

Pembangunan Sistem Informasi Zona Potensi Sumber Daya Kelautan Kabupaten Gunungkidul Berbasis HMVC Menggunakan *Google Maps* API dan JSON

Yerymia Alfa Susetyo¹, Pratyaksa Ocsa N. Saian², Ramos Somya³

Fakultas Teknologi Informasi

Universitas Kristen Satya Wacana

Email: ¹yerymia.alfa@uksw.edu, ²pratyaksa.ocsan@uksw.edu, ³ramos.somya@uksw.edu

Abstrak - Indonesia dikenal sebagai negara maritim atau kepulauan terbesar di dunia dengan dua per tiga dari luas wilayah Indonesia adalah laut. Sumber daya kelautan dan perikanan Indonesia diperkirakan bernilai US\$ 136,5 milyar. Guna meningkatkan nilai ekonomi sumber daya kelautan perlu dirumuskan strategi pengelolaan dengan mengintegrasikan teknologi informasi. Adapun tujuan dari penelitian ini adalah perancangan Sistem Informasi Geografi Zona Potensi Kelautan daerah Kabupaten Gunungkidul, Daerah Istimewa Yogyakarta (DIY) berbasis web sehingga memudahkan pemerintah setempat dalam mengelola informasi tentang potensi pariwisata, pertanian, dan perikanan, dan memudahkan masyarakat umum dalam mencari informasi tentang potensi kelautan di daerah Kabupaten Gunungkidul. Hasil dari penelitian ini adalah web sistem informasi geografi zona potensi kelautan dengan framework CodeIgniter dengan teknologi HMVC (Hierarchical Model View Controller) dan menggunakan servis dari Google Maps API. Aplikasi ini menggunakan format JSON dalam pertukaran data karena memudahkan pengembangan dalam mengembangkan aplikasi ke lintas platform.

Keywords: CodeIgniter, HMVC, Google Maps API, Sumber Daya Kelautan, JSON

I. PENDAHULUAN

Indonesia adalah suatu negara maritim atau kepulauan terbesar di dunia, dimana dua per tiga dari luas wilayah Indonesia adalah laut. Indonesia memiliki potensi sumber daya kelautan yang begitu melimpah, sehingga Indonesia ditargetkan menjadi poros maritim dunia [1]. Sumber daya kelautan dan perikanan Indonesia diperkirakan bernilai US\$ 136,5 milyar yang mencakup berbagai sektor seperti: perikanan, sumber daya wilayah pesisir, bioteknologi, pariwisata, dan migas [2].

Provinsi Daerah Istimewa Yogyakarta (DIY) adalah provinsi yang berbatasan pada sebelah selatan dengan Samudera Hindia. Panjang garis pantai Provinsi DIY mencapai 113 KM yang secara administratif masuk kedalam tiga wilayah kabupaten, yaitu Gunungkidul, Bantul dan Kulon Progo [2]. Wilayah pesisir Daerah Istimewa Yogyakarta (DIY) memiliki sumber daya kelautan bernilai ekonomi penting, seperti: ikan pelagis besar, tuna, cakalang, marlin, lamadang, udang, lobster, terumbu karang dan jasa pariwisata, yang strategis dalam perdagangan lokal, nasional, regional dan internasional [2].

Guna meningkatkan nilai ekonomi sumber daya kelautan secara berkelanjutan, perlu dirumuskan strategi pengelolaan secara tepat dengan mengintegrasikan teknologi informasi pada setiap kepentingan dalam keseimbangan aspek ekonomi, ekologis, sosial, antarsektor, dan segenap pelaku pembangunan. Penyusunan strategi pemanfaatan sumber daya pesisir harus didasarkan pada analisis isu-isu strategis dan permasalahan serta karakteristik wilayah untuk mencapai

sasaran ekonomi, sosial, lingkungan, dan mencegah konflik kepentingan [3].

Salah satu cara untuk meningkatkan sumber daya kelautan adalah dengan dibangunnya sistem informasi geografi, sistem ini berfungsi untuk memetakan zona potensi di tiap kecamatan pesisir pantai Kabupaten Gunungkidul. Dengan adanya pemanfaatan teknologi informasi berupa sistem informasi geografi berbasis *web* dan *mobile* diharapkan mampu memudahkan penyampaian informasi tentang potensi desa kepada masyarakat umum, sistem ini memanfaatkan teknologi berupa servis *Google Maps API*. Penggunaan servis tersebut berguna untuk memudahkan dalam memetakan daerah mana saja yang memiliki potensi yang dapat disajikan kepada masyarakat.

Berdasarkan latar belakang yang ada, maka penelitian ini bertujuan untuk membangun sistem informasi geografi berbasis *web* yang mampu memberikan informasi tentang potensi daerah yang dimiliki. Sistem informasi yang dibutuhkan adalah sistem informasi yang mampu diakses dengan mudah dan fleksibel. Pada penelitian ini dirancang suatu sistem informasi geografi (SIG) berbasis *web* menggunakan framework *CodeIgniter* (HMVC) dengan menerapkan *Google Maps API* dan juga *JavaScript Object Notation* (JSON) supaya dapat dikembangkan ke aplikasi *mobile*.

II. TINJAUAN PUSTAKA

A. HMVC

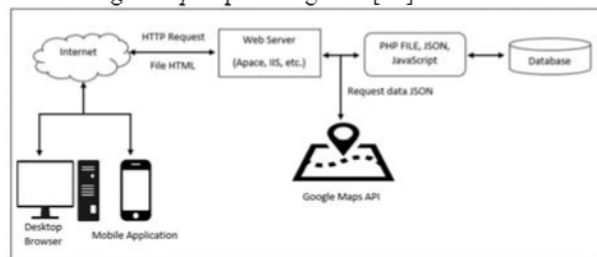
Arsitektur HMVC (*Hierarchical Model View Controller*)

adalah pengembangan dari arsitektur MVC (*Model View Controller*). HMVC adalah susunan dari pola MVC yang tersusun menjadi satu kesatuan aplikasi [7]. Struktur model *Model*, *View*, dan *Controller triad* (segitiga MVC) dibuat berlapis menjadi *hierarchy of parent-child MVC layers* atau hirarki lapisan orang tua-anak MVC [9]. Sehingga pola HMVC mampu memudahkan *programmer* untuk mendokumentasikan dan mengkodekan sistem dalam bentuk modul-modul yang lebih spesifik dan membuat aplikasi menjadi fleksibel dalam pengerjaan.

Arsitektur HMVC terdiri dari beberapa *triad*, dimana setiap *triad* berfungsi secara mandiri satu sama lain. Sebuah *triad* dapat meminta akses ke *triad* lain melalui *controller*-nya sendiri. Hal ini memungkinkan aplikasi dapat didistribusikan ke beberapa lokasi, jika diperlukan. Selain itu, lapisan *triad* MVC memungkinkan pengembangan aplikasi yang lebih kompleks

B. Google Map API

API atau *Application Programming Interface* adalah dokumentasi yang terdiri dari *interface*, fungsi, dan kelas struktur untuk membangun sebuah perangkat lunak. API dapat dikatakan sebagai penghubung suatu aplikasi dengan aplikasi lainnya yang memungkinkan *programmer* menggunakan sistem *function*. Google juga menyediakan layanan *Google Map API* yang memungkinkan para pembangun untuk mengintegrasikan *Google Map* ke dalam *website* masing-masing dengan menambahkan data *point* sendiri. Dengan menggunakan *Google Map API*, *Google Map* dapat ditampilkan pada *website* eksternal. Agar aplikasi *Google Map* dapat muncul di *website* tertentu, diperlukan adanya *API key*. *API key* merupakan kode unik yang digenerasikan oleh google untuk suatu *website* tertentu, agar server *Google Map* dapat mengenali [11].



Gambar 1. Arsitektur Website SIG dengan Google Maps API

Gambar 1 menjelaskan pada mulanya, *browser* dan *mobile application (user)* mengirimkan *request* (melalui jaringan internet) ke *web server*. Kemudian *web server* mengirimkan *request* ke *server* aplikasi (yang dibangun dengan menggunakan pemrograman *script* yang telah tersedia) dan *Google Maps*. Setelah itu *Google Maps* akan membaca *latitude* dan *longitude* yang dikirimkan dari *database* dalam bentuk *file JSON* sehingga dapat membentuk *marker* atau tanda pada peta. Setelah data telah diterima, maka akan dikirimkan ke *web server*, dan akhirnya ke *browser* milik *user* sesuai dengan format tampilan *template*.

C. JavaScript Object Notation (JSON)

JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah untuk dibaca dan ditulis serta mudah diterjemahkan oleh mesin. JSON adalah format

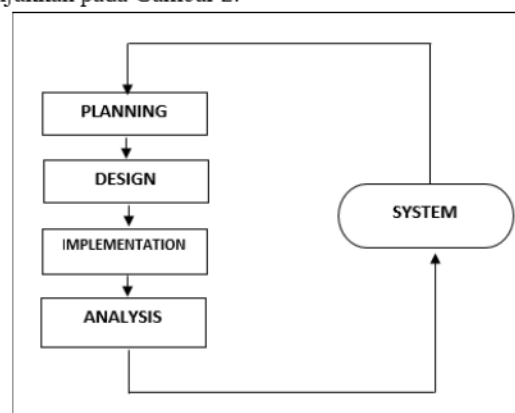
teks yang benar-benar independen namun menggunakan konvensi yang familiar bagi pemrogram yang termasuk dalam keluarga Bahasa C, termasuk C, C ++, C #, Java, JavaScript, Perl dan Python. Oleh karena sifat tersebut menjadikan JSON sebagai bahasa pertukaran data yang ideal [8]. JSON terbuat dari dua struktur utama, yaitu:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*structure*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman moderen mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini [10].

III. METODE PENELITIAN

Metode yang digunakan pada penelitian ini adalah metode *Agile Development*. *Agile Software Development* adalah sebuah grup metodologi pengembangan *software* yang berbasis pada pengembangan iteratif dimana keperluan dan solusi pengembangan *software* berubah terus menerus. Pada *Agile*, kepuasan pelanggan adalah prioritas utama, oleh karena itu *Agile* menekankan pada komunikasi yang intens dengan pelanggan [11]. Tahapan metode *Agile Development* ditunjukkan pada Gambar 2.



Gambar 2. Metode Agile Development

A. Planning

Pada tahap ini dilakukan pengumpulan data. Adapun data primer dan data sekunder yang diperlukan adalah:

- Data sumber daya perikanan pesisir pantai selatan DIY
- Data potensi wisata pesisir pantai selatan DIY
- Data sumber daya pertanian pesisir pantai selatan DIY

Lokasi penelitian adalah kawasan pantai selatan DIY sepanjang 113 km. Di sepanjang pantai 113 km, dilakukan kajian (identifikasi, inventarisasi, dan analisis ekonomi) sumber daya kelautan (perikanan dan jasa-jasa lingkungan) yang memiliki dampak ekonomik luas (*multiplier effect*).

Karakteristik dan spesifik lokasi/kawasan berpotensi ekonomi tinggi yang dijadikan lokasi penelitian, yaitu kawasan pelabuhan perikanan, tempat pendaratan ikan, dan budidaya udang serta wisata (pantai/bahari, kuliner, dan permainan).

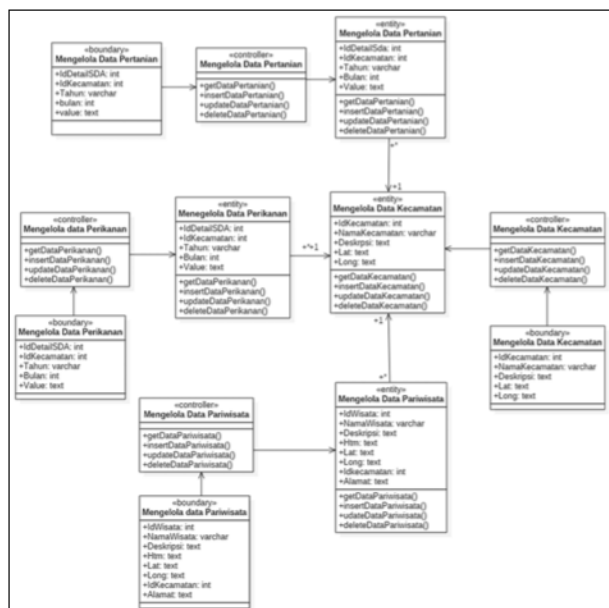
B. Design

Pada tahap ini perancangan Sistem Informasi Geografi Zona Potensi Sumber Daya Kelautan dimulai dengan merancang proses pada sistem menggunakan *Unified Modelling Language (UML)* yang meliputi *Use Case Diagram* dan *Class Diagram*.



Gambar 3. Use Case Diagram Sistem

Gambar 3 adalah *usecase diagram* dari sistem informasi geografi zona potensi kelautan yang menjelaskan proses dan alur sistem. Pada *usecase diagram* ini mempunyai dua aktor yaitu: *Admin* dan *User*. *Admin* memiliki semua akses pada sistem yaitu hak *CRUD (Create, Read, Update, Delete)* pada data potensi pariwisata, pertanian, dan perikanan. Aktor kedua adalah *User* atau pengguna aplikasi, *user* hanya dapat melihat data atau informasi sesuai dengan data potensi.



Gambar 4. Class Diagram Sistem

Gambar 4 merupakan *class diagram* yang menggambarkan *class* yang ada pada sistem, yaitu: *controller*, *model*, dan *view* *class* pada sistem. Fungsi dari masing-masing *class* berbeda-beda. Pertama adalah *boundary class* yang berfungsi sebagai penghubung dan pengelola data dalam *database*. Kedua, *controller class* adalah *class* yang berfungsi untuk mengerjakan fungsi yang berhubungan dengan *model class*. Ketiga adalah *entity class* yang berfungsi menggabungkan dan menampilkan fungsi-fungsi yang terdapat pada *controller* dan *boundary class*. *Class diagram* pada sistem ini mempunyai 4 *controller class* yang masing-masing mempunyai *boundary class* dan *entity class*. *Class* utama yaitu mengelola data kecamatan, mengelola data pariwisata, mengelola data pertanian, dan mengelola data perikanan.

C. Implementation

Pada tahap ini dilakukan implementasi desain ke dalam bentuk program dengan bahasa pemrograman yang telah ditentukan. Aplikasi pada penelitian ini dibangun menggunakan bahasa pemrograman PHP (*Hypertext Preprocessor*) dan diimplementasikan ke dalam *framework CodeIgniter* dengan menggunakan arsitektur *HMVC*.

D. Analysis

Pada tahap ini dilakukan pengujian fitur dengan metode *Blackbox testing*. Pengujian ini dilakukan untuk mengetahui kelemahan atau kesalahan yang ada pada setiap fitur. Sehingga mampu menciptakan aplikasi atau program yang memiliki kinerja yang optimal.

IV. HASIL DAN PEMBAHASAN

Hasil dari penelitian ini adalah sistem informasi geografi yang dapat membantu masyarakat dalam mencari informasi tentang zona potensi kelautan di daerah pesisir pantai Kabupaten Gunungkidul, DIY, yang meliputi potensi pariwisata, pertanian, dan perikanan. Sistem informasi ini memiliki dua pengguna. Pengguna yang pertama adalah *admin*, *admin* memiliki hak akses yang dapat melakukan *CRUD (Create, Read, Update, Delete)* ke semua data potensi, dan pengguna kedua adalah *user*, *user* memiliki hanya hak akses melihat data informasi data potensi.

Pada penelitian ini, sistem informasi geografi dibangun menggunakan aplikasi berbasis *web* menggunakan arsitektur *HMVC (Hierarchical Model, View, Controller)*. Arsitektur ini merupakan evolusi dari arsitektur *MVC (Model, View, Controller)*. Dalam penerapan konsep *MVC*, kode program dapat dibagi menjadi tiga kategori, yaitu: *Model* merupakan bagian dari penanganan yang berhubungan dengan pengolahan atau manipulasi *database*, *view* merupakan bagian yang menangani halaman *UI (User Interface)*, dan *controller* merupakan kumpulan intruksi aksi yang menghubungkan *model* dan *view* [8].

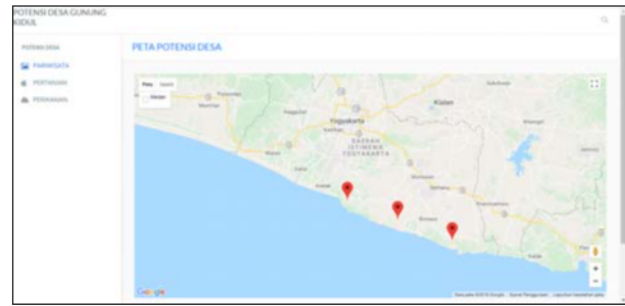
Setiap *triad MVC* berfungsi secara terpisah satu sama lain. *Triad A* dapat meminta akses ke *triad* lain melalui *controller*-nya sendiri. Hal ini memungkinkan aplikasi untuk didistribusikan ke beberapa *triad* jika diperlukan. Selain itu, lapisan *triad MVC* berguna untuk memudahkan pengembangan sistem yang lebih kompleks [9].



Gambar 5. Implementasi Arsitektur HMVC

Gambar 5 menjelaskan implementasi arsitektur HMVC pada penelitian ini. Pada implementasinya modul dibagi menjadi dua, yaitu *admin* dan *peta*, setiap modul memiliki pola MVC sendiri. Pada modul *admin* memiliki *models* yang bernama *AdminModel.php*, pada *file* *AdminModel.php* berisikan tentang perintah *query* CRUD data pada *database*, pada *folder* *views* berisikan *file* yang berfungsi untuk menangani halaman UI (*User Interface*), dan pada *folder* *controller* yang berisikan *file* *Admin.php* yang berisikan fungsi-fungsi untuk menggabungkan *models* dan *views*. Pada modul *peta* memiliki *folder* *models* yang berisikan *file* *PetaModel.php* yang berisikan *query* pemanggilan data potensi dari *database*, pada *folder* *views* berisikan *file* yang berfungsi untuk menampilkan peta dari *Google Maps* dan potensi desa, dan pada *folder* *controller* berisikan fungsi-fungsi untuk menggabungkan *models* dan *views* sehingga dapat menampilkan data potensi desa.

Pada penelitian ini menggunakan servis *Google Maps API* untuk menampilkan peta zona potensi kelautan, tampilan halaman *user* dapat dilihat pada Gambar 6.

Gambar 6. Implementasi *Google Maps API*

Gambar 8 merupakan tampilan awal halaman *user* pada menu potensi pariwisata, pada halaman ini menggunakan servis dari *Google Maps API* untuk memunculkan peta dan *marker* atau tanda untuk menjelaskan potensi apa saja yang ada di kecamatan tersebut.

```
1. function initGoogleMap(){
2. var infowindow = new google.maps.InfoWindow();
3. var map = new
  google.maps.Map(document.getElementById('map-
  canvas'), {
4. zoom: 10,
5. center: new google.maps.LatLng(-7.9931615,
  110.4418458)
6. });
7.
8. function placeMarker(loc) {
9. var latLng = new google.maps.LatLng( loc.lat,
  loc.long);
10. var marker = new google.maps.Marker({
11. position: latLng,
12. map: map});
```

Kode Program 1. *View* menampilkan *Google Maps API*

Kode program 1 baris ke satu sampai baris enam merupakan *function* yang disediakan oleh servis *Google Maps API* untuk menampilkan peta dengan titik tengah sesuai dengan *latitude* dan *longtitude* yang sudah ditentukan pada baris kelima, dan pada baris keempat berfungsi untuk mengatur seberapa besar perbesaran yang diinginkan saat pertama kali memuat halaman. Sedangkan baris delapan sampai baris dua belas merupakan *function* untuk menampilkan *marker* pada peta, pada baris kesembilan berfungsi untuk menyimpan data *latitude* dan *longitude* yang disimpan di *database* dan ditampilkan pada variabel *latLng*. Baris kesepuluh sampai dua belas berfungsi untuk menampilkan *marker* dimana variabel *position* memanggil data yang disimpan pada variabel *latLng* untuk menampilkan dimana letak *marker*.

Gambar 7 merupakan tampilan halaman *user* saat memilih salah satu kecamatan yang memiliki potensi hasil pertanian. Pada Gambar 7 dijelaskan hasil pertanian berupa jagung, padi dan kedelai setiap tahunnya dalam bentuk grafik yang ditampilkan berdasarkan pilihan tahun.



Gambar 7. Halaman Grafik Hasil Pertanian

Gambar 8 menjelaskan tampilan detail dari hasil panen salah satu potensi pertanian. Pada Gambar 7 terdapat menu Jagong, Padi, Kedelai, saat *user* memilih salah satu, maka akan ditampilkan detail hasil pertanian setiap bulannya.

PETA POTENSI DESA

Data Potensi Pertanian

Kembali

Show 10 entriesShowing 1 to 10 of 14 entries

No *	Nama Hasil Pertanian *	Bulan *	Tahun *	Hasil (hektar) *
1	Jagung	November	2016	0
2	Jagung	Desember	2016	0
3	Jagung	Januari	2017	2500
4	Jagung	Februari	2017	0
5	Jagung	Maret	2017	0
6	Jagung	April	2017	50

Gambar 8. Halaman Detail Hasil Pertanian

Gambar 9 merupakan tampilan halaman *user* saat memilih salah satu kecamatan yang memiliki potensi hasil perikanan. Pada Gambar 9 dijelaskan hasil pertanian berupa jagong, padi dan kedelai setiap tahunnya dalam bentuk grafik yang ditampilkan berdasarkan pilihan tahun.



Gambar 9. Halaman Grafik Hasil Perikanan

Gambar 10 menjelaskan tampilan detail dari hasil panen salah satu potensi perikanan. Pada Gambar 10 terdapat menu Lobster, Kakap, Udang, saat *user* memilih salah satu, maka akan ditampilkan detail hasil pertanian setiap bulannya.

PETA POTENSI DESA

Data Potensi Perikanan

Kembali

Show 10 entriesShowing 1 to 10 of 19 entries

No *	Nama Hasil Pertanian *	Bulan *	Tahun *	Hasil (hektar) *
1	Lobster	Mei	2016	25
2	Lobster	Juni	2016	150
3	Lobster	Juli	2016	25
4	Lobster	Agustus	2016	200
5	Lobster	September	2016	125
6	Lobster	Oktober	2016	100

Gambar 10. Halaman Detail Hasil Perikanan

Selain itu dihasilkan juga servis yang dapat diakses oleh *platform* lain dengan menerapkan format pertukaran data JSON. JSON (*JavaScript Object Notation*) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (*generate*) oleh komputer.

JSON merupakan data *array* yang telah dikonversi, setiap *array* dimulai dengan [(kurung kotak buka) dan diakhiri dengan] (kurung kotak tutup). Setiap nilai dipisahkan oleh , (koma).

```
[{"idWisata": "1",
  "namaWisata": "Pantai Sarangan",
  "deskripsi": "pantai Sarangan adalah pantai yang bagus",
  "alamat": "Area Hutan, Ngestirejo, Tanjungsari, Gunung Kidul Regency, Special Region of Yogyakarta 55881",
  "namaFasilitas": "Toilet. Rumah Makan. ",
  "htm": "Rp. 5000",
  "lat": "-8.147654",
  "long": "110.596466"}]
```

Gambar 11. Servis yang dihasilkan oleh JSON

Gambar 11 merupakan hasil dari data JSON, setelah *user* melakukan *request* data detail wisata yang disimpan di *database*. Data yang dilempar pada saat *user* melakukan *request* adalah data *idWisata*, setelah *idWisata* dipanggil maka data yang ditampilkan adalah namaWisata, deskripsi, alamat, namaFasilitas, htm, lat (*latitude*) dan long (*longitude*), data tersebut masih berbentuk *array* pada saat pengambilan data dari *database* dan dikonversi menjadi data JSON sebelum ditampilkan di halaman *user*. Kode program untuk mengambil data detail wisata dapat ditunjukkan pada Kode Program 2.

```

1. public function detailWisata($id){
2.     $var['info'] = $this->PetaModel-
       >getWisata($id);
3.     $var['fasilitas'] = $this->PetaModel-
       >getDetailWisata($id);
4.     $data['foto'] = $this->PetaModel-
       >getFotoWisata($id);
5.     $doto = array();
6.     $dete = array(
7.         'idWisata' => $key->idWisata,
8.         'namaWisata' => $key->namaWisata,
9.         'deskripsi' => $key->deskripsi,
10.        'alamat' => $key->alamat,
11.        'namaFasilitas' => $dutu,
12.        'htm' => $key->htm,
13.        'lat' => $key->lat,
14.        'long' => $key->long );
15.     array_push($doto, $dete);

```

Kode Program 2. Controller menampilkan data JSON

Pada Kode Program 2 merupakan fungsi untuk menampilkan data detail wisata berdasarkan idWisata, baris dua sampai empat merupakan variabel untuk mengambil data berdasarkan id dari *database* melalui PetaModel, setelah data ditampilkan dari *database*, kode program baris enam sampai empat belas data yang diminta dari *database* ditampung dalam *array*, pada kode program baris lima belas data *array* yang ditampung pada variabel *\$dete* ditampilkan menggunakan fungsi *array_push()*; dan ditampung pada variabel *\$doto*. Kemudian data *array* yang tersimpan pada variabel *\$doto* dikonversi menjadi data JSON menggunakan fungsi *json_encode*, fungsi dari *json_encode* adalah untuk menkonversi data, dari data *array* menjadi data JSON, yang nantinya akan ditampilkan menjadi data detail wisata seperti Gambar 11. Pengimplementasian data JSON dapat dilihat pada Gambar 12.



Gambar 12. Tampilan Halaman Detail Wisata

Gambar 12 merupakan tampilan dari halaman detail wisata, pada halaman ini ditampilkan data detail wisata beserta foto yang menjelaskan detail lingkungan potensi pariwisata.

Tahapan terakhir yang dilakukan adalah pengujian sistem. Pengujian menggunakan metode *blackbox testing* merupakan metode pengujian terhadap fungsi-fungsi. Metode *blackbox testing* bertujuan untuk mencari kesalahan atau kegagalan dalam sistem, sehingga dapat diketahui apakah sistem yang dibangun sudah berjalan sesuai harapan dan kebutuhan. Hasil dari pengujian sistem yang telah dilakukan dapat dilihat pada Tabel 1:

Tabel 1. Hasil pengujian *black-box*

Fungsi yang di Uji	Kondisi	Output yang diharapkan	Output yang dihasilkan sistem	Status Pengujian
Login	Username dan password benar	Sukses Login	Sukses Login	valid
	Username dan password salah / tidak diisi	Gagal Login	Gagal Login	valid
Tambah Data Pariwisata	Form diisi dengan benar	Sukses tambah data pariwisata	Sukses tambah data pariwisata	valid
Tambah Data Pertanian	Form diisi dengan benar	Sukses tambah data pertanian	Sukses tambah data pertanian	valid
Tambah Data Perikanan	Form diisi dengan benar	Sukses tambah data perikanan	Sukses tambah data perikanan	valid
Edit Data Pariwisata	Form diisi dengan benar	Sukses edit data pariwisata	Sukses edit data pariwisata	valid
Hapus Data Pariwisata	Pilih salah satu wisata klik button hapus	Sukses hapus data	Sukses hapus data	valid
Edit Data Pertanian	Form diisi dengan benar	Sukses edit data pertanian	Sukses edit data pertanian	valid
Hapus Data Pertanian	Pilih salah satu hasil pertanian klik button hapus	Sukses hapus data pertanian	Sukses hapus data pertanian	valid
Edit Data Perikanan	Form diisi dengan benar	Sukses edit data perikanan	Sukses edit data perikanan	valid
Hapus Data Perikanan	Pilih salah satu hasil perikanan klik button hapus	Sukses hapus data perikanan	Sukses hapus data perikanan	valid
Menampilkan marker Peta Kecamatan	Pilih menu potensi pariwisata	Menampilkan marker peta kecamatan	Menampilkan marker peta kecamatan	valid
Menampilkan marker Potensi Wisata Setiap Kecamatan	Klik marker kecamatan, pilih button potensi pariwisata	Menampilkan marker Potensi Wisata Setiap Kecamatan	Menampilkan marker Potensi Wisata Setiap Kecamatan	valid
Menampilkan Informasi Potensi Wisata	Klik marker potensi wisata, pilih button detail wisata	Menampilkan Informasi Potensi Wisata	Menampilkan Informasi Potensi Wisata	valid
Menampilkan Informasi Potensi Pertanian	Klik marker kecamatan, pilih button grafik pertanian	Menampilkan informasi potensi pertanian	Menampilkan informasi potensi pertanian	valid
Menampilkan Informasi Potensi Perikanan	Klik marker kecamatan, pilih button grafik perikanan	Menampilkan informasi potensi perikanan	Menampilkan informasi potensi perikanan	valid

Berdasarkan pengujian yang dilakukan pada aplikasi sistem informasi ini dapat dilihat status dari setiap pengujian yaitu valid, maka dapat disimpulkan bahwa aplikasi ini berjalan dengan baik dan sesuai yang diharapkan.

V. SIMPULAN DAN SARAN

Berdasarkan penelitian yang dilakukan, maka dapat disimpulkan bahwa:

- Pembangunan Sistem Informasi Geografi Zona Potensi Sumber Daya Kelautan Kabupaten Gunungkidul berbasis web dengan menggunakan *framework CodeIgniter HMVC (Hierarchical Model, View, Controller)* dapat memudahkan pengembang dalam membangun aplikasi, karena pola HMVC yang terstruktur dan tiap modul memiliki pola MVC (*Models, View, Controller*) sendiri, sehingga aplikasi yang diolah menjadi lebih efisien dan terstruktur.
- Penggunaan servis dari *Google Maps API* di aplikasi Sistem Informasi Geografi Zona Potensi Kelautan, berguna untuk menampilkan informasi geografis dalam bentuk peta, jadi akan mempermudah *user* dalam mencari informasi tentang potensi di setiap kecamatan.
- Tipe data JSON digunakan karena format pertukaran data yang ringan, mudah untuk dibaca dan ditulis serta mudah diterjemahkan oleh mesin dan mempermudah pengembang untuk mengembangkan aplikasi SIG Zona Potensi Kelautan ke lintas *platform*.

Untuk penelitian dan pengembangan lebih lanjut, bisa ditambahkan variabel potensi desa yang lain, sehingga mampu membantu masyarakat sekitar dalam mendapatkan informasi tentang potensi desa Kabupaten Gunungkidul. Selain itu dapat dikembangkan dari segi servis *Google Maps API* dengan ditambahkan garis *polygon* untuk menampilkan lokasi potensi yang dipilih.

DAFTAR PUSTAKA

- [1] Subardjo, Petrus. 2015. *Uji Kerawanan Terhadap Tsunami Dengan Sistem Informasi Geografis (SIG) Di Pesisir Kecamatan Kretak Kabupaten Bantul Yogyakarta*. Jurnal Kelautan Tropis, Volume 18(2), hal. 82-97
- [2] Sahubawa, Latif. 2015. *Kajian Sebaran Potensi Ekonomi Sumber Daya Kelautan di Pantai Selatan Daerah Istimewa Yogyakarta Sebagai Upaya Percepatan Investasi*. Jurnal Teknosains, Volume 4, hal. 101-198
- [3] Prasetyo S.Y.J.P., Hasiholan B., dan Hartomo K.D., 2012, *Updated Pranata Mangsa: Recombination of Local Knowledge and Agro Meteorology using Fuzzy Logic for Determining Planting Pattern*, IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 6, No 2, November 2012 ISSN (Online): 1694-0814
- [4] Kharistiani, Erna., Aribowo, Eko. 2013. *Sistem Informasi Geografis Pemetaan Potensi Sma/Smk Berbasis Web (Studi Kasus: Kabupaten Kebumen)*. Jurnal Sarjana Teknik Informatika Volume 1 Nomor 1, Juni 2013, e-ISSN: 2338-5197
- [5] Rahayudi, Yulian, dkk. 2016. *Rancang Bangun Sistem Informasi Geografis Pelaksanaan dan Pengawasan Kegiatan Perkebunan Kelapa Sawit Di PT. Wawasan Kebun Nusantara Kecamatan Seluas. Pontianak*.
- [6] Faisal, Abdul N. 2016. *Sistem Informasi Geografis Berbasis Web Untuk Penyebaran Fasilitas Umum Di Kabupaten Klaten*. Surakarta. Universitas Muhammadiyah Surakarta
- [7] Tanjung, Maulana Iqbal. 2011. *Analisis dan Perancangan Sistem Informasi Berbasis Website Menggunakan Arsitektur MVC dengan Framework Codeigniter*. Yogyakarta: STMIK AMIKOM.
- [8] Hadi, Diki Alfarabi. 2016. *Pengertian dan Cara Menggunakan CodeIgniter*, <https://www.malasngoding.com/pengertian-dan-cara-menggunakan-codeigniter/>. Diakses pada tanggal 5 Juli 2018
- [9] Cogan, Barry. 2010. *HMVC: An Introduction and Application*, <https://code.tutsplus.com/tutorials/hmvc-an-introduction-and-application--net-11850>. Diakses pada tanggal 5 Juli 2018.
- [10] JSON. 2016. *Introducing JSON*. Diambil 5 Mei 2018, dari <https://www.json.org>
- [11] Google Inc. 2016. *Google Maps API*. <https://developers.google.com/maps>. Diakses pada tanggal 5 Mei 2018

- [12] Nugraha, Firman. 2010. *Metode Pengembangan Aplikasi Secara Cepat dan Adaptif Menggunakan Agile Software Development (Online)*, (<https://teknojurnal.com/metode-pengembangan-aplikasi-secara-cepat-dan-adaptif-menggunakan-agile-software-development/>, diakses pada 22 Mei 2018).