

PENGUNAAN MODEL VIEW CONTROLLER (MVC) DALAM PERANCANGAN MODEL APLIKASI WEBGIS UNTUK INFORMASI RENTAN BANJIR DI SEMARANG

Dwati Wismarini¹, Hari Murti²

¹Program Studi Teknik Informatika, ²Program Studi Sistem Informasi
Fakultas Teknologi Informasi, Universitas Stikubank
e-mail: ¹thwismarini@edu.unisbank.ac.id, ²hmurti076@gmail.com

ABSTRAK

Pengembangan Aplikasi Sistem Informasi Geografis ke depan cenderung lebih mengarah kepada aplikasi berbasis WEB. Yang dalam hal ini lebih dikenal dengan WEBGIS. Hal ini dikarenakan terlihat adanya potensi besar dalam kaitannya dengan geo informasi di lingkungan aplikasi di lingkungan jaringan.

Terkait dengan pemanfaatan pengembangan aplikasi WEBGIS yang telah semakin banyak dibahas di berbagai penelitian, terdapat sebuah metode yang dinamakan Model-View-Controller (MVC) untuk pengembangan aplikasi berbasis WEB. MVC ini membagi aplikasi dalam dalam tiga bagian utama yaitu Model kemudian View dan Controller.

Dalam Penelitian ini MVC tersebut akan digunakan dalam perancangan Model prototyping Aplikasi WEBGIS untuk informasi Rentan Banjir di kota Semarang, yang mana dalam penelitian tersebut akan dilakukan desain arsitektur Framework berbentuk Three-tier menggunakan MVC,berikut desain general folder organization. Begitu juga dilakukan rancangan prototyping Script code untuk Model, View dan Controller.

Kata Kunci: WEBGIS, Model-View-Controller(MVC),Three-Tier FrameWork, Rentan Banjir Semarang

1. PENDAHULUAN

Secara umum Sistem Informasi Geografis dikembangkan berdasarkan pada prinsip input/masukan data, manajemen, manipulasi, analisis dan representasi data. Sedangkan Aplikasi Sistem Informasi Geografis tumbuh tidak hanya secara jumlah namun juga dari jenis keragaman aplikasinya. Untuk saat ini dan ke depannya pengembangan aplikasi Sistem Informasi Geografis tersebut mengarah kepada aplikasi berbasis Web yang dikenal dengan WebGis. Hal ini dikarenakan pengembangan aplikasi di lingkungan jaringan telah menunjukkan potensi besar dalam kaitannya dengan geo informasi. Telah banyak penelitian yang membahas mengenai pemanfaatan aplikasi WebGis ini.

Pada sebuah penelitian, aplikasi WebGis ini dimanfaatkan sebagai salah satu alat bantu untuk melakukan dokumentasi, edukasi, penyebaran informasi dan bahkan memungkinkan terjadinya interaksi antar komunitas secara spasial. WebGis ini dinamakan Petakita, di mana dalam Petakita memiliki potensi untuk mensosialisasikan hasil kajian arkeologi.[1].

Di sisi lain, telah dikenal sebuah Metode Model-View-Controller (MVC) untuk pengembangan aplikasi berbasis WEB yang mana aplikasi dapat dibagi-bagi dalam tiga bagian utama yaitu Model, sebagai bagian kode aplikasi yang berhubungan dengan basis data, kemudian View, sebagai bagian kode yang berhubungan dengan tampilan ke pengguna dan Controller, sebagai bagian kode yang menghubungkan antara Model dan View.

Dalam sebuah peelitian untuk pembuatan aplikasi Rencana Studi Fakultas Teknologi Informasi di Universitas SatyaWacana (UKSW) dengan menggunakan MVC, dapat disimpulkan bahwa MVC dapat membantu dalam pengembangan, pemeriksaan kesalahan dan perawatan aplikasi tersebut. Hal ini dikarenakan perancangan untuk model adalah terpenting dan menjadi dasar pembuatan aplikasi selanjutnya, sedangkan perancangan view yang dibantu oleh JQuery menjadikan tampilan menjadi lebih interaktif dan menarik sedangkan perancangan controller menjadi penghubung antara bagian view dan model yang menentukan alur aplikasi berjalan.[2].

Berdasarkan uraian tersebut, maka topik Penggunaan Model View Controller (MVC) dalam Perancangan Model Aplikasi Webgis Untuk Informasi Rentan Banjir Di Semarang ini dipilih dengan tujuan dapat menghasilkan sebuah model aplikasi WEBGIS yang selain dapat memberikan informasi yang akurat dengan tampilan interaktif dan menarik juga dapat memberikan keuntungan saat pembuatan sistem, pemeliharaan sistem, pun pengembangan sistem selanjutnya. Hal ini dikarenakan bagian-bagian (model, view dan controller) dari sistem saling terpisah, sehingga ketika satu bagian diubah, maka tidak mempengaruhi bagian yang lain.

2. TINJAUAN PUSTAKA

2.1 Kerentanan Banjir

Kerentanan banjir didefinisikan sebagai ukuran risiko yang dikombinasikan dengan tingkat kemampuan sosial dan ekonomi untuk mengatasi peristiwa banjir. Definisi ini mengacu pada karakteristik pribadi atau kelompok dalam hal kemampuan mereka untuk mengantisipasi, dan mengatasi dampak banjir [3][4][5].

2.2 WEBGIS

WebGIS adalah suatu bentuk proses untuk mendistribusikan, mempublikasikan, mengintegrasikan, berkomunikasi dan menyediakan informasi yang berbentuk teks, peta digital serta menjalankan fungsi-fungsi analisis dan query, yang terkait dengan aplikasi GIS atau pemetaan digital dengan menggunakan jaringan internet.[6]

Dalam hal ini, WebGIS dirasakan sangat diperlukan untuk melakukan berbagai proyek, pengembangan dan perencanaan wilayah ataupun manajemen sumber daya alam. Hal tersebut dikarenakan pengguna masih kurang tercukupi dengan adanya penyebaran data spasial yang dilakukan dengan menggunakan media yang telah ada seperti media cetak (peta), CD-Rom, dan media penyimpanan lainnya. Adapun melalui media tersebut pengguna harus datang dan melihat langsung data pada tempatnya. Mobilitas dan kecepatan dalam memperoleh informasi mengenai data dengan media-media lain tanpa menggunakan WEBGIS masih terasa kurang.[7]

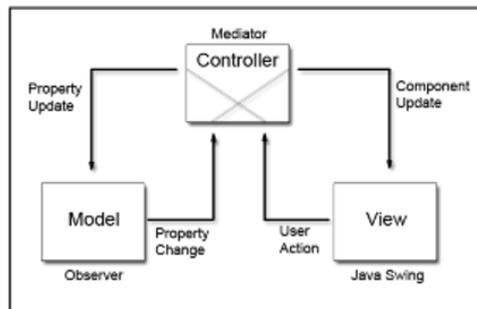
2.3 Pemodelan Aplikasi WEB

Pemodelan Aplikasi WEB digunakan untuk mengurangi kompleksitas, mendokumentasikan keputusan rancangan dan memfasilitasi komunikasi di dalam tim proyek. Pemodelan tersebut bertujuan menyediakan suatu spesifikasi dari sistem yang akan dibangun dengan derajat detail yang cukup bagi implementasi sistem tersebut dan hasil dari proses pemodelan tersebut adalah model yang merepresentasikan aspek-aspek yang relevan dari sistem dalam bentuk yang disederhanakan dan juga comprehensible (dapat dipahami).[8]

2.4 Model View Controller (MVC)

Arsitektur Model View Controller (MVC) adalah sebuah pola yang terbukti membangun proyek secara lebih efektif. Hal ini dilakukan dengan memilah komponen antara Model, View dan Controller pada bagian-bagian dalam proyek. Adapun bagian-bagian dalam MVC adalah sebagai berikut : (1) Model merupakan layer yang merepresentasikan data untuk digunakan oleh aplikasi sebagaimana proses bisnis yang diasosiasikan terhadapnya. Dengan memilahnya sebagai bagian terpisah, seperti enampungan data, persistence, serta roses manipulasi, terpisah dari bagian lain aplikasi. (2) View merupakan layer yang mengandung keseluruhan detail dari implementasi user interface. Di sini, komponen grafis menyediakan representasi proses internal aplikasi dan menuntun alur interaksi user terhadap aplikasi (3) Controller merupakan layer yang menyediakan detail alur proram dan transisi layer dan juga bertanggung jawab akan penampungan events yang dibuat oleh user dari View dan melakukan update terhadap komponen Model menggunakan data yang dimasukkan oleh user.[9]

Adapun ketiga komponen (Model, View, Controller) tersebut merupakan bagian-bagian tersendiri yang terpisah satu sama lain namun tetap berinteraksi diantaranya.



Gambar 1. Arsitektur Model View Controller (MVC)

Gambar 1 tersebut menunjukkan tiga (3) komponen yang terdapat dalam konsep MVC dan interaksi yang terjadi.[10]

3. METODE PENELITIAN

3.1. Obyek Penelitian

Obyek penelitian dari penelitian ini adalah informasi Rentan Banjir di Semarang

3.2. Teknik Pengumpulan Data

Pengumpulan data dimaksudkan agar emendapatkan bahan-bahan yang relevan, akurat dan reliable. Maka teknik pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut :

a. Dokumentasi

Mengumpulkan dokumen-dokumen data-data baik spasial dan non spasial yang berhubungan dengan obyek penelitian, yaitu rentan banjir di Semarang.

b. Observasi

Dengan melakukan pengamatan dan pencatatan secara sistematis tentang hal-hal yang berhubungan dengan basis data dokumen-dokumen baik data spasial dan non spasial dari rentan banjir di Semarang.

c. Studi Pustaka

Dengan pengumpulan data dari bahan-bahan referensi, arsip, dan dokumen yang berhubungan dengan permasalahan dalam penelitian ini.

3.3. Metode Pengembangan

Penelitian ini menggunakan model *prototyping*. Di dalam model ini sistem dirancang dan dibangun secara bertahap dan untuk setiap tahap pengembangan dilakukan percobaan-percobaan untuk melihat apakah sistem sudah bekerja sesuai dengan yang diinginkan. Tahap-tahap *prototyping* adalah:

a. Analisa

Pada tahap ini dilakukan analisa tentang masalah penelitian dan menentukan pemecahan masalah yang tepat untuk menyelesaikannya.

b. Disain

Pada tahap ini dibangun rancangan sistem dengan beberapa diagram bantu seperti Data Flow Diagram.

c. Prototype

Pada tahap ini dibangun aplikasi berbasis web yang sesuai dengan disain dan kebutuhan system juga menggunakan arsitektur MVC (Model-View-Controller) dengan membuat data model untuk merepresentasikan informasi dari database, view untuk menampilkan data dan controller yang akan menggabungkan keduanya bersama-sama.

d. Pengujian

Pada tahap ini dilakukan pengujian pada pustaka fungsi yang sudah dibangun.

e. Evaluasi

Pada tahap ini dilakukan evaluasi apakah performa aplikasi sudah sesuai dengan yang diharapkan, apabila belum maka dilakukan penyesuaian-penyesuaian secukupnya.

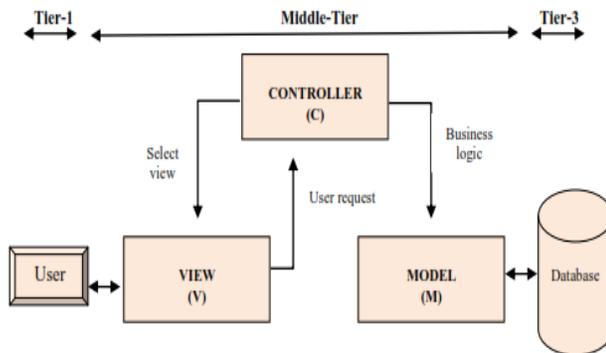
f. Penyesuaian

Tahap ini dilakukan apabila pada evaluasi performa aplikasi kurang memadai dan dibutuhkan perbaikan, tahap ini melakukan penyesuaian dan perbaikan pada aplikasi sesuai dengan kebutuhan.

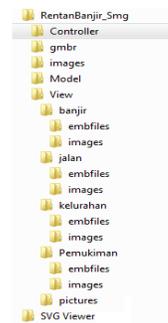
4. HASIL DAN PEMBAHASAN

4.1 Arsitektur Sistem

Di dalam Aplikasi Webgis Untuk Informasi Rentan Banjir Di Semarang ini, telah diputuskan untuk mendesign arsitektur dan operasi dari framework. yang mana merupakan aplikasi framework Three-tier yang meliputi tiga bagian utama yaitu users, MVC components dan database. Tier pertama terdiri dari hanya beberapa user. User dapat mengirim permintaan dan menerima respon oleh komponen-komponen View pada middle-tier. Tier kedua adalah middle-tier yang terdiri dari komponen-komponen yang dinamakan Model, View dan Controller. Komponen-komponen ini dapat berkomunikasi dan memproses data satu sama lain. Terakhir, adalah Tier ketiga terdiri hanya database dimana data disimpan secara permanen. Pada Gambar 2 ditunjukkan bagaimana data mengalir melewati aplikasi menggunakan MVC, sedangkan gambar 3 menunjukkan organisasi folder secara umum untuk aplikasi WebGis Informasi Rentan Banjir di Semarang.



Gambar 2. Framework Architecture



Gambar 3. General Folder Organization

Adapun operasi-operasi dari Arsitektur tersebut dapat dideskripsikan dalam langkah-langkah berikut ini :

- a. Input user diterima oleh komponen “View” melalui Graphical User Interface (GUI). Dalam sebuah client GUI stand alone, interaksi user dapat dengan cara klik button atau pemilihan menu.
- b. “Controller” memproses permintaan-permintaan user. Berdasarkan permintaan user, Controller memanggil metoda-metoda dalam View dan Model untuk menyelesaikan aksi permintaan. Sebuah Controller GUI murni menerima input dari user dan menginstruksikan model dan port view untuk melakukan aksi berdasarkan input tersebut. Controller tersebut menyesuaikan permintaan pada Model. Model tersebut mewakili atau merangkum kondisi dan logika bussines aplikasi. Logika dan kondisi bussines aplikasi tersebut menangkap tidak hanya kondisi dari sebuah proses atau system, tetapi juga bagaimana sistem tersebut bekerja.
- c. View tersebut merespon untuk output dari Model. View GUI murni memberikan ke sebuah model dan menyediakan content-content model pada permukaan tampilan.

4.2 Implementasi Arsitektur MVC

Arsitektur Model View Controller (MVC) dalam perancangan Model Aplikasi Webgis Untuk Informasi Rentan Banjir Di Semarang diimplementasikan dalam setiap modul dalam aplikasi Webgis tersebut. Setiap modul memiliki satu model, satu controller, dan beberapa view. Berikut ini akan dijelaskan implementasi arsitektur MVC pada component Aplikasi.

a. Model

Model perlu dibuat sebelum mulai memisahkan view untuk mendapatkan informasi yang akan ditampilkan. Bagian front-end digunakan untuk pengguna umum dan khusus, sedangkan bagian back-end digunakan untuk administrator. Langkah yang diperlukan dalam mengoperasikan Model adalah pertama mengimpor library model data, kemudian mendeklarasikan class model dengan sebuah ama sebagai ekstensi dari Jmodel. Nama class model harus terdiri dari nama komponen lalu diikuti oleh kaa Model dan nama file model. Terdapat sebuah fungsi yang berguna mengecek dan melihat apakah data telah diload berdasarkan Idnya. Di dalamnya juga terdapat perintah untuk meakukan query terhadap database. Terdapat juga sebuah metode ntuk mengenkapsulasi nama dari elemen query yang menjamin bahwa nama elemen dienkapsulasi dalam delimiter yang benar. Hasil query di-load dengan menggunakan sebuah metode loadObject().

```

<?php
defined('_JEXEC') or die('Restricted Access');
jimport('joomla.application.component.model');

<form name="form" method="post" action="<?php $PHP_SELF ?>">
<table width="100%" border="1" cellspacing="0" cellpadding="0" style="border:solid 1px #999999">
<tr>
<td height="32" colspan="9" bgcolor="#FFFFFF" class="style2" scope="col">
<span class="style3">.</span><select name="status" class="style2">
<option-- Cari Data Status --</option>
<option value="Sangat Rawan">Sangat Rawan</option>
<option value="Rawan">Rawan</option>
<option value="Tidak Rawan">Tidak Rawan</option>
</select>
<span class="style3">.</span>
<input name="kelurahan" type="text" class="style2" value="Kelurahan" />
<span class="style3">.</span>
<input name="Submit" type="submit" class="style2" value="Cari" />
<span class="style3">
..... </span>
<span class="style4">
<?php if ($_POST['status'] and $_POST['kelurahan']) {
include ".././././koneksi.php";
$query = mysql_query ("select * from th3 where Keterangan = '$_POST[status]' and Kelurahan = '$_POST[kelurahan]'");
}
else
(include ".././././koneksi.php";
$query = mysql_query ("select * from th3");)
$a = mysql_num_rows ($query);
echo "$a";?>
Data ditemukan</span><span class="style3">.....<a href="cetak.php">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Setting Data Spasial</title>
<style type="text/css">
<!--
.style2 {font-size: 12px; font-family: Arial, Helvetica, sans-serif;}
body {
margin-left: 0px;
margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;
}
.style3 {color: #FFFFFF}
.style4 {color: #0066FF}
-->
</style>
</head>

<body onload="javascript:window.print()">
<table width="100%" border="1" cellpadding="0" cellspacing="0" style="border:solid 1px #999999">
<tr>
<th width="2%" bgcolor="#F0F0F0" class="style2" scope="col">No</th>
<th width="14%" bgcolor="#F0F0F0" class="style2" scope="col">Status</th>
<th width="17%" bgcolor="#F0F0F0" class="style2" scope="col">Keterangan</th>
<th width="19%" bgcolor="#F0F0F0" class="style2" scope="col">Curah Hujan </th>
<th width="13%" bgcolor="#F0F0F0" class="style2" scope="col">Guna Lahan </th>
<th width="14%" bgcolor="#F0F0F0" class="style2" scope="col">Jenis Tanah </th>
<th width="12%" bgcolor="#F0F0F0" class="style2" scope="col">Kelurahan</th>
<th width="9%" bgcolor="#F0F0F0" class="style2" scope="col">Kecamatan</th>
</tr>
<?php
if ($_REQUEST['status'] and $_REQUEST['kelurahan']) {
include "../..//koneksi.php";
$query = mysql_query ("select * from th3 where Keterangan ='$_POST[status]' and Kelurahan ='$_POST[kelurahan]'");
} else
{include "../..//koneksi.php";
$query = mysql_query ("select * from th3");}
while ($a=mysql_fetch_array($query)) {?>
<tr>
<td class="style2"><div align="center"><?php echo "$a[MVSVGNo]"; ?></div> </td>
<td class="style2"><div align="center"><?php echo "$a[Keterangan]"; ?></div></td>
<td class="style2"><div align="center"><?php echo "$a[Keterangan]"; ?></div></td>
<td class="style2"><?php echo "$a[Tingkat_ch]"; ?></td>
<td class="style2"><?php echo "$a[Keterang_1]"; ?></td>
<td class="style2"><div align="center"><?php echo "$a[Jns_tanah]"; ?></div> </td>
<td class="style2"><?php echo "$a[Kelurahan]"; ?></td>
<td class="style2"><?php echo "$a[Kecamatan]"; ?></td>
</tr>
<?php } ?>
</table>
</body>
</html>

```

Gambar 5. Script Kode View

c. Controller

Controller bekerja berdasarkan task apa yang telah diminta dan berdasarkan task tersebut maka controller mengambil data dari model dan mengirimkan data dari model tersebut ke view. Jadi tugas controller adalah bekerja berdasarkan inputan user yang kemudian dikenal dengan nama task, memanggil metode pada model untuk memanipulasi data pada tabel, dan mengirimkan data dari model ke View untuk ditampilkan pada browser. Dalam hal ini script kode sebuah controller akan mengendalikan alur program.

Nama dari class controller harus terdiri dari nama komponen lalu diikuti oleh kata Controller. Controller dibuat dengan mewarisi model dari class JController. Untuk itu harus mengimpor classnya. Fungsi display() dipanggil secara default ketika controller dieksekusi, dengan alur yaitu pertama mendapatkan sebuah objek view dengan melewati nama view, dan menuliskan ke dalam fungsi member getView(), kemudian memilih model untuk view dengan metode getModel() dan mengaturnya dalam view dengan metode setModel(), terakhir menampilkan layout view yang diinginkan.

```

defined('_JEXEC') or die('Restricted Access');
jimport('joomla.application.component.controller');
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.d
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Setting Data Spasial</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;
}
.style1 {
font-family: Arial, Helvetica, sans-serif;
font-size: 12px;
}
.style2 {color: #FFFFFF;}
-->
</style></head>

<body>
<form id="form1" name="form1" method="post" action="update.php">
<?php
include ".../.../koneksi.php";
$query = mysql_query ("select * from th3 where MVSIGNO='$_REQUEST[no]'");
while ($a=mysql_fetch_array($query)) {?>
<table width="100%" border="1" cellspacing="0" cellpadding="0">
<tr>
<td width="16%" height="35" bgcolor="#F0F0F0" scope="col"><span class="style1">No</span></td>
<td width="84%" scope="col"><div align="left">
<span class="style2">...</span>
<input name="MVSIGNO" type="text" class="style1" readonly="yes" value="<?php echo $a[MVSIGNO]; ?>" style="width: 100%; height: 20px; border: 1px solid black;" />
<input name="thirec" type="hidden" class="style1" readonly="yes" value="<?php echo $a[thirec]; ?>" style="width: 100%; height: 20px; border: 1px solid black;" />
</div></td>
</tr>
<tr>
<td bgcolor="#F0F0F0" height="35"><span class="style1">Keterangan</span></td>
<td <span class="style2">...</span>
<input name="Keterangan" type="text" class="style1" value="<?php echo $a[Keterangan]; ?>" readonly="yes" style="width: 100%; height: 20px; border: 1px solid black;" />
</td>
</tr>
<tr>
<td bgcolor="#F0F0F0" height="35"><span class="style1">Kelurahan</span></td>
<td <span class="style2">...</span>
<input name="Kelurahan" type="text" class="style1" value="<?php echo $a[Kelurahan]; ?>" />
</td>
</tr>
<tr>
<td bgcolor="#F0F0F0" height="35"><span class="style1">Curah Hujan</span></td>
<td <span class="style2">...</span>
<input name="Tingkat_ch" type="text" class="style1" value="<?php echo $a[Tingkat_ch]; ?>" />
</td>
</tr>
<tr>
<td bgcolor="#F0F0F0" height="35"><span class="style1">Guna Lahan</span></td>
<td <span class="style2">...</span>
<input name="Keterang_1" type="text" class="style1" value="<?php echo $a[Keterang_1]; ?>" />
</td>
</tr>
<tr>
<td bgcolor="#F0F0F0" height="35"><span class="style1">Jenis Tanah</span></td>
<td <span class="style2">...</span>
<input name="Jns_tanah" type="text" class="style1" value="<?php echo $a[Jns_tanah]; ?>" />
</td>
</tr>
<tr>
<td bgcolor="#F0F0F0" height="35"><span class="style1">Keterangan</span></td>
<td <span class="style2">...</span>
<input name="Keterangan" type="text" class="style1" value="<?php echo $a[Keterangan]; ?>" />
</td>
</tr>
<tr>
<td bgcolor="#F0F0F0" height="35"><span class="style1">Kecamatan</span></td>
<td <span class="style2">...</span>
<input name="Kecamatan" type="text" class="style1" value="<?php echo $a[Kecamatan]; ?>" />
</td>
</tr>
<tr>
<td colspan="2" style="text-align: center; padding-top: 10px;">
<input type="submit" name="Submit" value="Ubah" />
<input name="button" type="button" onclick="self.history.back()" value="Batal" />
</td>
</tr>
</table>
<?php ?>
</form>
</body>
</html>

```

Gambar 6. Kode Script Kontroler

5. KESIMPULAN

Berdasarkan pembahasan yang telah dilakukan sebelumnya, maka dapat diambil beberapa kesimpulan sebagai berikut:

- a. Penerapan arsitektur model-view-controller (MVC) dalam pemodelan aplikasi WEBGIS untuk informasi Rentan Banjir di kota Semarang ini dapat meningkatkan modularitas dan reusabilitas dari system. Hal ini dimungkinkan karena source code menjadi lebih rapi dan pemisahan antara logika bisnis dan antarmuka pengguna yang lebih eksplisit.
- b. Dengan menggunakan arsitektur MVC ini kompleksitas dari kode dalam perangkat lunak dapat dikurangi secara signifikan. Dengan demikian, juga dapat meningkatkan fleksibilitas system perangkat lunak yang dalam hal ini adalah WEBGIS.
- c. Prototyping merupakan metode pengembangan perangkat lunak yang cocok dalam pengembangan aplikasi yang tidak memiliki kepastian terhadap efisiensi algoritma. Hal ini dikarenakan kemampuan penyesuaian harus dilakukan dengan cara berinteraksi antara manusia dengan mesin, sehingga dapat dilakukan perbaikan dengan cepat.

6. SARAN

Penelitian yang dilakukan terhadap topik ini masih sangat banyak kendala dan banyak kekurangan di sana sini. Maka saran-saran yang dapat diberikan untuk penelitian berikutnya adalah :

- a. Penelitian ini hanya mengulas tentang langkah penerapan model-view-controller (MVC), berdasarkan desain framework, belum mengulas langkah pembuatan dalam Framework itu sendiri serta analisisnya, maka untuk penelitian selanjutnya dapatlah dideskripsikan ulasan langkah-langkah dan analisis penerapan MVC dengan sebuah framework tertentu.
- b. Perlu diujicobakan untuk Framework lain yang menggunakan arsitektur model-view-controller (MVC) dan dilakukan langkah analisis lebih detail tentang penggunaan Framework lain tersebut.
- c. Perlu juga diujicobakan dengan mengkolaborasi pemanfaatan suatu Framework dengan teknologi-teknologi WEBGIS yang lain.

DAFTAR PUSTAKA

- [1] Sumarno, 2014, *Pemanfaatan WebGIS "Petakita" untuk Dokumentasi dan Sosialisasi Objek Arkeologi*, Jurnal Itenas Rekayasa, LPPM Itenas, No.1, Vol. XVIII, Januari 2014
- [2] Touselak, M. 2012. *Aplikasi Rencana Studi Menggunakan Metode Model View Controller (Studi Kasus: FTI UKSW)*. Program Studi Teknik Informatika Fakultas Teknologi Informasi Universitas Kristen SatyaWacana
- [3] Cannon, T., 1994, 'Vulnerability analysis and the explanation of "natural" disasters', in A. Varley (ed.) *Disasters, development and the environment*, Chichester: John Wiley.
- [4] Scoones, I., 1999, *Sustainable Rural Livelihoods, A Framework for Analysis*, IDS Working Paper 72, Brighton: University of Sussex.
- [5] Smith, Keith., 1996, *Environmental Hazards: Assessing Risk and Reducing Disaster*. London; New York: Routledge.
- [6] Prahasta. E., 2007. *Membangun Aplikasi Web-based GIS dengan Mapserver*. Bandung: Informatika.
- [7] Rabbasa dan Setiawan, 2006, *Aplikasi Open Source untuk Pemetaan Online*, Bogor : Seameo Biostrop.
- [8] Solikin, I., 2014, *Perancangan Sistem Informasi Penjualan Berbasis Framework Model View Controller (MVC) Pada PT Thamrin Brother Cabang Oki*. Jurnal Media Informatika dan Komputer. 4 (1), pp. 1-
- [9] J.E.N.I., 2018, *Pengenalan Arsitektur MVC.*, <http://74.125.153.132/search?q=cache:pPK0WuGfXoJ:dahlan.unimal.ac.id/files/modul-web/JENI-Web%2520Programming-Bab%25207-MVC%2520Intro.pdf+bab+7+pengenalan+arsitektur+mvc&cd=1&hl=id&ct=clnk&gl=id>, diakses tanggal 4 November 2018.
- [10] Wiharto Y., Irawan A., 2016, *Perancangan Sistem Perpustakaan Menggunakan Model View Controller (MVC) dengan metode Object Modelling Technique (OMT) pada Sekolah Menengah Pertama Negeri 44 Palembang*, Jurnal Teknologi Informasi Program Studi Teknik Informatika dan Sistem Informasi, Universitas Bunda Mulia, Volume 12, Nomor 1, Juni 2016, ISSN : 1979 -1496