IMPLEMENTASI MVC PADA SITUS PORTAL PENCARIAN UNIVERSITAS DI DAERAH ISTIMEWA YOGYAKARTA

Rionaldi Sugiarto
Katon Wijana, Wimmie Handiwidjojo

Abstrak

Daerah Istimewa Yogyakarta merupakan kota pelajar dan memiliki banyak perguruan tinggi. Karena begitu banyaknya perguruan tinggi, akan menyebabkan para calon mahasiswa baru mengalami kesulitan dalam mencari informasi seputar perguruan tinggi di Daerah Istimewa Yogyakarta. Selain itu, setiap perguruan tinggi memiliki program studi yang berbeda-beda.

Oleh karena itu, penulis menerapkan implementasi MVC (Model-View-Controller) pada situs portal pencarian universitas di Daerah Istimewa Yogyakarta. MVC merupakan suatu metode yang digunakan penulis untuk merancang sistem berbasis website. Tampilan (view) website dihasilkan dari pengolahan database yang terdapat dalam model. Proses pengolahan tersebut dengan menggunakan query SQL (Structured Query Language). Agar hasil dari pengolahan database tersebut dapat ditampilkan, maka diperlukan controller untuk menghubungkan model dengan view.

Hasil dari implementasi MVC (Model-View-Controller) pada situs portal ini ternyata dapat membantu calon mahasiswa baru untuk mengetahui perguruan tinggi apa saja yang tersedia di Daerah Istimewa Yogyakarta dan dapat melakukan pencarian perguruan tinggi yang diinginkan beserta peta lokasinya. Sehingga dapat disimpulkan bahwa penggunaan MVC sangat berperan penting dalam pembuatan situs portal pencarian universitas di Daerah Istimewa Yogyakarta.

Kata Kunci : *Universitas, Daerah Istimewa Yogyakarta, MVC, SQL*

1. Pendahuluan

Daerah Istimewa Yogyakarta memiliki begitu banyak universitas, baik universitas negeri maupun universitas swasta. Selain itu juga terdapat banyak sekolah tinggi dan akademi yang tersebar di seluruh Daerah Istimewa Yogyakarta. Hal tersebut akan menimbulkan masalah bagi calon mahasiswa baru, baik mereka yang berasal dari kota Yogyakarta maupun yang berasal dari luar kota Yogyakarta. Masalah yang terjadi adalah mereka tidak tahu universitas apa saja yang terdapat di Daerah Istimewa Yogyakarta dan dimana letak universitas tersebut.

Untuk mengatasi masalah tersebut, penulis merancang sebuah situs portal untuk pencarian universitas di Daerah Istimewa Yogyakarta. Situs portal tersebut dirancang dengan dukungan MVC (Model-View-Controler). Sedangkan pencarian universitas menggunakan *query* SQL. Dengan demikian, kata kunci pencarian universitas tersebut dapat berdasarkan penggalan nama maupun singkatan dari universitas tersebut. Selain itu program bantu ini dilengkapi dengan dukungan *Google Maps* sehingga calon mahasiswa baru dapat mengetahui letak lokasi universitas yang diinginkan.

Dengan implementasi MVC pada situs portal tersebut, diharapkan calon mahasiswa baru dapat terbantu dalam memilih universitas yang diinginkan. Sehingga pada saat calon mahasiswa baru tersebut sudah lulus Sekolah Menengah Atas, mereka sudah mempunyai tujuan yang pasti untuk melanjutkan studinya di universitas yang mereka inginkan.

2. Landasan Teori

2.1. Sejarah MVC (*Model-View-Controller*)

Model-View-Controller atau MVC adalah sebuah metode untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memprosesnya

(Controller). Dalam implementasinya, kebanyakan *framework* dalam aplikasi website adalah berbasis arsitektur MVC (Griffiths , 2010).

MVC memisahkan pengembangan aplikasi berdasarkan komponen utama yang membangun sebuah aplikasi seperti manipulasi data, antarmuka pengguna, dan bagian yang menjadi kontrol dalam sebuah aplikasi. MVC pertama kali dipublikasikan oleh peneliti Xerox Parac yang bekerja dalam pembuatan bahasa pemrograman *Smalltalk* sekitar tahun 1970 sampai 1980.

2.2. Bagian-Bagian MVC

MVC terdiri atas tiga bagian penting, yaitu *Model*, *View*, dan *Controller*. *Model* mewakili struktur data, biasanya model berisi fungsi-fungsi yang membantu seseorang dalam pengelolaan *database* atau manipulasi data seperti *insert, update, delete*. *View* adalah bagian yang mengatur tampilan berupa halaman web.

Controller merupakan bagian yang menjembatani *model* dan *view*. *Controller* berisi perintah-perintah yang berfungsi untuk memproses suatu data (*Model*) dan mengirimkannya ke halaman web (*View*). Dengan menggunakan metode MVC, maka aplikasi akan lebih mudah untuk dirawat dan dikembangkan. Untuk memahami metode pengembangan aplikasi menggunakan MVC diperlukan pengetahuan tentang pemrograman berorientasi objek (OOP).

2.3. Jenis-Jenis MVC

Terdapat dua jenis dalam MVC, diantaranya adalah *Server Side* MVC dan *Mixed Client and Server Side* MVC. *Server Side* MVC biasa terjadi pada aplikasi web tradisional, yang tidak melibatkan *client side* seperti *Javascript, Java Applet, Flash*, dan lain-lain. *Server Side* MVC menyerahkan keseluruhan proses bisnis pada *server*, aplikasi pada sisi pengguna hanya dapat menerima. MVC jenis ini kadang-kadang disebut juga dengan nama *Thin Client*.

Pada *Mixed Client Side and Server Side* MVC 1, *client* tidak menggunakan model sebagai jembatan untuk melakukan komunikasi pada server, dibandingkan dengan *Server Side* MVC, arsitektur ini memiliki tingkat kompleksitas yang lebih tinggi karena lebih banyak komponen yang terlibat. Untuk selanjutnya arsitektur ini disebut dengan *Mixed* MVC 1.

Pada *Mixed Client Side and Server Side* MVC 2, *client* menggunakan model sebagai jembatan untuk melakukan komunikasi pada *server*, dibandingkan dengan arsitektur MVC yang lain, arsiterktur ini memiliki tingkat kompleksitas yang paling tinggi karena lebih banyak komponen yang terlibat, sehingga membutuhkan sumber daya yang lebih besar pula. Untuk selanjutnya arsitektur ini disebut dengan *Mixed* MVC 2.

2.4. Rich Internet MVC

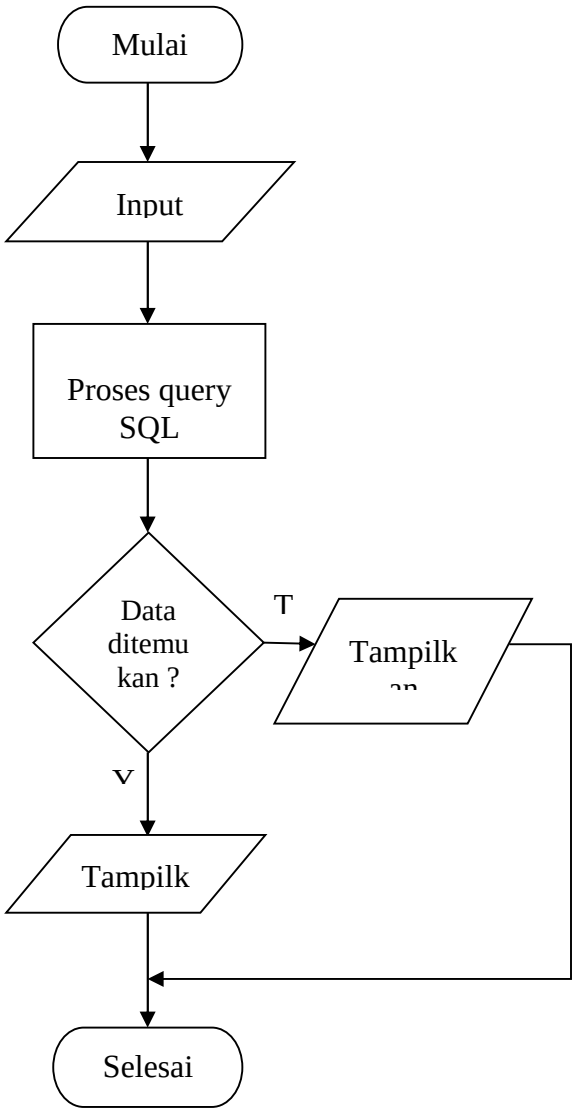
Application MVC *Rich Internet Application* (RIA) disebut juga dengan nama *Fat Client*, karena merupakan aplikasi web yang memiliki kemampuan dan fungsi hampir seperti aplikasi *desktop*. RIA pada sisi *client*, memiliki mesin untuk mengambil data yang berada pada *server*, sehingga pada *client* terdapat bagian MVC sendiri dan hanya membutuhkan bagian *model* pada sisi *server*.

3. Perancangan Sistem

3.1 Perancangan Input

Data yang digunakan penulis dalam penelitian ini merupakan data yang didapat dari hasil pencarian di internet. Hasil pencarian tersebut kemudian disaring secara manual oleh penulis dengan cara mengecek alamat situs perguruan tinggi tersebut satu persatu. Kemudian data yang sudah tersaring tersebut dimasukan secara manual kedalam database MySQL.

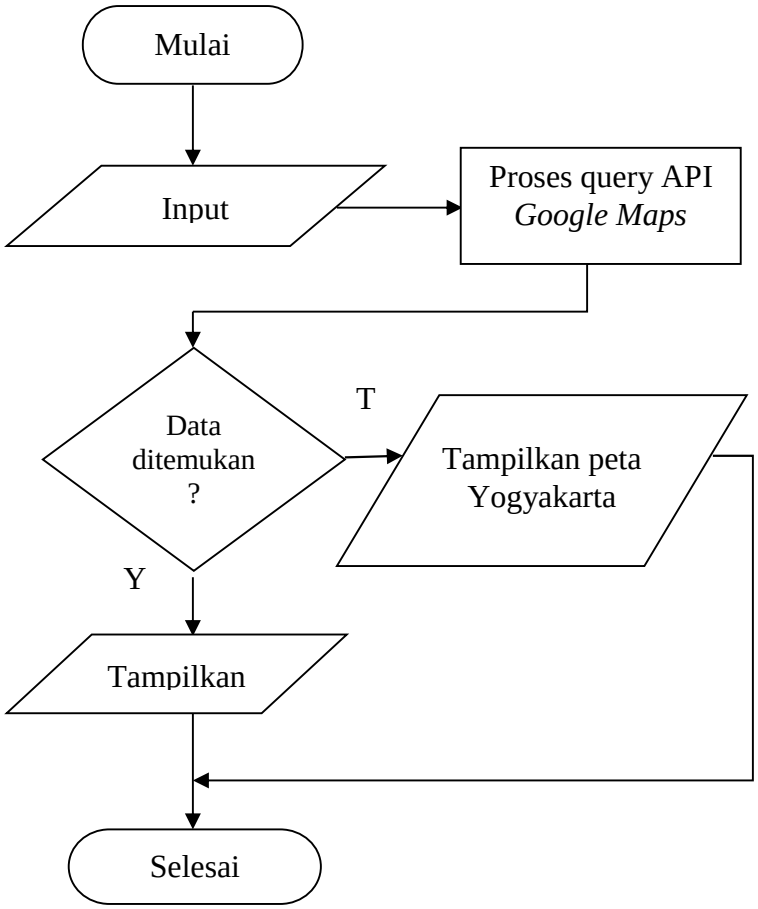
3.2 *Flowchart* Pencarian Universitas



Gambar 1 Flowchart Pencarian Universitas

Flowchart diatas menggambarkan bagaimana proses penginputan pencarian universitas sampai dengan menampilkan output. Inputan data dapat berupa nama, potongan nama, maupun nama program studi. Jika data tersebut ditemukan, maka output yang akan dihasilkan adalah menampilkan data yang sesuai dengan yang diinginkan oleh pengguna. Sedangkan jika data tidak ditemukan dalam database, maka output yang akan dihasilkan adalah menampilkan semua data yang tersedia di dalam database.

3.3 *Flowchart* Pencarian Lokasi Universitas

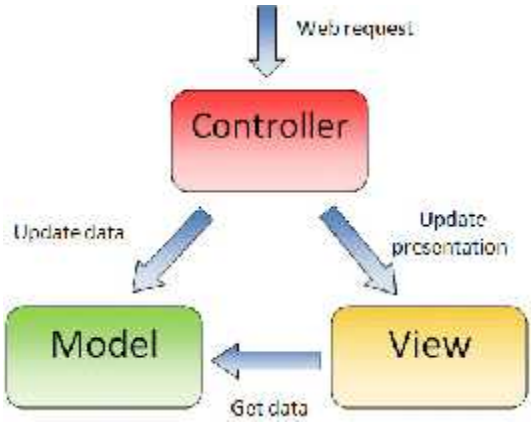


Gambar 2 Flowchart Pencarian Lokasi Universitas

Flowchart diatas menggambarkan bagaimana proses pencarian lokasi universitas di Daerah Istimewa Yogyakarta sampai dengan menampilkan output. Inputan dapat berupa nama universitas maupun nama suatu tempat. Jika data ditemukan, maka output yang akan dihasilkan adalah menampilkan peta universitas yang dituju. Sedangkan jika data tidak ditemukan, maka output yang akan dihasilkan adalah menampilkan peta Kota Yogyakarta.

3.4 Perancangan MVC (*Model-View-Controller*)

MVC merupakan pola perancangan yang digunakan untuk mengembangkan aplikasi (Pratama, 2010). MVC memisahkan implementasi antara *Model* yaitu pemodelan entitas bisnis dalam *real world* ke bentuk *object*, dengan *view* yang merupakan bagian yang nantinya akan berhadapan langsung dengan pengguna dan *Controller* yang merupakan bagian yang mengandung proses bisnis dimana bagian ini menghubungkan antara *event-event / input* dari *user* yang dilakukan melalui UI dengan *object* / entitas bisnis yang terlibat, sehingga apa yang ingin dilakukan oleh pengguna terhadapt objek, bisa dilaksanakan.



Gambar 3 Diagram MVC

Pada Gambar 3.5 diatas, menjelaskan tentang bagaimana hubungan antara *Model, View,* dan *Controller*. *Controller* dapat mengubah data yang ada di *model* dan dapat mengubah hasil tampilan dari *view*. Selain itu, *controller* berfungsi untuk menghubungkan *model* dengan *view*. Data yang terdapat ditampilkan di *view*, diambil dari *model* melalui *controller* melalui suatu *parameter*.

3.5 Perancangan MVC Pada Pencarian Universitas

Perancangan MVC pada pencarian universitas diperlukan *model* untuk melakukan pengolahan *database* pada tabel kampus dan prodi. Pengolahan *database* tersebut menggunakan *query* SQL yaitu SELECT. Berikut adalah contoh perancangan *model* untuk pencarian universitas.

```
<?php
 class Cari_model extends CI_Model {
 public function sKampus($nama)
 {
          return $this->db->query("SELECT * FROM kampus
          JOIN prodi where kampus.no = prodi.no AND
          nama like '%" . $nama . "%'or kampus.no =
          prodi.no AND nik like '%" . $nama . "%'or
          kampus.no = prodi.no AND jurusan like '%"
          .$nama . "%'");
 }
}
```

Program 1 Perancangan *Model* pencarian universitas

Perancangan selanjutnya adalah membuat *view* untuk menampilkan data dari hasil pengolahan *database* yang sudah dibuat di *model*. Berikut adalah contoh perancangan *view* untuk menampilkan hasil pengolahan dari *model*.

```
<HTML>
<HEAD>
<TITLE>Halaman Pencarian Universitas</TITLE>
</HEAD>
<BODY>
```

```
<table align="center">
     <tr>
       <td>Pencarian</td>
           <td>
           <form method="post">
          <input  type="text"  class="first"  id="name"
          name="nama" size="50" />
           </form>
           </td>
     </tr>
</table>

<table>
        <tr>
           <th width="25">No</th>
           <th width="200">Universitas</th>
           <th width="100">Singkatan</th>
           <th width="125">Program Studi</th>
        </tr>
      <?php $no_urut = 1 ?>
        <?php foreach($kampus->result() as $b) : ?>
        <tr>
          <td align="center">
          <?php echo $no_urut ?></td>
           <td><?php echo $b->nama ?></td>
           <td><?php echo $b->nik ?></td>
   <td><?php echo $b->jurusan ?></td>
        </tr>
        <?php $no_urut++ ?>
        <?php endforeach ?>
</table>
</BODY>
</HTML>
```

Program 2 Perancangan *View* Pencarian Universitas

Setelah model dan view selesai dibuat, langkah terakhir adalah merancang *controller* untuk menghubungkan *model* dan *view* yang sudah dirancang diatas. Berikut adalah contoh perancangan *controller* untuk menghubungkan *model* dengan *view*.

```
<?php
class CariKampus extends CI_Controller {
 public function __construct(){
   parent::__construct();
   $this->load->model('cari_model');
 }

 public function search()
 {
   $nama = "";
```

```
    $nik  = "";

    if(!empty($_POST['nama']))
    {
      $nama = $_POST['nama'];
    }

            $data['kampus'] = $this->
            cari_model->sKampus($nama);
    $this->load->view('cari_view',$data);
  }
}
?>
```

Program 3 Perancangan *Controller* Pencarian Universitas

Setelah ketiga langkah tersebut berhasil dirancang, maka hasil dari perancangan MVC pencarian tersebut dapat diuji coba melalui jaringan lokal yang biasa disebut *localhost*. Kata kunci pencarian tersebut berdasarkan *parameter nama* dari *textbox* yang terdapat di *view* dan kemudian dihubungkan dengan *paramater $nama* yang terdapat di *model* melalui *$data* yang terdapat di *controller*.

4. Implementasi Sistem

Tahap implementasi sistem adalah tahap dimana perancangan sistem diwujudkan dalam bentuk aplikasi nyata. Implementasi sistem dibagi menjadi tiga hal yaitu tahap persiapan, implementasi sistem untuk pengguna, kemudian dilanjutkan dengan analisis untuk proses pengujian hasil output sistem.

Tampilan yang dihasilkan setelah perancangan MVC diatas berhasil dibuat adalah sebagai berikut.

Gambar 4 Halaman Pencarian Universitas

Gambar 4 diatas merupakan bentuk halaman pencarian universitas. Pada halaman ini hanya tersedia satu *textbox* untuk menginputkan pencarian universitas. Kata kunci yang dapat diinputkan seperti nama atau potongan kata dari universitas, singkatan universitas, maupun program studi.

Selain itu halaman ini juga tersedia fasilitas pencarian lanjutan. Fasilitas pencarian lanjutan tersebut merupakan sebuah *link* yang akan men-*direct* ke halaman pencarian lanjutan.

Ketika kata kunci yang diinginkan berhasil dimasukan ke dalam *textbox*, maka sistem akan memproses kata kunci tersebut dan akan menampilkan hasil yang sesuai dengan kata kunci tersebut. Berikut adalah contoh hasil penginputan dengan kata kunci "Duta".

Tabel 1 Hasil Pencarian Universitas

No	Universitas	Singkatan	Program Studi
1	Universitas Kristen Duta Wacana	UKDW	Teknik Informatika
2	Universitas Kristen Duta Wacana	UKDW	Sistem Informasi
3	Universitas Kristen Duta Wacana	UKDW	Manajemen
4	Universitas Kristen Duta Wacana	UKDW	Akuntansi
5	Universitas Kristen Duta Wacana	UKDW	Biologi
6	Universitas Kristen Duta Wacana	UKDW	Kedokteran
7	Universitas Kristen Duta Wacana	UKDW	Arsitektur
8	Universitas Kristen Duta Wacana	UKDW	Desain Produk
9	Universitas Kristen Duta Wacana	UKDW	Theologia

Tabel diatas merupakan bentuk hasil pencarian yang dilakukan. Pada tabel tersebut merupakan contoh hasil pencarian universitas yang memiliki kata kunci atau penggalan kata "duta". Semua data yang mempunyai penggalan kata "duta" akan ditampilkan dalam bentuk *grid* dengan nama universitas, singkatan, dan nama program studi. Pada halaman ini juga disediakan *link* untuk kembali ke halaman pencarian universitas.

Pencarian universitas tersebut dapat dilakukan secara lebih *detail* dengan cara mengakses *link* pencarian lanjutan pada Gambar 4 diatas. Berikut adalah contoh halaman pencarian lanjutan tersebut.

Gambar 5 Halaman Pencarian Lanjutan

Halaman ini dapat diakses jika *link* pencarian lanjutan pada Gambar 4 diatas berhasil diakses. Halaman ini digunakan untuk melakukan pencarian universitas secara lebih spesifik. Pencarian ini diolah berdasarkan jenis lembaga seperti universitas, akademi, sekolah tinggi, politeknik, dan institut. Selain itu juga berdasarkan program studi sepeti teknik informatika, sistem informasi, dan sebagainya. Jika program studi yang diinginkan tidak tersedia di halaman ini, pengguna dapat menginputkan nama program studi yang diinginkan melalui *textbox* yang sudah disediakan. Pencarian ini akan diproses setelah jenis kriteria telah dipilih dan tombol cari ditekan.

Tabel 2 Hasil Pencarian Lanjutan

No	Universitas	Singkatan	Program Studi
1	Universitas Kristen Duta Wacana	UKDW	Sistem Informasi
2	Universitas Ahmad Dahlan	UAD	Sistem Informasi
3	Universitas Teknologi Yogyakarta	UTY	Sistem Informasi
4	Universitas Warmasa Mangala	UNWAMA	Sistem Informasi

Tabel 2 diatas merupakan bentuk hasil pencarian yang dilakukan. Pada gambar tersebut merupakan contoh hasil pencarian universitas yang memiliki program studi sistem informasi. Hasil yang akan ditampilkan berupa *grid* dengan nama universitas, singkatan, dan nama program studi. Pada halaman ini juga disediakan *link* untuk kembali ke halaman pencarian universitas.

5. Kesimpulan

Berdasarkan hasil penelitian dan analisa, dapat disimpulkan beberapa hal sebagai berikut :

- Program 1, 2, dan 3 diatas membuktikan bahwa implementasi dengan menggunakan metode MVC dapat diterapkan dalam perancangan situs portal pencarian universitas.
- Berdasarkan Program 1, pencarian multikriteria dengan menggunakan *query* SQL dapat diterapkan untuk melakukan pencarian universitas di Daerah Istimewa Yogyakarta.

Beberapa saran untuk perbaikan program di masa mendatang adalah sebagai berikut :

- Menambah jumlah kriteria pencarian universitas pada pencarian lanjutan.
  Membuat halaman website yang berbasis mobile.

Daftar Pustaka

Pratama, Antonius N. W. *Codeigniter : cara Mudah Membangun Aplikasi PHP*. Jakarta Selatan : Media Kita, 2010.

Griffiths, Adam. *Codeigniter 1.7 Professional Development.* Birmingham. Packt Publishing, 2010

Saputra, Agus. *Trik Kolaborasi Codeigniter dan Jquery Edisi Pertama.* Yogyakarta. Lokomedia, 2011.

http://www.prowebpro.com/articles/mvc.html. 24 Juli 2011

http://troels.dk/db/rdbms. 13 Oktober 2011