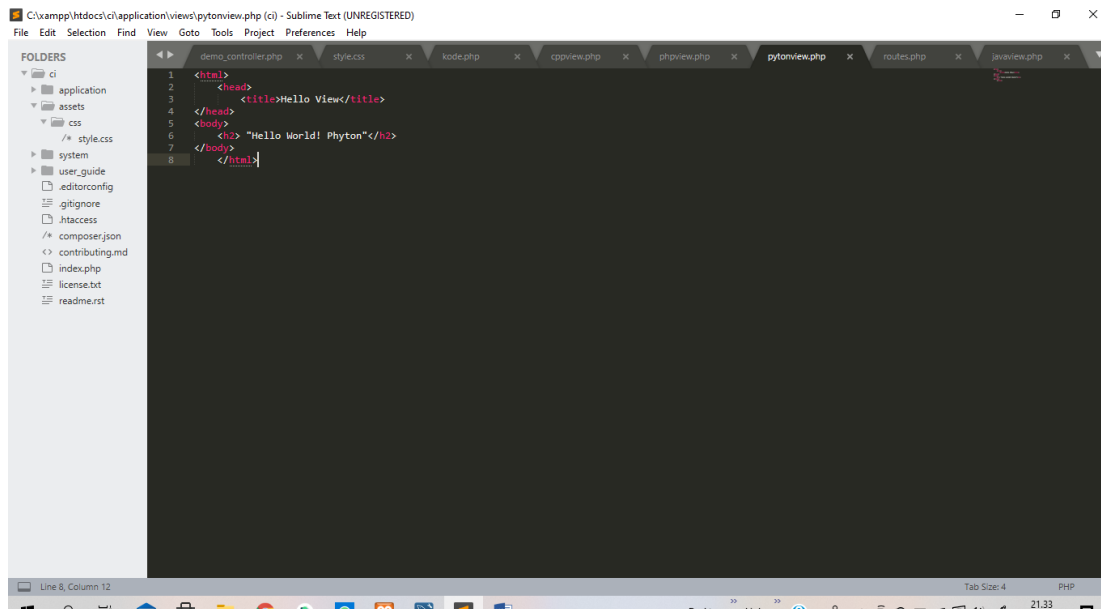
cppview.php



A screenshot of the Sublime Text editor interface, similar to the one above but for the file cppview.php. The title bar shows the path C:\xampp\htdocs\application\views\cppview.php (ci) - Sublime Text (UNREGISTERED). The menu bar and sidebar are identical. The main editor area shows the content of cppview.php with line numbers 1 through 8. The code is an HTML template, identical to the one above, but the body contains 'Hello World! CPP'. The status bar at the bottom shows 'Line 8, Column 12', 'Tab Size: 4', and the file type 'HTML'. The Windows taskbar at the very bottom shows various application icons and the system clock at 21:32 on 21/03/2021.

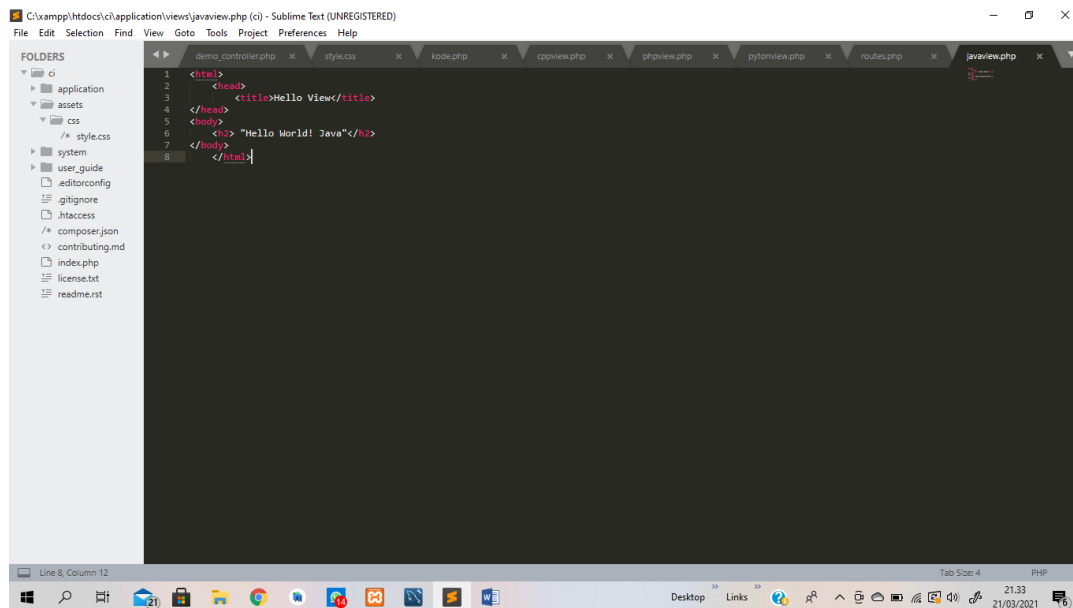
```
1 <html>
2   <head>
3     <title>Hello View</title>
4   </head>
5   <body>
6     <h2> "Hello World! CPP"</h2>
7   </body>
8 </html>
```

pythonview.php



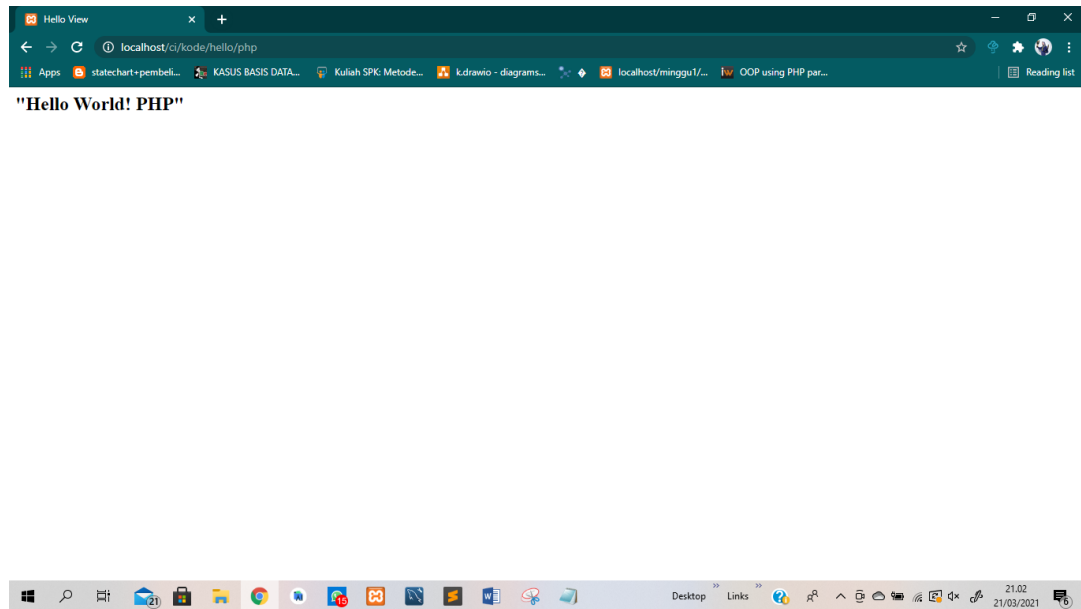
```
1 <html>
2 <head>
3 <title>Hello View</title>
4 </head>
5 <body>
6 <h2> "Hello World! Phyton"</h2>
7 </body>
8 </html>
```

javaview.php

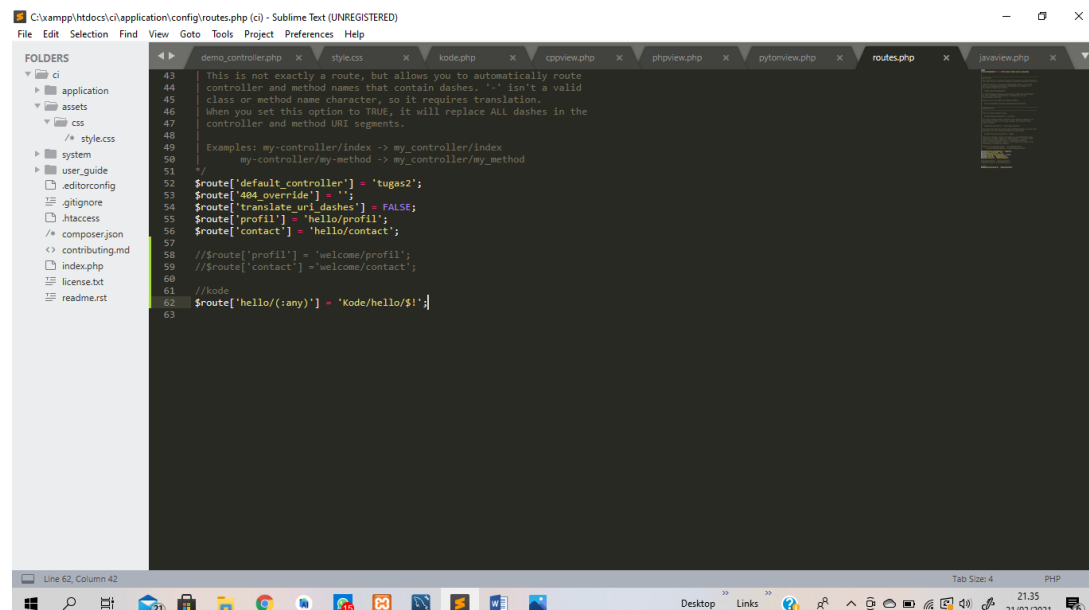


```
1 <html>
2 <head>
3 <title>Hello View</title>
4 </head>
5 <body>
6 <h2> "Hello World! Java"</h2>
7 </body>
8 </html>
```

Segmen di URL, mengikuti pendekatan Model-View-Controller, biasanya mewakili: <http://localhost/ci/kode/hello/php> . Segmen pertama mewakili kelas pengontrol yang harus dipanggil. Segmen kedua mewakili fungsi kelas, atau metode, yang harus dipanggil. Segmen ketiga, dan setiap segmen tambahan, mewakili ID dan variabel apa pun yang akan diteruskan ke pengontrol.



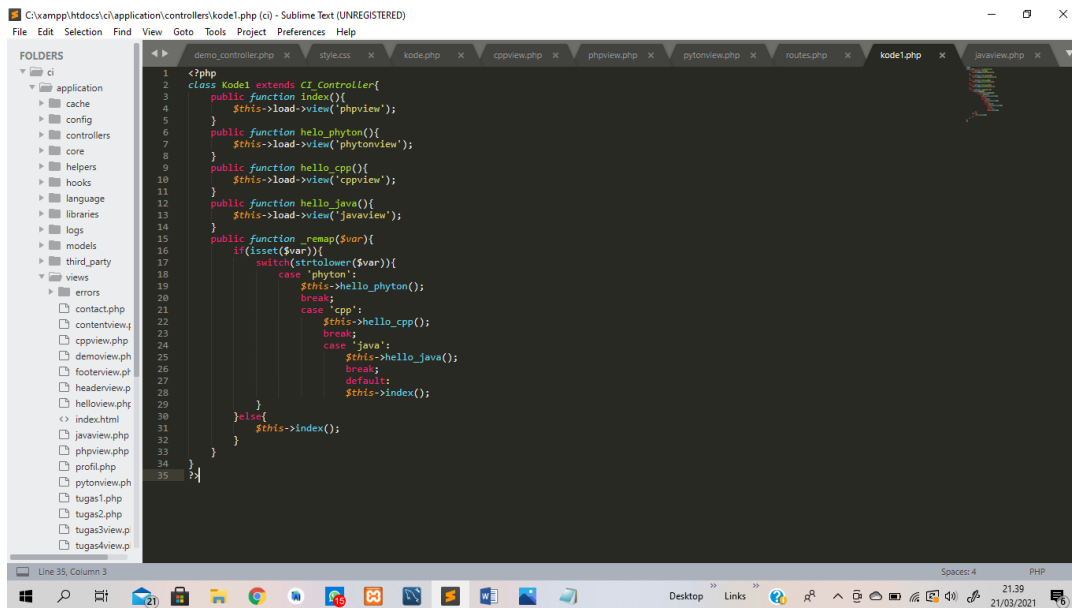
Library URI dan URL Helper berisi fungsi yang memudahkan untuk bekerja dengan data URI kita. Selain itu, URL kita dapat dipetakan ulang menggunakan fitur perutean URI agar lebih fleksibel. Sehingga saya menambahkan kode program pada route sebagai berikut :



Pada routes.php saya menambahkan `$route['hello/(:any)'] = 'Kode/hello/$1';` yang arti dari program tersebut adalah ketika kita mengetikkan pada URL `http://localhost/ci/hello/(apapun yg user ketikkan)` maka akan diarahkan ke controller Kode dan methodnya hello.

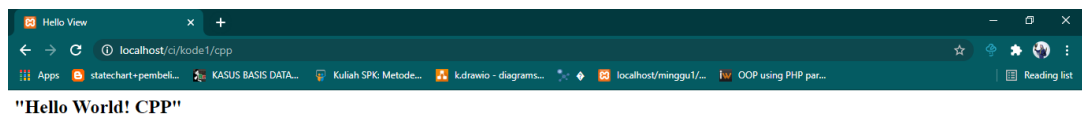
Memetakan nama metode yang akan dipanggil

Kode1

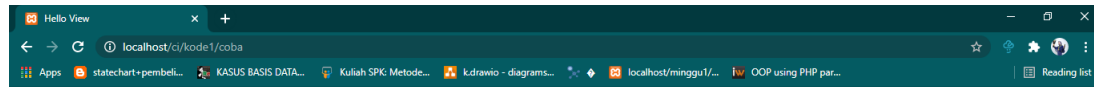


```
1 <?php
2 class Kode1 extends CI_Controller{
3     public function index(){
4         $this->load->view('phpview');
5     }
6     public function hello_python(){
7         $this->load->view('pythonview');
8     }
9     public function hello_cpp(){
10        $this->load->view('cppview');
11    }
12    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                    $this->index();
29            }
30        }else{
31            $this->index();
32        }
33    }
34 }
35 }
```

Penjelasan pada program diatas adalah apabila kita menuliskan apapun setelah method Kode1 , disini saya mencontohkan <http://localhost/ci/kode1/cpp> maka program akan menangkap function _remap dan kemudian dilanjutkan ke function index untuk mengarahkan view mana yang akan ditampilkan pada output.



Namun apabila variabel yang kita tuliskan tidak ada dalam function `_remap` maka program akan menampilkan halaman index.



"Hello World! PHP"



BAB IV

PENUTUP

4.1 Kesimpulan

URL routing (route) adalah salah satu metode yang digunakan untuk memetakan URL ke dalam sumber daya tertentu dengan memberikan nama lain dari alamat sumber daya yang dimaksud. URL routing sering digunakan untuk beberapa hal seperti menjadikan URL sumber daya yang sulit dibaca manusia dengan membuat pemetaan baru ke URL alias dari route yang lebih mudah dibaca manusia, membuat URL sumber daya menjadi lebih pendek dengan memberikan penamaan routing yang lebih pendek, dan memantau agar URL sesuai dengan format yang diinginkan dengan memanfaatkan fungsi regular expression (regex).

