

LAPORAN PEMROGRAMAN WEB FRAMEWORK

“Model dan Parsing Data”



DISUSUN OLEH :

EDO KRISTIAN YUSAK

E31192060

GOLONGAN C

MANAJEMEN INFORMATIKA

TEKNOLOGI INFORMASI

POLITEKNIK NEGERI JEMBER

2021

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi telah banyak membantu pekerjaan setiap orang, lembaga atau perusahaan dalam menyelesaikan suatu masalah dan menjalankan tugasnya. Ketatnya persaingan teknologi menyebabkan peningkatan kebutuhan akan penggunaan teknologi informasi. Setiap orang, lembaga atau perusahaan yang ada saat ini selalu mengikuti perkembangan teknologi kebutuhan informasi untuk dapat membantu kegiatan operasionalnya dengan menggunakan sistem yang lebih efektif dan efisien dalam menjalankan bisnisnya. Salah satu kegiatan bisnis yang sedang populer saat ini adalah pembuatan dan penggunaan website untuk berbagai keperluan bagi setiap orang, lembaga atau perusahaan dalam mempromosikan penjualan ataupun kebutuhan bisnis dan manajerial.

Selain itu berkembangnya keberadaan framework PHP (Hypertext Preprocessor) untuk pembuatan website menjadi tantangan tersendiri bagi user dalam membuat aplikasi berbasis web. Framework merupakan sekumpulan library yang diorganisasikan pada sebuah rancangan arsitektur untuk memberikan ketepatan, kecepatan, kemudahan dan konsistensi dalam pengembangan aplikasi. Framework PHP yang sangat banyak tersedia menjadi kendala bagi seseorang untuk memilih menggunakannya. Saat ini framework yang populer dan cukup banyak digunakan untuk aplikasi website seperti Laravel, Yii, CodeIgniter, Zend, CakePHP, Symfony, dan lain sebagainya.

1.2 Rumusan Masalah

- 1.2.1 Bagaimana konsep Model pada CI?
- 1.2.2 Bagaimana menerapkan parsing data pada CI?

1.3 Tujuan

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

BAB II

DASAR TEORI

2.1 Model

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::__construct();  
6      }  
7  
8      function getProducts()  
9      {  
10         return $this->db->query("SELECT * FROM AA_PRODUCTS");  
11     }  
12 }  
13 ?>
```

2.2 Parsing Data

Proses parsing data terjadi pada fungsi `parsingData()`. Sebelum proses parsing dimulai, data yang diterima yang telah ditampung sebelumnya ditampilkan/dikirimkan kembali melalui serial port. Parsing data dilakukan dengan cara mengecek tiap karakter yang ada pada variabel `dataIn`. Tiap karakter tersebut akan dibandingkan dengan *tail* (karakter terakhir paket data) yang berupa karakter pagar (#) dan karakter pemisah antar data yang berupa karakter koma(.). Pengecekan dimulai dari index 1 (karakter kedua) variabel `dataIn`, hal ini dikarenakan index 0 (karakter pertama) merupakan *header* (karakter pertama) dalam paket data.

Setiap karakter yang telah melalui proses pengecekan, akan ditampung kedalam sebuah variabel string yang dikelompokkan dalam sebuah array yang diberi nama `dt`. Jika pada proses pengecekan ditemukan karakter koma (,), maka dengan otomatis variabel penampung data setelah dicek akan berpindah dengan cara increment index array nya. Proses tersebut akan terus dilakukan berulang kali sampai perulangan berakhir. Setelah proses parsing data selesai, data hasil proses parsing yang terdapat pada array data akan ditampilkan. Dengan begitu, proses parsing data pun selesai dilakukan.

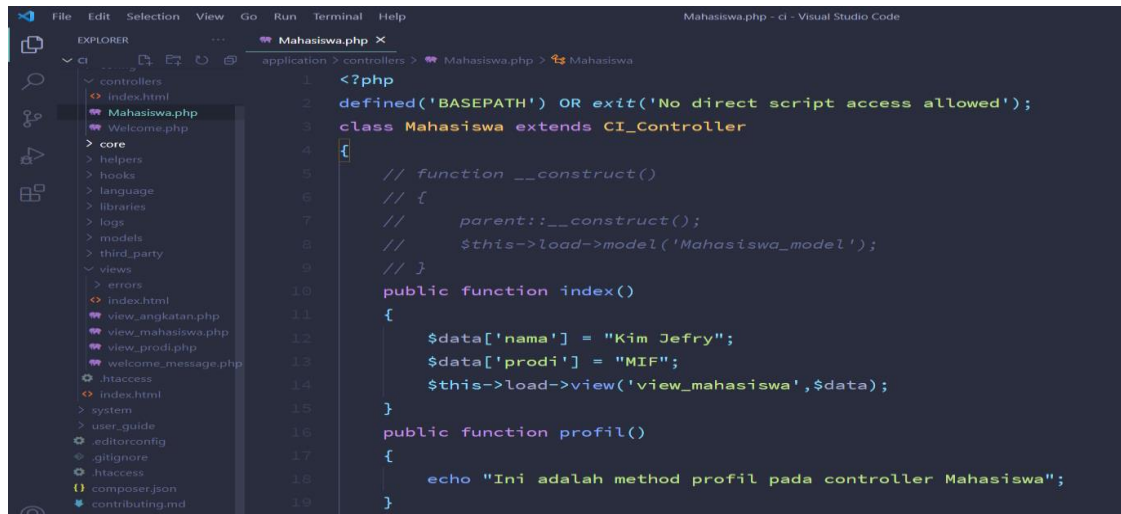
BAB III

HASIL DAN PEMBAHASAN

3.1 Kegiatan Praktikum

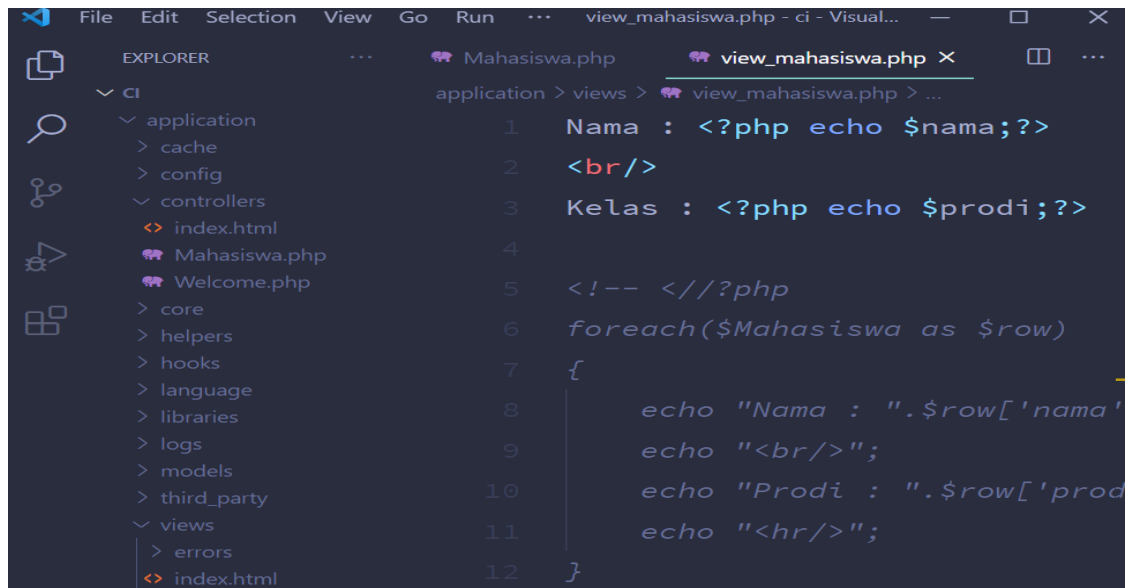
3.1.1 Parsing data dari controller ke views

Membuat file Controller dengan nama Mahasiswa.php dan file View dengan nama view_mahasiswa.php



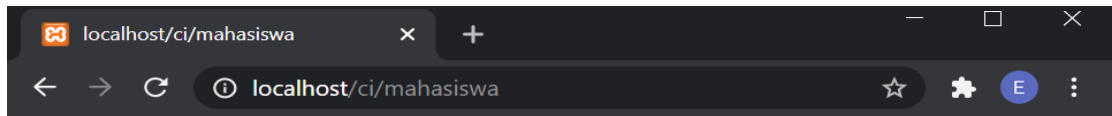
```
<?php
defined('BASEPATH') OR exit('No direct script access allowed');
class Mahasiswa extends CI_Controller
{
    // function __construct()
    // {
    //     parent::__construct();
    //     $this->load->model('Mahasiswa_model');
    // }
    public function index()
    {
        $data['nama'] = "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.



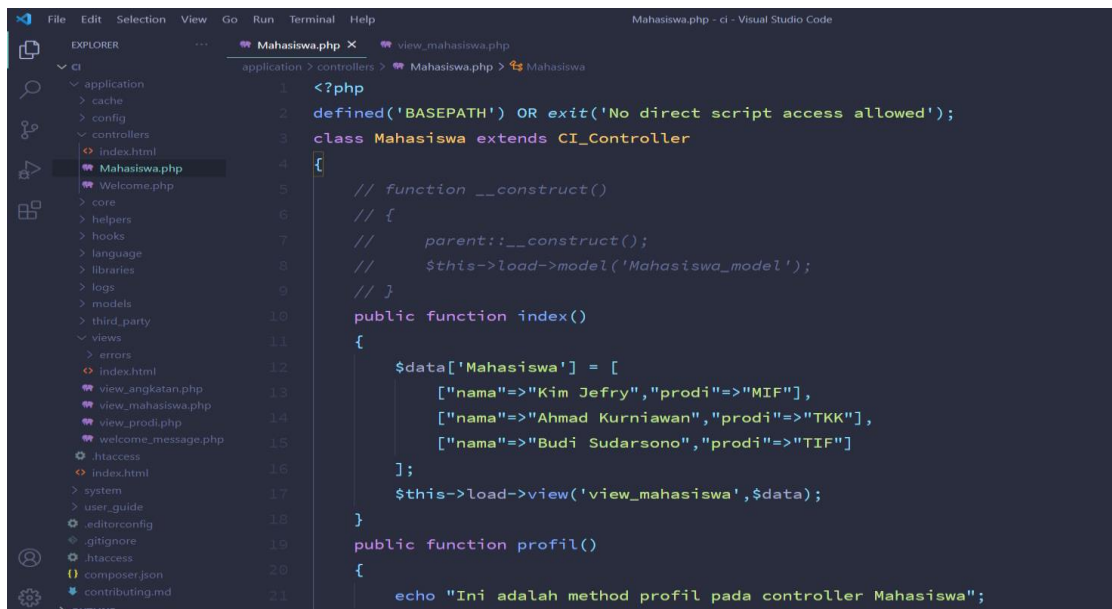
```
1 Nama : <?php echo $nama;?>
2 <br/>
3 Kelas : <?php echo $prodi;?>
4
5 <!-- <?php
6 foreach($Mahasiswa as $row)
7 {
8     echo "Nama : ".$row['nama'];
9     echo "<br/>";
10    echo "Prodi : ".$row['prodi'];
11    echo "<hr/>";
12 }
```

Output :

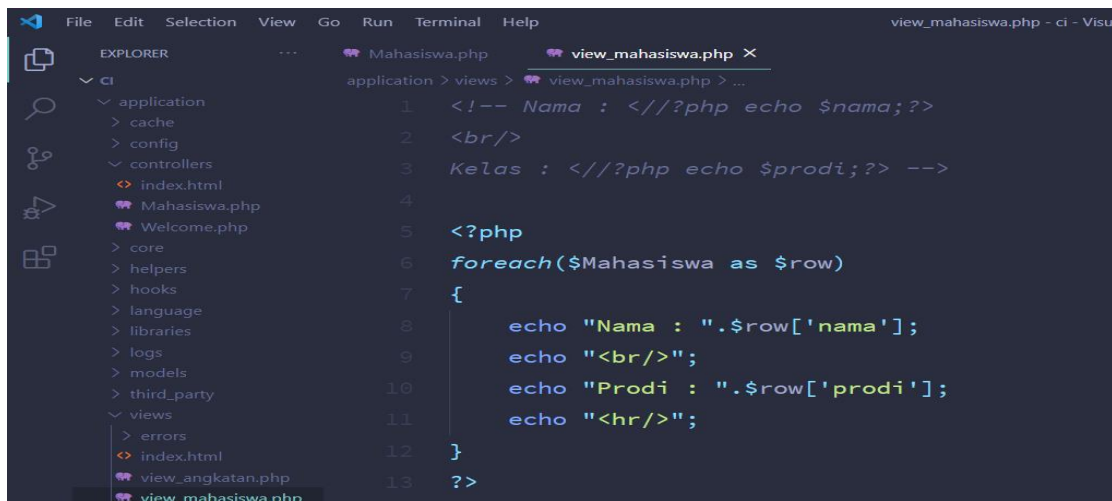


Nama : Kim Jefry
Kelas : MIF

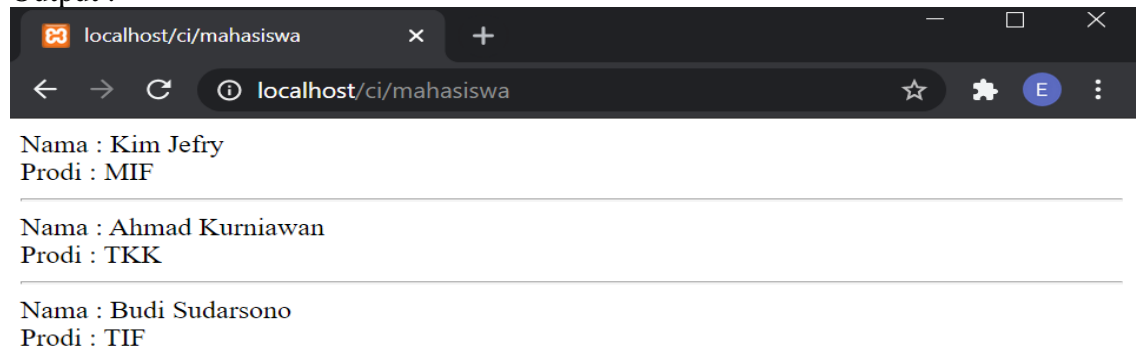
Saat kita mempassing data dari controller, kita perlu menuliskannya dalam bentuk array, dimana elemen array didalamnya (dalam contoh ini adalah nama, dan kelas) akan menjadi variabel yang dapat dipanggil dibagian view. Memodifikasi skrip dibagian function index controller Mahasiswa, menjadi seperti berikut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



Kemudian memodifikasi file view : view_mahasiswa.php menjadi seperti dibawah ini :

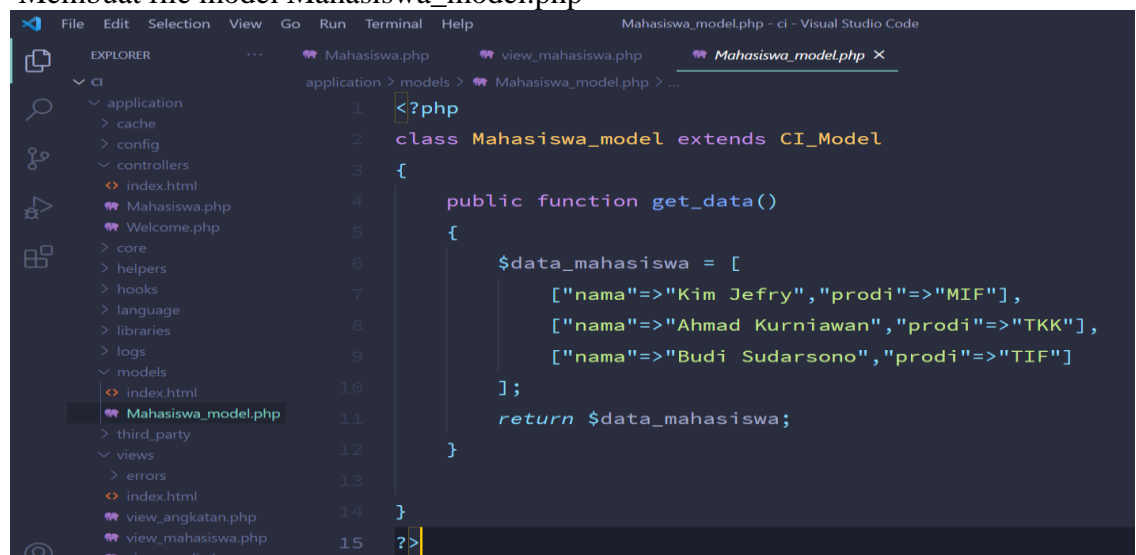


Output :

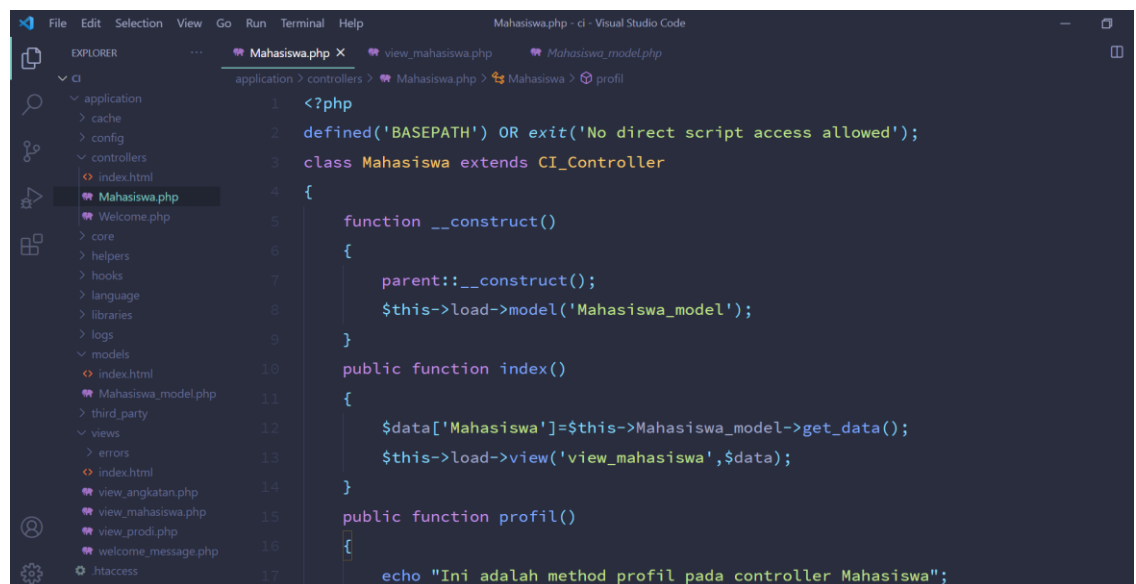


3.1.2 Parsing data dari Model ke Views

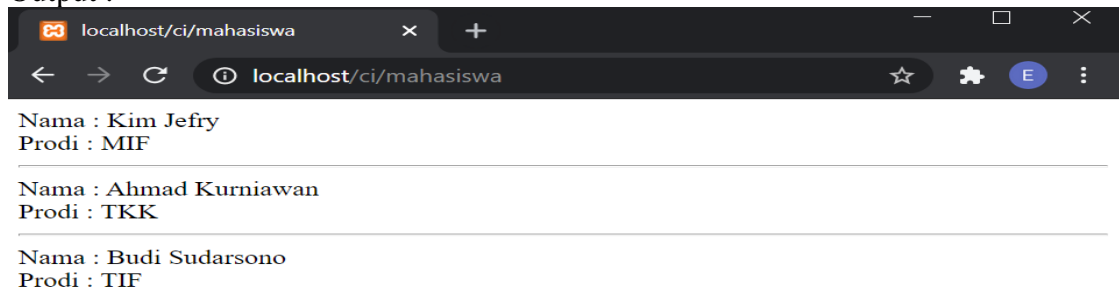
Membuat file model Mahasiswa_model.php



Mengubah bagian controller pada Mahasiswa.php



Output :



3.2 Latihan / Soal

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

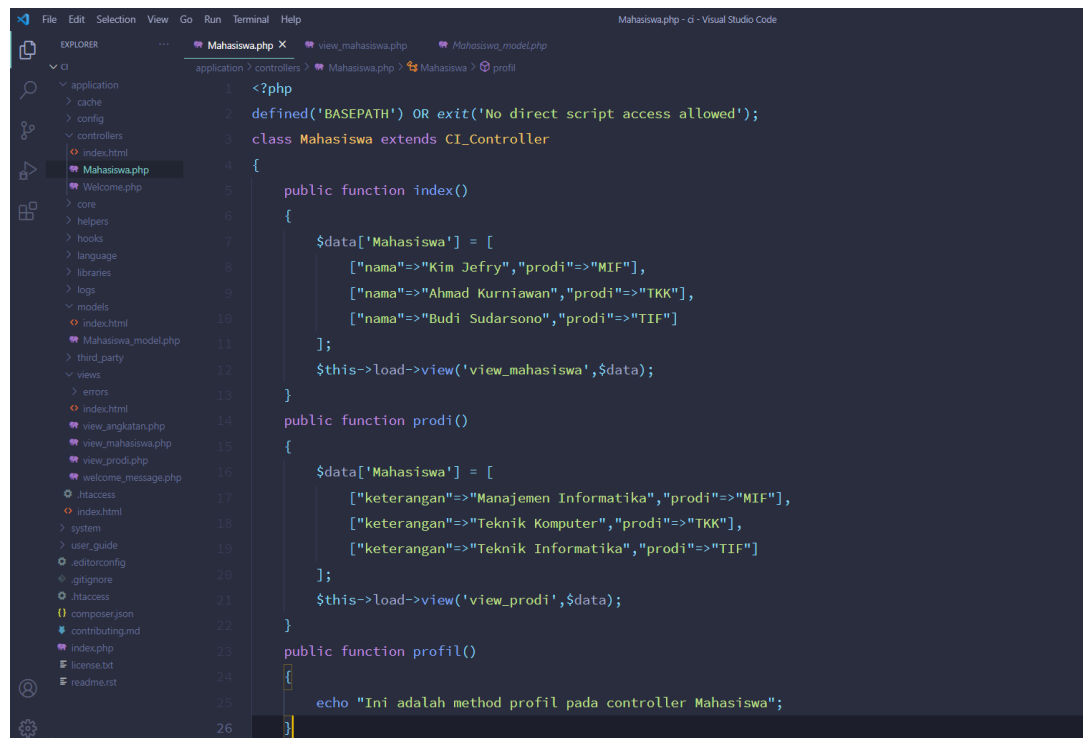
3.2.1.1 view_prodi.php

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

➤ Menambahkan function prodi dalam controller pada Mahasiswa.php



- Kemudian untuk menampilkan tabel seperti pada contoh maka memodifikasi file view : view_prodi.php menjadi seperti dibawah ini :

```

1 <table border="1" cellspacing="0">
2
3 <tr>
4 <th>
5     padding: 10px;
6 </th>
7 <th>
8     text-align: center;
9 </th>
10 </tr>
11 <tr>
12 <th>No</th>
13 <th>Prodi</th>
14 <th>Keterangan</th>
15 </tr>
16 <?php
17     $nomer=1;
18     foreach($Mahasiswa as $row)
19     {
20     ?>
21 <tr>
22 <td><?= $nomer; ?></td>
23 <td><?= $row["prodi"]; ?></td>
24 <td><?= $row["keterangan"]; ?></td>
25 </tr>
26 <?php $nomer++; ?>
27 <?php } ?>

```

- Output :

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

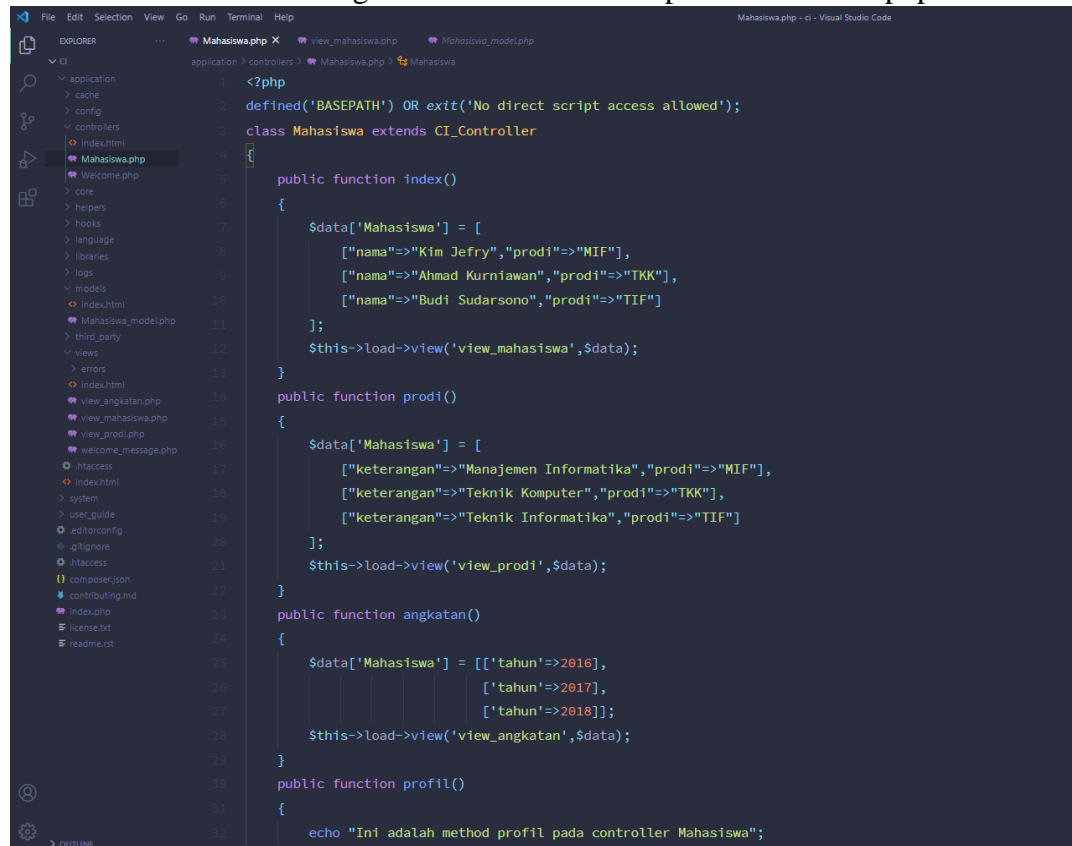
3.2.1.2 view_angkatan.php

Tampilan dari halaman tersebut adalah seperti berikut :

Daftar Angkatan

No	Tahun
1	2016
2	2017
3	2018

- Menambahkan function angkatan dalam controller pada Mahasiswa.php



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

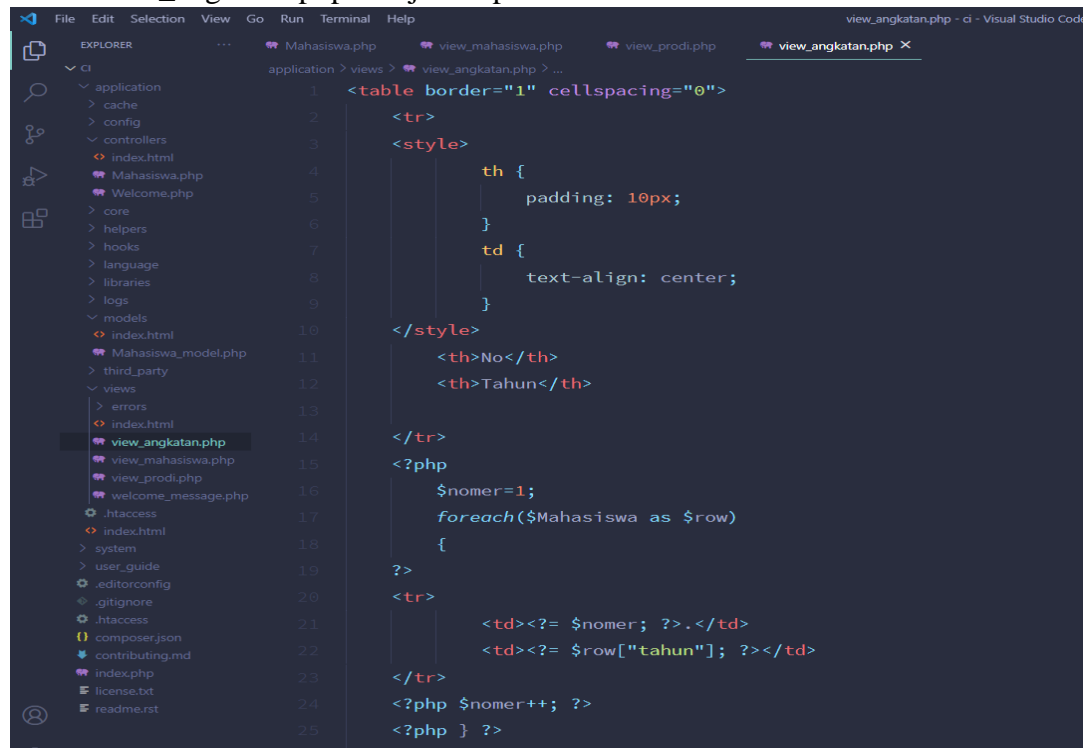
class Mahasiswa extends CI_Controller
{
    public function index()
    {
        $data['Mahasiswa'] = [
            ["nama"=>"Kim Jefry","prod1"=>"MIF"],
            ["nama"=>"Ahmad Kurniawan","prod1"=>"TKK"],
            ["nama"=>"Budi Sudarsono","prod1"=>"TIF"]
        ];
        $this->load->view('view_mahasiswa',$data);
    }

    public function prod1()
    {
        $data['Mahasiswa'] = [
            ["keterangan"=>"Manajemen Informatika","prod1"=>"MIF"],
            ["keterangan"=>"Teknik Komputer","prod1"=>"TKK"],
            ["keterangan"=>"Teknik Informatika","prod1"=>"TIF"]
        ];
        $this->load->view('view_prod1',$data);
    }

    public function angkatan()
    {
        $data['Mahasiswa'] = [['tahun'=>2016],
                               ['tahun'=>2017],
                               ['tahun'=>2018]];
        $this->load->view('view_angkatan',$data);
    }

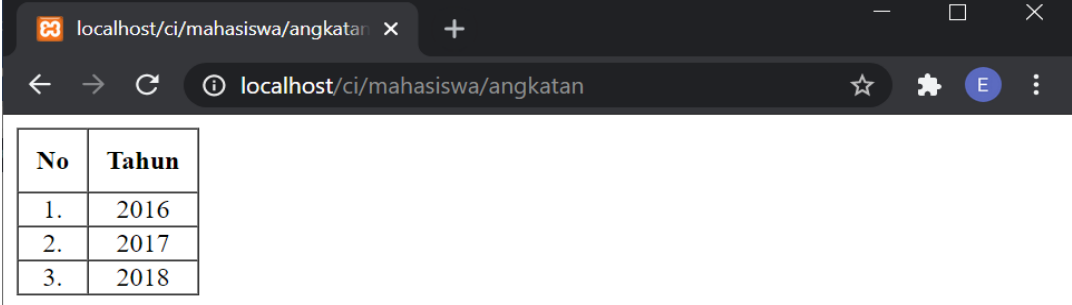
    public function profil()
    {
        echo "Ini adalah method profil pada controller Mahasiswa";
    }
}
```

- Kemudian untuk menampilkan tabel seperti pada contoh maka memodifikasi file view : view_angkatan.php menjadi seperti dibawah ini :



```
<table border="1" cellspacing="0">
<tr>
<style>
    th {
        padding: 10px;
    }
    td {
        text-align: center;
    }
</style>
<th>No</th>
<th>Tahun</th>
</tr>
<?php
    $nomer=1;
    foreach($Mahasiswa as $row)
    {
    <tr>
        <td><?= $nomer; ?></td>
        <td><?= $row["tahun"]; ?></td>
    </tr>
    <?php $nomer++; ?>
    <?php } ?>
```

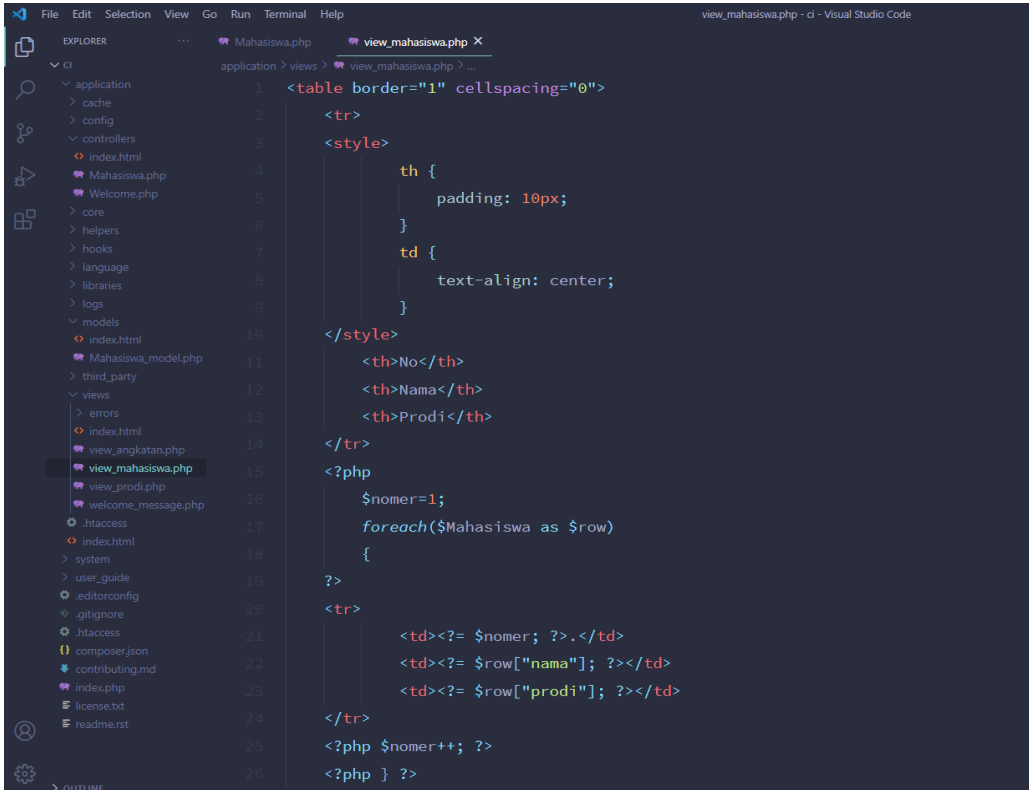
➤ Output :



No	Tahun
1.	2016
2.	2017
3.	2018

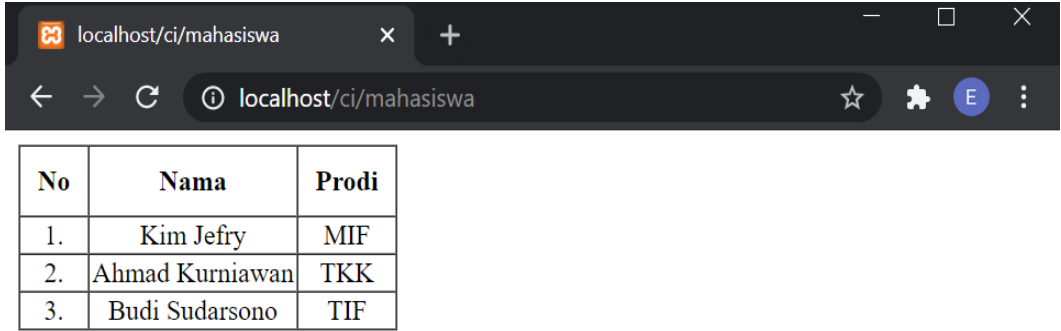
3.2.1.3 Mengubah tampilan view_mahasiswa menjadi bentuk tabel

➤ Untuk menampilkan tabel pada view_mahasiswa seperti pada contoh maka memodifikasi file view : view_mahasiswa.php menjadi seperti dibawah ini :



```
1 <table border="1" cellspacing="0">
2
3 <tr>
4
5 <style>
6     th {
7         padding: 10px;
8     }
9     td {
10        text-align: center;
11    }
12 </style>
13 <th>No</th>
14 <th>Nama</th>
15 <th>Prodi</th>
16 </tr>
17 <?php
18     $nomer=1;
19     foreach($Mahasiswa as $row)
20     {
21     <?>
22     <tr>
23         <td><?= $nomer; ?></td>
24         <td><?= $row["nama"]; ?></td>
25         <td><?= $row["prodi"]; ?></td>
26     </tr>
27     <?php $nomer++; ?>
28     <?php } ?>
```

➤ Output :



No	Nama	Prodi
1.	Kim Jefry	MIF
2.	Ahmad Kurniawan	TKK
3.	Budi Sudarsono	TIF

BAB IV

PENUTUP

4.1 Kesimpulan

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.

Proses parsing data terjadi pada fungsi `parsingData()`. Sebelum proses parsing dimulai, data yang diterima yang telah ditampung sebelumnya ditampilkan/dikirimkan kembali melalui serial port. Parsing data dilakukan dengan cara mengecek tiap karakter yang ada pada variabel `dataIn`. Tiap karakter tersebut akan dibandingkan dengan *tail* (karakter terakhir paket data) yang berupa karakter pagar (#) dan karakter pemisah antar data yang berupa karakter koma(.). Pengecekan dimulai dari index 1 (karakter kedua) variabel `dataIn`, hal ini dikarenakan index 0 (karakter pertama) merupakan *header* (karakter pertama) dalam paket data.

DAFTAR PUSTAKA

- Deacon, J. (2009), "Model-View-Controller (MVC) Architecture", JOHN DEACON Computer Systems Development, Consulting & Training, Copyright © 2009, John Deacon Page 1 of 6, <http://www.jdl.co.uk> , <http://www.johndeacon.net> .
- Octafian, D. T. (2015). Web Multi E-Commerce Berbasis Framework CodeIgniter. J. Teknol. Dan Inform. 5, 1–22 .
- Wardana, (2010). "Pemahaman Konsep Mode-View-Controller," in Menjadi Master PHP dengan Framework Code Igniter, Jakarta, Elex Media Computindo. p. 52.