

## 1. Pendahuluan

Pendidikan adalah usaha sadar untuk menyiapkan peserta didik melalui kegiatan bimbingan, pengajaran, dan latihan bagi peranannya di masa yang akan datang. Sistem pendidikan formal berstruktur hirarkis dan memiliki kelas yang berurutan dari sekolah dasar sampai perguruan tinggi yang juga dijalankan dengan bermacam-macam program dan lembaga khusus yang digunakan untuk pelatihan teknis dan profesional [1]. Proses pendidikan formal dilaksanakan di lembaga pendidikan yang merupakan tempat atau sistem sosial dimana terjadi transfer ilmu pengetahuan dan berjalannya suatu proses pendidikan. Lembaga pendidikan formal terdiri atas 3 jenjang yaitu, pendidikan dasar (SD, MI, SMP, dan MTs), pendidikan menengah (SMA, MA, SMK, MAK) dan pendidikan tinggi (akademi, politeknik, sekolah tinggi, institut dan universitas) [2].

Kota Salatiga dikenal sebagai kota pendidikan yang memiliki 126 sekolah untuk pendidikan dasar, 30 sekolah untuk pendidikan menengah, dan 5 perguruan tinggi [3]. Orangtua maupun peserta didik yang berkeinginan mencari lembaga pendidikan di kota Salatiga, terkadang hanya berbekal nama jalan atau arah letak bangunan sehingga hanya mendapat perkiraan bayangan saja alih – alih sebuah peta yang dilengkapi dengan petunjuk. Hal ini membuat peserta didik harus melakukan pencarian informasi dengan mendatangi langsung lokasi lembaga pendidikan satu per satu yang membutuhkan waktu lama sehingga tidak efisien. Berdasarkan latar belakang yang ada, dibuat perancangan sistem informasi pencarian lembaga pendidikan di kota Salatiga berbasis *web*.

Sistem informasi pencarian merupakan salah satu cara melakukan pencarian dengan memasukkan kata kunci yang sesuai dengan representasi informasi yang ingin dicari. Setelah sistem menampilkan hasil pencarian, *User* dapat mengikuti tautan-tautan yang diberikan untuk mendapatkan informasi yang diinginkan [4]. Perancangan sistem ini bertujuan untuk membangun sistem informasi pencarian lembaga pendidikan di kota Salatiga berbasis *web*. Manfaat yang diharapkan dari penelitian ini yaitu untuk menyajikan informasi yang lengkap, meningkatkan mutu pendidikan, serta memberikan kemudahan bagi pengguna dalam melakukan pencarian lembaga pendidikan di kota Salatiga berdasarkan nama, alamat, fasilitas dan konsentrasi pendidikan. Sistem yang dibangun berbasis *web* dengan menggunakan *framework CodeIgniter* dan *Google Maps API*. Pengguna dalam sistem ini yaitu *Admin* yang mengelola sistem dan *User* yang dapat melihat informasi yang disajikan sistem. Sistem yang dibuat berbasis *web* yang bersifat dinamis dan memerlukan koneksi *internet* untuk menampilkan peta.

## 2. Kajian Pustaka

Dalam penelitian yang berjudul “Sistem Informasi Geografis SMA dan SMK sederajat di Purwokerto berbasis *web*”, penelitian tersebut membahas tentang perancangan Sistem Informasi Geografis yang dibuat menggunakan perangkat lunak *Arcview*, *MapViewSVG*, dan *AppServ* untuk *database server*.

Fungsi utama dalam sistem tersebut mencakup informasi lokasi sekolah, sarana prasarana, dan informasi jarak dari terminal ke sekolah.

Kekurangan yang terdapat pada sistem tersebut yaitu data hanya dibatasi pada jenjang SMA dan SMK, perhitungan jarak hanya terbatas antara satu terminal dengan salah satu sekolah dan proses digitasi peta harus diulang untuk menambahkan titik lokasi baru [5].

Penelitian kedua yaitu pada penelitian yang berjudul “Sistem Informasi Geografis (GIS) Universitas Indonesia (UI) berbasis *web* dengan menggunakan *Google Maps API*”, dalam penelitian tersebut dibangun sebuah aplikasi peta areal kampus UI berbasis *Google Maps* dengan 3 fitur utama yaitu menampilkan objek berdasarkan kategori, pencarian objek, serta menampilkan rute. Aplikasi yang dibuat menggunakan bahasa pemrograman utama *JavaScript*, *HTML* untuk tampilan antar muka *web* [6].

Penelitian ketiga yaitu penelitian yang berjudul “*Web SIG Sebaran SMA di kota Bogor Dengan Arcview dan Mapserver*”. Sistem ini dibuat sebagai media penyebaran informasi ke masyarakat melalui internet dan membantu serta memudahkan masyarakat mendapatkan informasi penyebaran SMA dan sekolah sederajat di kota Bogor. Aplikasi dibangun menggunakan *Arcview* dan *Mapserver*. Kekurangan dari penelitian ini adalah belum tersedia fasilitas edit data spasial [7].

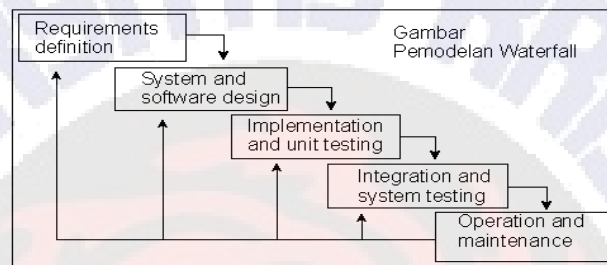
Penelitian keempat dengan judul “*Aplikasi Sistem Informasi Geografis Pendidikan Kota Depok Berbasis Web Menggunakan Quantum GIS*”. Sistem ini dibuat untuk memberikan informasi data statistik kepada pengguna dalam bentuk peta tematik dengan perbedaan warna untuk mencari posisi dari sarana pendidikan di Kota Depok. Aplikasi dibuat dengan menggunakan *QuantumGIS*, *MapServer 5*, dan *PostgreSQL*. Kekurangan yang terdapat dari sistem ini adalah kesulitan saat proses digitasi peta dan belum tersedia fasilitas untuk menambah titik sarana pendidikan [8].

Penelitian kelima yaitu berjudul “*Pembuatan Sistem Informasi Geografis (SIG) Penyebaran Sekolah Menengah Pertama (SMP) di Kota Depok Berbasis Web*”. Aplikasi yang dibuat menggunakan *Arcview*, *MapServer*, dan *framework Chameleon* dapat memberikan informasi mengenai penyebaran sekolah di tiap kecamatan di kota Depok. Kelemahan dari penelitian tersebut yaitu tidak dapat melakukan penambahan titik secara langsung pada peta karena perlu dilakukan digitasi peta untuk menambah titik lokasi terbaru [9].

Perbedaan yang terdapat pada penelitian ini yaitu penggunaan *Google Maps* untuk menjawab kelemahan dari penelitian sebelumnya. Penambahan titik lokasi pada penelitian Perancangan Sistem Informasi Pencarian Lembaga Pendidikan di kota Salatiga mudah diperbaharui karena untuk menampilkan peta *Google Maps* tidak memerlukan proses digitasi. Penelitian kali ini dapat menampilkan pencarian lembaga pendidikan berdasarkan nama, alamat, fasilitas, dan konsentrasi pendidikan. Sistem yang dibuat juga mampu untuk menampilkan rute antara lokasi *User* dan lembaga pendidikan serta dapat menghitung jarak antara rute yang dipilih. Dalam penelitian ini data untuk menampilkan titik lokasi di peta yaitu melalui sistem koordinat titik *latitude* dan *longitude* tanpa proses digitasi dan dimasukkan secara manual pada sistem saat penambahan data sekolah.

### 3. Metode Perancangan

Pengembangan sistem ini menggunakan metode *waterfall*. *Waterfall* dipilih karena jika terdapat suatu kesalahan, pengembang tidak perlu mengulangi fase dari awal, tetapi hanya cukup mengulang satu tahapan sebelumnya dan jika masalah belum terselesaikan cukup kembali ke tahap berikutnya lagi. Tahap-tahap metode *waterfall* terlihat pada Gambar 1 [10].



Gambar 1 Metode Waterfall [10]

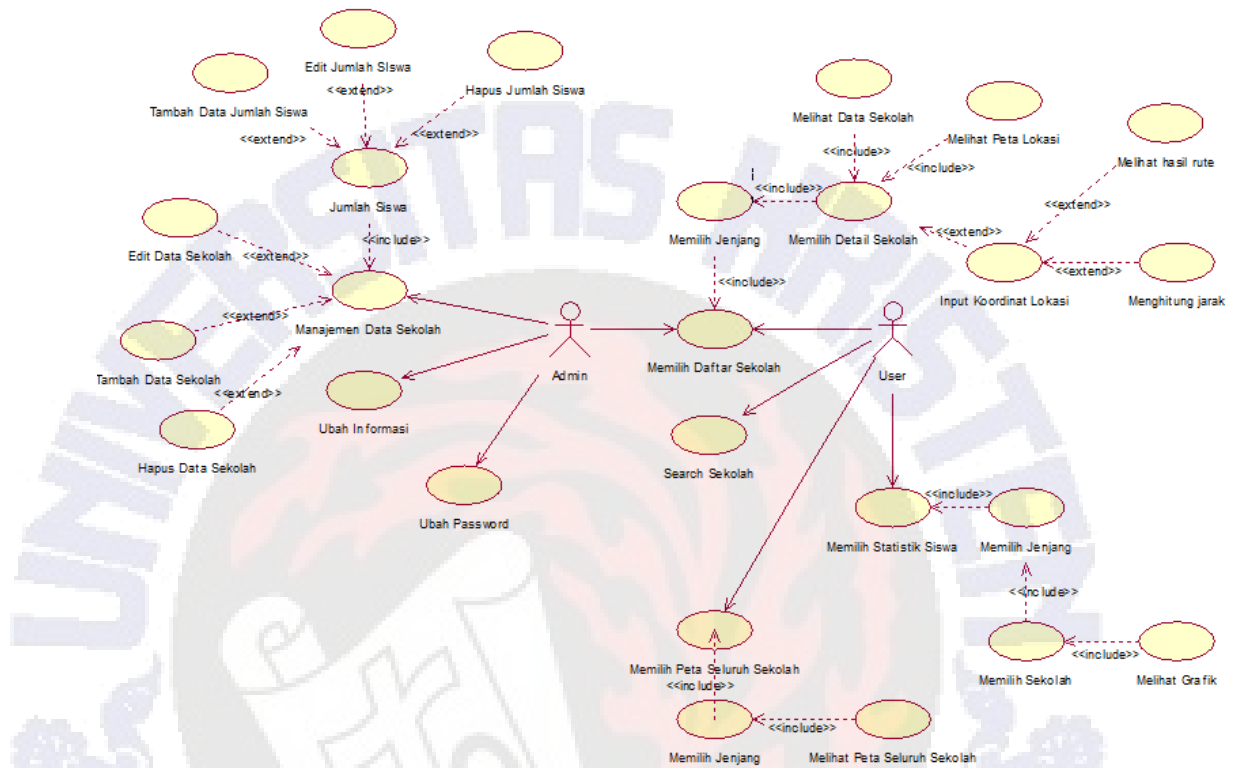
Gambar 1 merupakan tahapan umum dari model proses ini. Aktivitas yang dilakukan pada setiap tahap di model *waterfall* adalah :

- Requirements definition*. Melakukan wawancara dengan staf Bagian Perencanaan, Evaluasi dan Pelaporan pada Dinas Pendidikan, Pemuda, dan Olahraga kota Salatiga tentang kebutuhan aplikasi yang akan dibuat.
- System and software design*. Proses yang terdiri dari penentuan desain kebutuhan *hardware*, *software*, komponen, modul, antar muka pengguna dan data untuk memenuhi kebutuhan yang diperlukan pada tahap pertama.
- Implementation and unit testing*. Proses perancangan perangkat lunak yang direalisasikan sebagai serangkaian program atau perangkat lunak.
- Integration and system testing*. Unit program atau program individual diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa persyaratan sistem telah dipenuhi.
- Operation and maintenance*. Sistem diimplementasi dan dipakai. Pemeliharaan mencakup koreksi dari berbagai *error* yang tidak ditemukan pada tahap – tahap terdahulu, perbaikan atas implementasi unit sistem dan pengembangan pelayanan sistem, sementara persyaratan – persyaratan baru ditambahkan.

Untuk mendukung pelaksanaan tersebut dilakukan studi literatur dan observasi: (1) Studi literatur yaitu mengumpulkan data – data dan informasi tertulis pada buku, jurnal, laporan penelitian, maupun informasi yang didapat dari internet; (2) Observasi untuk mencari data dan informasi dengan mengunjungi langsung lembaga pendidikan di kota Salatiga.

Tahap berikutnya yaitu pemodelan data dan perancangan program pada sistem ini menggunakan *Unified Modelling Language* (UML). UML adalah bahasa grafis untuk mendokumentasikan, menspesifikasikan, dan membangun sistem perangkat lunak [11]. Langkah pertama yaitu merancang *use case diagram*

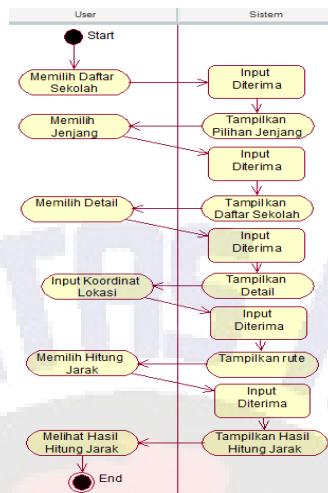
untuk menampilkan hubungan antara aktor dengan *use case* di dalam sistem [12]. *Use case* secara keseluruhan dalam sistem dapat dilihat pada Gambar 2.



**Gambar 2** *Use Case Diagram* Sistem

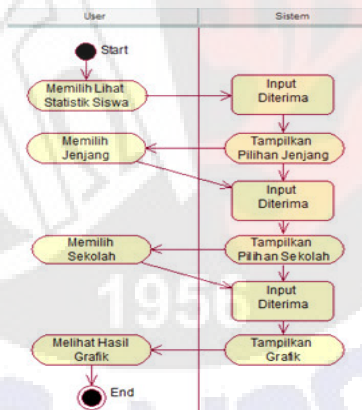
Pada Gambar 2 dapat dilihat bahwa aktor *User* hanya dapat melihat informasi yang disediakan dalam sistem informasi pencarian lembaga pendidikan di kota Salatiga berbasis *web*. Sistem akan menampilkan informasi mengenai data lembaga pendidikan dan data atributnya. Aktor *Admin* memiliki hak untuk mengelola data termasuk tambah data, *edit*, dan hapus data mengenai data lembaga pendidikan, jumlah siswa dan ubah informasi.

*Activity diagram* berfungsi untuk menggambarkan logika prosedural, proses bisnis dan alur kerja [12]. Perancangan *activity diagram* dibagi menjadi 2 bagian yaitu *User* dan *Admin*. *Activity diagram* untuk *User* dapat dilihat pada Gambar 3, Gambar 4, dan Gambar 5.



**Gambar 3** Activity diagram Daftar Sekolah

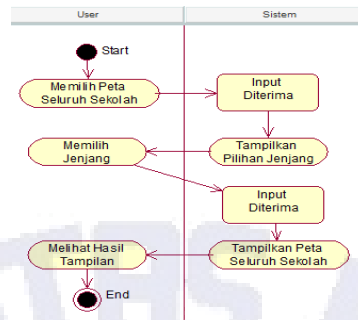
Proses dari Gambar 3 yaitu *User* memilih menu Daftar Sekolah kemudian memilih jenjang. Setelah itu sistem akan menampilkan daftar sekolah dan *User* memilih *Detail* untuk menampilkan *detail* data sekolah. Selanjutnya *User* dapat memasukkan koordinat lokasi di peta untuk mencari tahu rute dan menghitung jarak antara sekolah dengan titik lokasi yang dipilih. Sistem segera mengolah data dan menampilkan hasilnya kepada *User*.



**Gambar 4** Activity diagram Statistik Jumlah Siswa

Gambar 4 menunjukkan jika *User* memilih menu Lihat Statistik Siswa, sistem akan memberi pilihan jenjang dan *User* dapat memilih salah satu pilihan. Selanjutnya sistem akan menerima *input* dan memberi daftar pilihan sekolah – sekolah. *User* memilih satu sekolah di antara daftar sekolah yang tersedia, sistem akan menerima *input* dan memberi hasil grafik jumlah siswa pertahun suatu sekolah sesuai pilihan *User*.

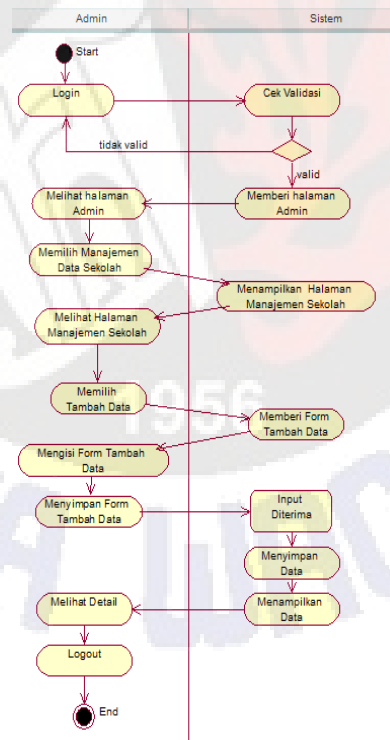




**Gambar 5** Activity diagram Peta Seluruh Sekolah

Gambar 5 menggambarkan *User* memilih menu Peta Seluruh Sekolah, sistem akan menerima *input*, dan memberi pilihan jenjang sekolah. *User* dapat memilih salah satu jenjang, setelah itu sistem memproses dan memberi tampilan peta seluruh sekolah yang ada di kota Salatiga berdasarkan jenjang yang dipilih sebelumnya.

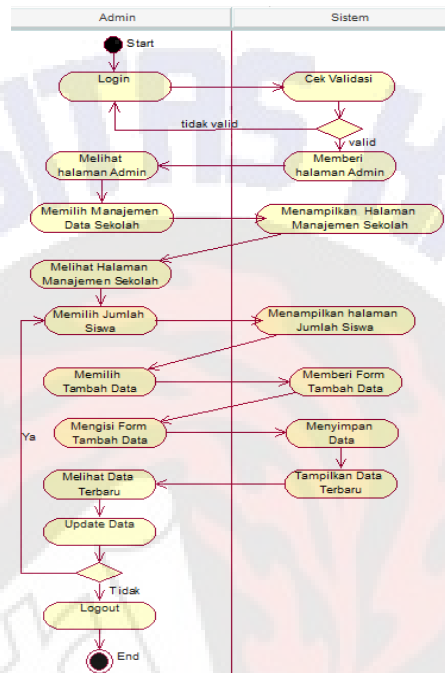
Perancangan *activity diagram* bagian Admin dapat dilihat pada Gambar 6 dan Gambar 7.



**Gambar 6** Activity diagram Tambah Data Sekolah

Pada Gambar 6 Admin *login* sebelum dapat mengolah data sistem. Setelah itu sistem akan mengecek validasi Admin, jika benar akan ditampilkan halaman Admin dan jika tidak sistem akan mengulang proses *login* dari awal. Setelah

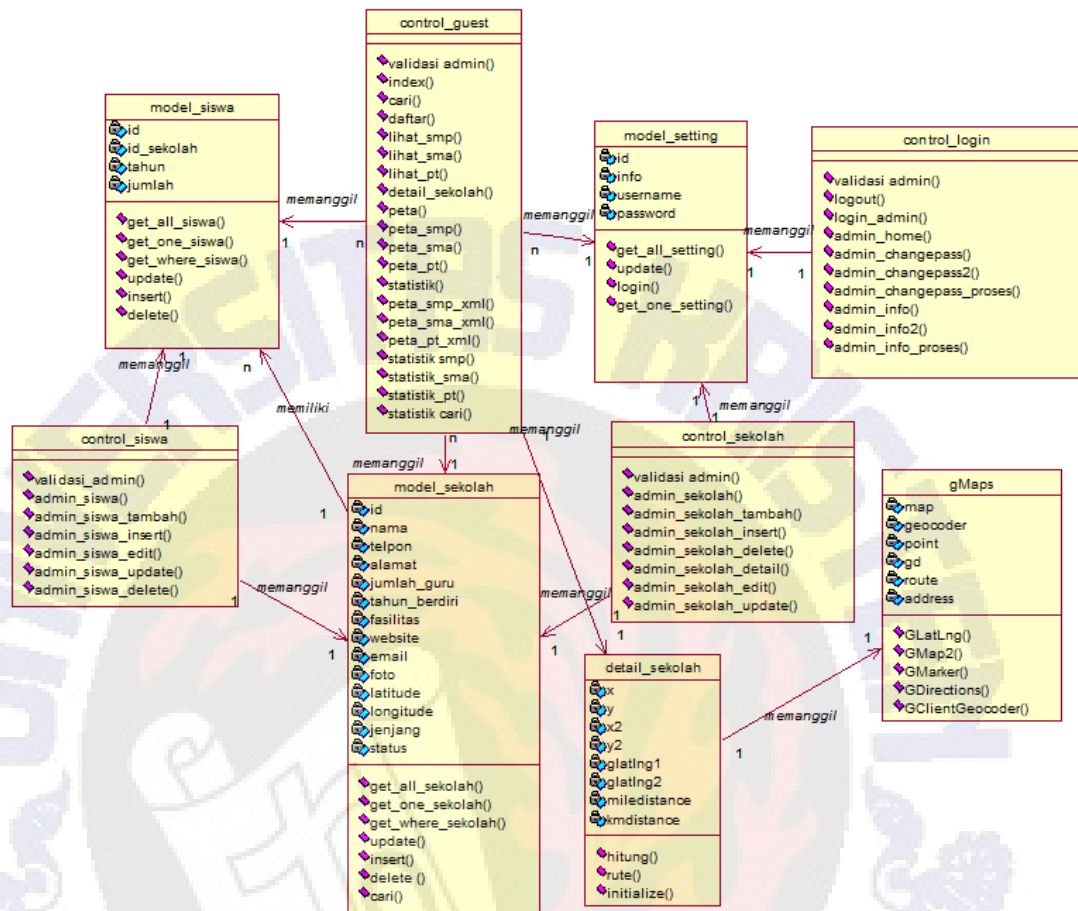
validasi, Admin dapat memilih menu Manajemen Sekolah dan memilih Tambah Data untuk melakukan penambahan data sekolah dengan mengisi *form*. Sistem akan menyimpan data untuk kemudian ditampilkan di sistem. Admin dapat *logout* sesudahnya.



**Gambar 7** Activity diagram Jumlah Siswa

Gambar 7 dijelaskan sebagai berikut: Admin melakukan *login* dan sistem memvalidasi data. Admin dapat mengakses menu Manajemen Data Sekolah dan memilih Jumlah Siswa. Halaman Jumlah Siswa ditampilkan oleh sistem dan Admin dapat melakukan penambahan, *edit*, dan hapus data. Setelah itu sistem akan menyimpan perubahan yang ada. Proses perubahan data dapat diulangi dengan memilih Jumlah Siswa, dan Admin dapat *logout* setelahnya.

Perancangan *class diagram* dapat dilihat pada Gambar 8 yang digunakan untuk menampilkan beberapa kelas serta paket – paket yang ada dalam sistem perangkat lunak yang sedang dikembangkan. Dapat dilihat pada Gambar 8, terdapat 9 *class* yang digunakan untuk merancang struktur program dari aplikasi yang dibangun. Masing-masing *class* memiliki fungsi dan atribut yang berbeda dan saling berhubungan untuk membentuk fungsi program yang komplit.



Gambar 8 Class diagram sistem

Basis data dibutuhkan untuk menyimpan data dalam sistem. Rancangan tabel yang dibuat adalah Tabel Jumlah Siswa, Tabel Setting, dan Tabel Sekolah dapat dilihat pada Tabel 1, Tabel 2, dan Tabel 3.

Tabel 1 Tabel Jumlah Siswa

Field	Tipe	Keterangan
id	integer(11)	primary
id_sekolah	integer(11)	
tahun	varchar(255)	
jumlah	integer(11)	

Tabel 2 Tabel Setting

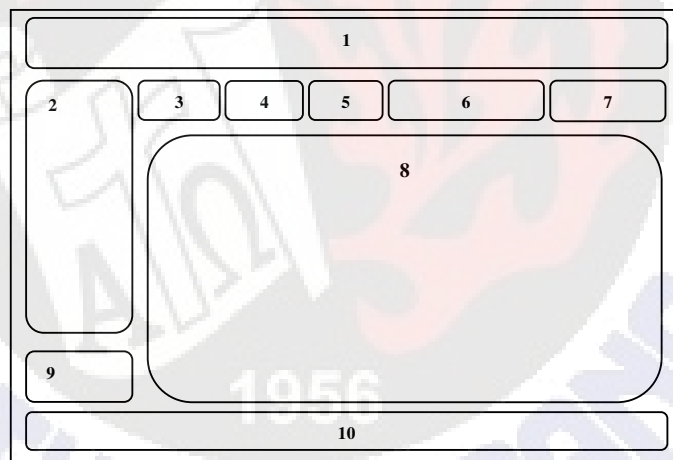
Field	Tipe	Keterangan
id	integer(11)	primary
username	varchar(255)	
password	varchar(255)	
informasi	text	



**Tabel 3** Tabel Data Sekolah

Field	Tipe	Keterangan
id	integer(11)	<i>primary</i>
Nama	varchar(255)	
Telp	varchar(255)	
alamat	varchar(255)	
jumlah_guru	text	
tahun_berdiri	varchar(255)	
fasilitas	text	
website	varchar(255)	
e-mail	varchar(255)	
foto	varchar(255)	
latitude	double	
longitude	double	
jenjang	varchar(255)	
status	varchar(255)	

Perancangan desain antar muka diperlukan sebagai dasar untuk membuat tampilan aplikasi yang akan dibuat. Rancangan dapat dilihat pada Gambar 9.

**Gambar 9** Antar Muka Aplikasi

Gambar 9 menunjukkan ada 10 bagian dari antar muka aplikasi yang berturut-turut merupakan *header web*, menu, *link* jenjang SMP, *link* jenjang SMA, *link* jenjang perguruan tinggi, *link* statistik jumlah siswa, kolom *search*, menu konten, *form login*, dan *footer*.

#### 4. Hasil Perancangan Dan Pengujian

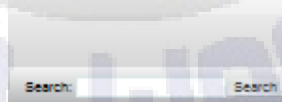
Tahap pertama yaitu integrasi *Google Maps* API dengan mendapatkan *Google Maps* API *key*. *Key* ini didapatkan dari situs <http://code.Google.com/APIs/Maps/signup.html>. Setelah terdaftar, *key* yang digunakan dalam penelitian ini adalah

“ABQIAAAAPDUET0Qt7p2VcSk6JNU1sBSM5jMcmVqUpI7aqV44cW1cEECiThQYkcZUPRjN9vy\_TWxWvuLoOfSFBw”. Tahap selanjutnya adalah hasil dari perancangan sistem yang telah dibuat untuk mengetahui apakah sistem layak untuk digunakan atau tidak.



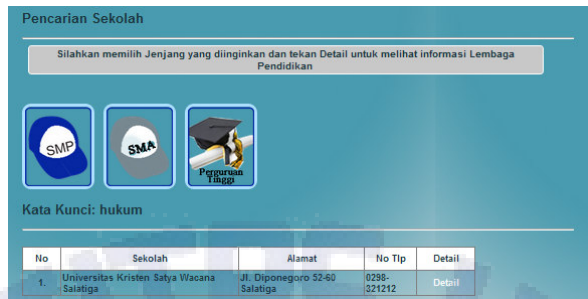
**Gambar 10** Tampilan Awal

Pada Gambar 10 tampak tampilan antar muka awal saat sistem diakses. Tampilan antar muka ini terdiri dari *header* dengan logo situs, *link bar* untuk mengakses menu secara langsung, kolom *search* untuk pencarian sekolah, menu yang terdiri dari menu-menu utama di dalam sistem, *form login* untuk *Admin*, dan *footer*.



**Gambar 11** Search

Gambar 11 merupakan tampilan menu *Search* yang digunakan *User* untuk mencari sekolah berdasarkan nama, alamat, fasilitas, atau konsentrasi studi. Setelah *User* memasukkan kata kunci nama sekolah, sistem akan mencari dan menampilkan hasil pencarian sekolah yang dimaksud seperti terlihat pada Gambar 12.



**Gambar 12** Hasil Pencarian

Pada Gambar 12 sistem akan menampilkan hasil berupa nama sekolah yang dicari, alamat sekolah, nomer telpon dan *link Detail* yang dapat dipilih untuk melihat informasi lebih lengkap mengenai sekolah yang dicari.



**Gambar 13** Informasi Sekolah

Gambar 13 merupakan tampilan informasi terperinci mengenai sekolah yang dipilih yaitu informasi nama, alamat, *website*, *email*, foto, tahun berdiri, jumlah pengajar, letak geografis, status, fasilitas dan konsentrasi pendidikan (bila ada). Halaman ini juga dilengkapi dengan peta yang menunjukkan titik lokasi sekolah. *User* juga dapat melakukan hitung jarak dan melihat rute antara lokasi sekolah dengan lokasi pilihan *User*. Gambar 14 merupakan tampilan peta yang didalamnya terlihat titik lokasi sekolah.



**Gambar 14** Peta Sekolah

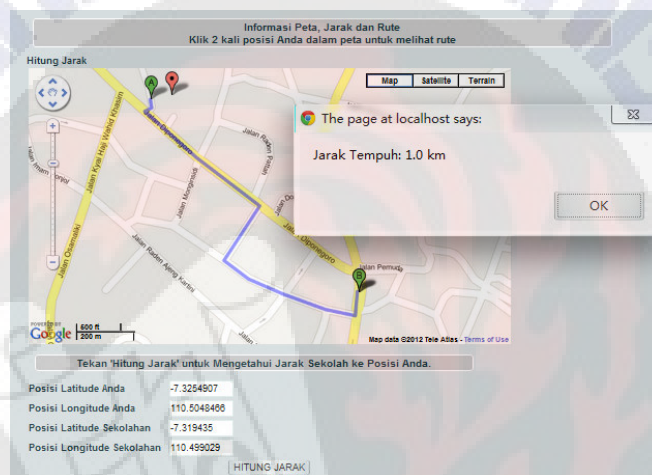
Kode program untuk menampilkan peta *Google Maps* pada halaman *detail* dapat dilihat pada Kode Program 1.

#### Kode Program 1 Integrasi *Google Maps*

```
<script
src="http://Maps.Google.com/Maps?file=api&v=2&sensor=false&key=ABQIAAA
APDUET0Qt7p2VcSk6JNU1sBSM5jMcmVqUpI7aqV44cW1cEECiThQYkcZUPRjN9vy_TWxWvuLoOfSFBw"
type="text/javascript"></script> <script type="text/javascript">
var map;
var geocoder;
var address;
function initialize() {
map = new GMap2(document.getElementById("map_canvas"));
map.setCenter(new GLatLng(-7.343436,110.503197), 15);
map.setUIToDefault(); GEvent.addListener(map, "click", getAddress);
geocoder = new GClientGeocoder();}
function getAddress(overlay, latlng) { if (latlng != null) {
address = latlng;
geocoder.getLocations(latlng, showAddress);}}
function showAddress(response) {
map.clearOverlays();
if (!response || response.Status.code != 200) {
alert("Status Code:" + response.Status.code);
} else {
place = response.Placemark[0];
point = new GLatLng(place.Point.coordinates[1],
place.Point.coordinates[0]);
marker = new GMarker(point);
map.addOverlay(marker);
document.getElementById("x").value = place.Point.coordinates[1];
document.getElementById("y").value = place.Point.coordinates[0];}}
function shownewAddress(address) {
if (geocoder) {geocoder.getLatLng(address,
function(point) {
if (!point) {
alert(address + " not found");
} else {
map.setCenter(point, 17);
var marker = new GMarker(point);
map.addOverlay(marker);
marker.openInfoWindowHtml(address);
}} ); } }
```

Pertama disisipkan *key* untuk integrasi *Google Maps API* pada *web* yaitu “ABQIAAAAPDUET0Qt7p2VcSk6JNU1sBSM5jMcmVqUpI7aqV44cW1cEECiThQYkcZUPRJn\_TWxWvuLoOfSFBw”. Fungsi *initialize()* digunakan untuk mengatur peta secara *default*, titik tengah peta pada kota Salatiga yang memiliki *latitude* – 7.343436 dan *longitude* 110.503197 sedangkan angka 15 merupakan skala perbesaran peta.

Selanjutnya pada peta pada halaman *Detail*, dapat dilakukan fungsi melihat rute dan hitung jarak. *User* dapat memilih salah satu titik lokasi pada peta untuk melihat hasil kedua fungsi tersebut. Gambar 15 merupakan hasil tampilan fungsi melihat rute dan hitung jarak.



**Gambar 15** Hitung Jarak dan Lihat Rute

Gambar 15 merupakan tampilan peta untuk fungsi lihat rute dan hitung jarak. Sebelumnya *User* memilih lokasi yang diinginkan dengan melakukan 2 kali klik pada peta untuk melihat rute. Rute akan ditampilkan dengan sebuah garis berwarna biru. Selanjutnya *User* dapat menekan tombol Hitung Jarak untuk mendapatkan informasi jarak tempuh dengan satuan kilometer. Kode program untuk hitung jarak dapat dilihat pada Kode Program 2, sedangkan kode program untuk melihat rute dapat dilihat pada Kode Program 3.

#### **Kode Program 2** Hitung Jarak

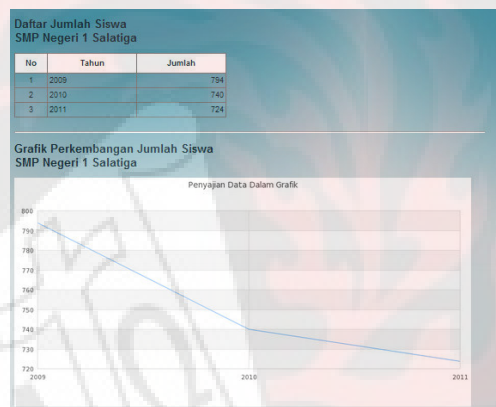
```
function hitung() {
    var x = document.getElementById("x").value;
    var y = document.getElementById("y").value;
    var x2 = document.getElementById("x2").value;
    var y2 = document.getElementById("y2").value;
    var glatlng1 = new GLatLng(x, y);
    var glatlng2 = new GLatLng(x2, y2);
    var miledistance = glatlng1.distanceFrom(glatlng2, 3959).toFixed(1);
    var kmdistance = (miledistance * 1.609344).toFixed(1);
    alert('Jarak Tempuh: ' + kmdistance + ' km!');
    return false; }
```

Fungsi pada Kode Program 2 merupakan fungsi untuk menghitung jarak antara lembaga pendidikan dan lokasi yang dipilih oleh *User*. Baris kode `var kmdistance = (miledistance * 1.609344)` digunakan untuk kalkulasi jarak antara dua titik di peta. Dua titik yang dimaksud yaitu `glatlng1` dan `glatlng2`.

#### Kode Program 3 Lihat Rute

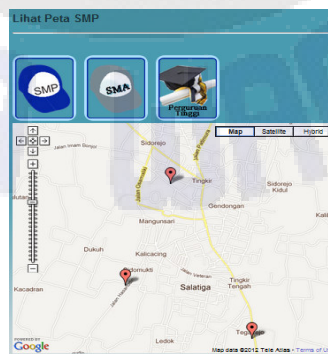
```
var gd = new GDirections(map,null);
gd.load("from: <?php echo $daftar->latitude; ?>,<?php echo $daftar->longitude; ?> to: " + place.Point.coordinates[1] + "," + place.Point.coordinates[0]);
var route; GEvent.addListener(gd,"load", function() {
var numRoute = gd.getNumRoutes(); });
```

Pada Kode Program 3 dapat dilihat kode program untuk melihat rute, fungsi `GDirections()` dipanggil setelah mendapatkan koordinat dari *input* yang dimasukkan *User* sebelumnya. Fungsi `GDirections()` merupakan fungsi *Google Maps* API yang menangani rute pada peta dari satu lokasi ke tempat lainnya.



Gambar 16 Statistik Jumlah Siswa

Gambar 16 merupakan tampilan Statistik Jumlah Siswa berupa grafik berbentuk garis. Grafik ini menampilkan perkembangan jumlah siswa per tahun di satu sekolah.



Gambar 17 Peta Seluruh Sekolah

Gambar 17 merupakan halaman Peta Seluruh Sekolah yang menampilkan seluruh sekolah di kota Salatiga sesuai dengan jenjang yang dipilih.



Daftar Sekolah Di Kota Salatiga									
[ Tambah Data ]									
No	Jenjang	Nama	Alamat	Status	Jml Siswa	Detail	Edit	Hapus	
1	SMA	SMA Negeri 2 Salatiga	Jl. Tegayono No. 79	Negri	Jml Siswa	Detail	Edit	Hapus	
2	SMA	SMA Kristen 1 Salatiga	OSI MALUKU NO. 32 SALATIGA	Swasta	Jml Siswa	Detail	Edit	Hapus	
3	SMA	MADRASAH ALYAH NEGERI SALATIGA	JL. KH. WAHID HASYIM NO. 12	Swasta	Jml Siswa	Detail	Edit	Hapus	

**Gambar 18** Manajemen Daftar Sekolah

Untuk menampilkan data sekolah dalam aplikasi ini, maka dibutuhkan halaman *administrator*. Gambar 18 merupakan menu Manajemen Daftar Sekolah yang digunakan *Admin* untuk mengelola data sekolah dengan fitur berupa Tambah Data untuk menambah daftar sekolah, Jml Siswa untuk mengelola statistik jumlah siswa per tahun di suatu sekolah, dan fungsi *Edit* dan Hapus untuk merubah data sekolah dan *Detail* yang berguna untuk menampilkan hasil *input* terakhir.

**Gambar 19** Tambah Daftar Sekolah

Pada Gambar 19 dapat dilihat sebuah *form* yang dapat dimasukkan data terkait dengan sekolah yang ingin didaftarkan dalam aplikasi. Pada *form* ini ada dua kolom berupa *latitude* dan *longitude* yang harus diisi untuk menampilkan titik lokasi dari lembaga pendidikan.

### **Black Box Testing**

Tahap berikutnya yaitu dilakukan pengujian validasi terhadap sistem yang telah dirancang. *Black box* merupakan pendekatan yang digunakan sebagai pelengkap untuk menemukan kesalahan [13]. Validasi sistem dapat dilihat pada Tabel 4.

**Tabel 4** Hasil Pengujian *Black Box*

No	Point Pengujian	Validasi <i>Input</i>	Data <i>Input</i>	Hasil Uji
1.	Proses <i>login</i>	<i>username</i>	dikosongkan	gagal <i>login</i>
		<i>password</i>	<i>value name</i>	berhasil <i>login</i>
			dikosongkan	gagal <i>login</i>
			<i>value password</i>	berhasil <i>login</i>
2.	Cari sekolah	kata kunci	nama sekolah	berhasil tampil
3.	Tambah Daftar Sekolah	<i>input</i> Tambah Data	pilih menu	berhasil dibuat
		lihat <i>detail</i> data	pilih menu	berhasil dibuat
4.	Tambah Jumlah Siswa	<i>input</i> tambah data	pilih menu	berhasil dibuat
5.	Ubah Informasi	<i>input</i> baru	pilih menu	berhasil dibuat
		lihat data	pilih menu	berhasil dibuat
6.	Ubah <i>Password</i>	<i>input username</i>	Pilih menu	berhasil dibuat
		<i>input password</i>	Pilih menu	berhasil dibuat
7.	<i>Logout</i>	<i>logout</i>	Pilih menu	berhasil keluar

Pada pengujian yang dilakukan pada Tabel 4, dapat dilihat bahwa proses validasi sistem sudah sesuai dengan apa yang dirancang dan berjalan sesuai yang diharapkan. Proses pencarian sekolah dapat dilakukan dengan memasukkan kata kunci berupa nama sekolah. Sedangkan proses tambah daftar sekolah dan *detail* data berhasil dibuat setelah dilakukan *input* data ke dalam sistem. Tambah jumlah siswa, ubah informasi, dan ubah *password* bagi *Admin* juga berhasil dibuat.

### Pengujian Pengguna

Pengujian ini bertujuan untuk mengukur kualitas aplikasi yang dibuat dengan 30 orang sebagai sampel. Penyusunan dan perhitungan pengujian ini menggunakan skala *Likert* yang merupakan cara sistematis untuk memberi skor pada indeks [14]. Di dalam skala *Likert* tersebut diberi bobot nilai 1 sampai 5, yaitu Sangat Kurang (SK), Kurang (K), Cukup (C), Baik (B), dan Sangat Baik (SB). Dari hasil kuesioner dapat dihitung hasil penilaian keseluruhan sistem melalui pengujian dengan skala *Likert* dapat dilihat pada Tabel 5.

**Tabel 5** Hasil Penilaian Keseluruhan

Pertanyaan	SB	B	C	K	SK	Total	Rata-Rata
1	2	21	7	-	-	115	3.8
2	6	20	4	-	-	122	4.06
3	3	15	12	-	-	111	3.7
4	2	21	7	-	-	115	3.8
5	1	21	8	-	-	113	3.7
6	8	18	4	-	-	124	4.1
7	7	19	4	-	-	123	4.1
Total dan Rata-rata Keseluruhan						823	3.89

Pada Tabel 5 dapat dilihat bahwa hasil penilaian keseluruhan sistem memperoleh nilai rata-rata 3.89 yang berarti menunjukkan nilai (B) atau Baik karena terletak antara rentang nilai Baik.

## 5. Simpulan

Berdasarkan hasil perancangan sistem informasi pencarian lembaga pendidikan di kota Salatiga berbasis *web*, maka dapat disimpulkan bahwa teknologi *web* yang diterapkan dalam sistem ini dapat digunakan dengan baik oleh *user* dan dapat membantu dalam pencarian lembaga pendidikan. Hal ini ditunjukkan dengan adanya perolehan rata-rata nilai (B) yang berarti baik. Dengan aplikasi ini *user* dapat mencari lokasi berdasarkan nama, alamat, fasilitas, dan konsentrasi lembaga pendidikan di kota Salatiga yang sedang dicari khusus untuk jenjang SMP, SMA dan perguruan tinggi baik Negeri maupun Swasta.

Sistem pencarian lembaga pendidikan di Kota Salatiga ini masih memiliki keterbatasan yang muncul, untuk itu saran yang dapat disampaikan adalah : penambahan jenjang SD, dibuat pembagian kategori lembaga pendidikan berdasarkan Kecamatan, serta integrasi dengan situs Pemkot Salatiga sehingga diharapkan dapat memberi manfaat yang besar di sektor pendidikan kota Salatiga.

## 6. Daftar Pustaka

- [1] Coombs, Philip dan Ahmed, Manzoor, 1978, *Attacking Rural Poverty: How Non Formal Education Can Help*, Baltimore: Johns Hopkins University Press.
- [2] Kesowo, Bambang, 2003, *Undang-Undang Republik Indonesia Nomor 20 Tahun 2003 Tentang Sistem Pendidikan Nasional*,  
[www.dikti.go.id/Archive2007/UUno20th2003-Sisdiknas.htm](http://www.dikti.go.id/Archive2007/UUno20th2003-Sisdiknas.htm). Diakses tanggal 15 Februari 2012.
- [3] Staff Bagian Perencanaan, Evaluasi dan Pelaporan Dinas Pendidikan Kota Salatiga.
- [4] Wicaksono, Hendro, 2005. *Arsitektur Informasi : Membantu Pencarian Informasi* di *Website*.  
[http://hendrowicaksono.multiply.com/journal/item/7?&show\\_interstitial=1&u=%2Fjournal%2Fitem](http://hendrowicaksono.multiply.com/journal/item/7?&show_interstitial=1&u=%2Fjournal%2Fitem). Diakses tanggal 20 Juni 2012
- [5] Prihatin, Tri, 2009, *Sistem Informasi Geografis SMA dan SMK Sederajat di Purwokerto Berbasis Web*,  
[http://library.upnvj.ac.id/index.php?p=show\\_detail&id=5640](http://library.upnvj.ac.id/index.php?p=show_detail&id=5640). Diakses tanggal 1 Maret 2012.
- [6] Ichtara, Cita, 2008, *Implementasi Aplikasi Sistem Informasi Geografis (SIG) Universitas Indonesia (UI) Berbasis Web dengan Menggunakan Google Maps* API,

- <http://www.digilib.ui.ac.id/opac/themes/libri2/detail.jsp?id=126728&lokalo=lokal>. Diakses tanggal 1 Maret 2012.
- [7] Kartika, Sari, 2012. *Web SIG Sebaran SMA Di Kota Bogor Dengan Arcview dan Mapserver*, <http://repository.gunadarma.ac.id/bitstream/123456789/2519/1/119.pdf>. Diakses 18 Juni 2012.
- [8] Novianti, 2012. *Aplikasi Sistem Informasi Geografis Pendidikan Kota Depok Berbasis Web Menggunakan Quantum GIS*, [http://repository.gunadarma.ac.id/bitstream/123456789/1537/1/Artikel\\_11105172.pdf](http://repository.gunadarma.ac.id/bitstream/123456789/1537/1/Artikel_11105172.pdf). Diakses 18 Juni 2012.
- [9] Hapsari, Galuh dan Ernastuti, 2012. *Pembuatan Sistem Informasi Geografis (SIG) Penyebaran Sekolah Menengah Pertama (SMP) di Kota Depok Berbasis Web*, <http://repository.gunadarma.ac.id/handle/123456789/946>. Diakses 18 Juni 2012.
- [10] Pressman, Roger.S, 2001, *Software Engineering : A Practitioner's Approach*, Amerika Serikat : McGraw-Hill.
- [11] Sulistyorini, Prastuti, 2009, *Visual Modelling dengan Menggunakan UML dan Rational Rose*, *Jurnal Teknologi Informasi DINAMIK* : Vol XIV No.1, [www.unisbank.ac.id/ojs/index.php/fti1/article/download/87/82](http://www.unisbank.ac.id/ojs/index.php/fti1/article/download/87/82). Diakses tanggal 10 Maret 2012.
- [12] Fowler, Martin, 2005, *UML Distilled*, Yogyakarta: Andi.
- [13] Ayuliana, 2009, *Testing dan Implementasi*, <http://staffsite.gunadarma.ac.id/rifiana/index.php?stateid=download&ii=26083&part=files>. Diakses tanggal 25 April 2012.
- [14] Waryanto, Budi, dan Millafati, Yuan, 2006, *Transformasi Data Skala Ordinal ke Interval dengan Menggunakan Makro Minitab*, <http://staffsite.gunadarma.ac.id/myunanto/index.php?stateid=download&ii=19376&part=files>. Diakses tanggal 21 April 2012.