

DAFTAR PUSTAKA

- Ahmat Adil, S. K. M. S. n.d. *Sistem Informasi Geografis*. Penerbit Andi.
- Angga Kurniawan, Suendri, and Triase. 2019. "Sistem Informasi Geografis Pemetaan Lokasi Panti Asuhan Di Kota Medan." *JISTech (Journal of Islamic Science and Technology)* 3(2):118–26.
- Anggraeni, E. Y., E. Risanto, Y. Basuki, D. Nofianto, A. A. C, and A. Offset. n.d. *Pengantar Sistem Informasi*. Penerbit Andi.
- Awangga, R. M., and Y. H. Setyawan. 2019. *Pengantar Sistem Informasi Geografis: Sejarah, Definisi Dan Konsep Dasar*. Kreatif.
- Batubar, Febrin Aulia. 2012. "Perancangan Website Pada Pt . Ratu Enim Palembang." 15–27.
- Dedy Rahman Prehanto, S. K. M. K., S. T. M. K. I Kadek Dwi Nuryana, and S. M. Pustaka. 2020. *Buku Ajar Konsep Sistem Informasi*. Scopindo Media Pustaka.
- Enterprise, J. 2016. *Pengenalan HTML Dan CSS*. Elex Media Komputindo.
- Enterprise, J. 2018. *HTML, PHP, Dan MySQL Untuk Pemula*. Elex Media Komputindo.
- Hanafi. 2017. "Konsep Penelitian R & D Dalam Bidang Pendidikan." *Saintifika Islamica: Jurnal Kajian Keislaman* 4(2):130.
- Hutahaean, J. 2015. *Konsep Sistem Informasi*. Deepublish.
- Ikhwan, Ali. 2018. "Penerapan Data Mining Untuk Analisis Pola Penerimaan Security Baru Pada Perusahaan Biro Jasa PT. DM Security Medan Menggunakan Metode Algoritma Apriori." 83.
- Irawan, Yudie, Nanik Susanti, and Wiwit Agus Triyanto. 2016. "Analisa Dan Perancangan Sistem Informasi Sekolah Berbasis Website Untuk Penyampaian Informasi Sekolah Dan Media Promosi Kepada Masyarakat." *Simetris : Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer* 7(1):257.
- Irwan Padli Nasution, Muhammad, and Samsudin Samsudin. 2018. "Using Google Location APIs to Find an Accurate Criminal Accident Location." *International Journal of Engineering & Technology* 7(3):1818. Jayanti, Dwi,

- and Iriani Siska. 2014. "Sistem Informasi Penggajian Pada CV . Blumbang Sejati Pacitan." *Journal Speed - Sentra Penelitian Engineering Dan Edukasi* 6(3):36–43.
- Judiani, Sri. 2011. "Implementasi Pendidikan Karakter Di Sekolah Dasar Dan Menengah, Kemendiknas." *Jurnal Pendidikan Dan Kebudayaan* 16(Pendidikan):280–88.
- Julianti, M. Ramaddan, Agus Budiman, and Agil Patriosa. 2018. "Perancangan Sistem Informasi Geografis Pemetaan Lokasi Apotek Di Wilayah Kota Bogor Berbasis Web." *Jurnal Sisfotek Global* 8(1):13–19.
- Lubis, K. H. 2018. *Roman Medan: Sebuah Kota Membangun Harapan*. PT Gramedia Pustaka Utama.
- Masykur, Fauzan. 2014. "Implementasi Sistem Informasi Geografis Menggunakan Google Maps Api Dalam Pemetaan Asal Mahasiswa." *Jurnal SIMETRIS* 5(2):181–86.
- Maulana, Ikhsan, Muhammad Irawan Padli Nasution, and Ali Ikhwan. 2020. "Aplikasi Pendaftaran Siswa Baru Menggunakan Algoritma Best First Search Pada SMP Negeri 1 Medab." *Journal of Chemical Information and Modeling* 53(9):1689–99.
- Miftahul Jannah, S. C. C. 2019. *Mahir Bahasa Pemrograman PHP*. Elex Media Komputindo.
- Nasution, Nur Wahyudin. 2015. "Kepemimpinan Pendidikan Di Sekolah." *Atma Rekha : Jurnal Pastoral Dan Kateketik* 5(2):25.
- Novianto, Dian. 2016. "Implementasi Sistem Informasi Pegawai (Simpeg) Berbasis Web Menggunakan framework Codeigniter Dan Bootstrap." *Ilmiah Informatika Global* 7(1):10–16.
- Nur, M. Adnan, and Nurilmiyanti Wardhani. 2018. "Penerapan Formula Faversine Dalam Perhitungan Luas Wilayah Menggunakan Koordinat Google Maps." *Jurnal IT* 9(1):58–64.
- Purnomo, H. 2019. *Pemodelan Dan Simulasi Untuk Pengelolaan Adaptif Sumber Daya Alam Dan Lingkungan*. PT Penerbit IPB Press.
- Rianandra, Arsali, and Akhmad Aminuddin Bama. 2015. "Studi Perbandingan

- Penentuan Posisi Geografis Berdasarkan Pengukuran Dengan GPS (Global Positioning System), Peta Google Earth, Dan Navigasi.Net.” *Jurnal Penelitian Sains Mipa UNSRI* 17(2):82–90.
- Rifai, Ahmad. 2013. “Sistem Informasi Pemantauan Posisi Kendaraan Dinas Unsri Menggunakan Teknologi GPS.” *Jurnal Sistem Informasi* 5(2):603–10.
- Rosa, and Shalahuddin. 2013. “Pemodelan Visual Dengan Menggunakan UML Dan Rational Rose.” *Jurnal Teknologi Informasi DINAMIK* 14(1):23–29.
- Samsudin, Samsudin. 2019. “Optimalisasi Penerimaan Remunerasi Dosen Menggunakan Metode Rule Base Reasoning.” *Klik - Kumpulan Jurnal Ilmu Komputer* 6(3):224.
- Sarintan Kaharu Sarinta, and Oki sakina. 2016. “Perancangan Sistem Informasi Pengolahan Data Akademik.” *Perancangan Sistem Informasi Pengolahan Data Akademik Pada TK Al-Hidayah Lolu* 2(1).
- Yanto, R. 2016. *Manajemen Basis Data Menggunakan MySQL*. Deepublish.
- Yulianto, Yulianto, Ramadiani Ramadiani, and Awang Harsa Kridalaksana. 2018. “Penerapan Formula Haversine Pada Sistem Informasi Geografis Pencarian Jarak Terdekat Lokasi Lapangan Futsal.” *Informatika Mulawarman : Jurnal Ilmiah Ilmu Komputer* 13(1):14.
- Zufria, Ilka. 2013. “Pemodelan Berbasis UML (Unified Modeling Language) Dengan Strategi Teknik Orientasi Objek User Centered Design (UCD) Dalam Sistem Administrasi Pendidikan Pemodelan Berbasis UML (Unified Modeling Language) Dengan.” *Journal Sains & Teknologi* 1(1):1–16.

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

LAMPIRAN

Lampiran 1. Surat Izin Riset

1. Surat Pengajuan Riset di Portal SI-SELMA ke-Balitbang

		KEMENTERIAN AGAMA REPUBLIK INDONESIA UNIVERSITAS ISLAM NEGERI SUMATERA UTARA MEDAN FAKULTAS SAINS DAN TEKNOLOGI Jl. Willem Iskandar Pasar V Medan Estate 20371 Telp. (061) 6615683-6622925 Fax. 6615683	
Nomor	: B.438/ST.I/ST.V.2/TL.00/5/2021	28 Mei 2021	
Lampiran	: -		
Hal	: Izin Riset		
Yth. Bapak/Ibu Kepala Badan Penelitian dan Pengembangan Kota Medan			
<i>Assalamualaikum Wt. Wb.</i>			
Dengan Hormat, diberitahukan bahwa untuk mencapai gelar Sarjana Strata Satu (S1) bagi Mahasiswa Fakultas Sains dan Teknologi adalah menyusun Skripsi (Karya Ilmiah), kami tugaskan mahasiswa:			
Nama	: Anggi Dessisiliya		
NIM	: 0702171042		
Tempat/Tanggal Lahir	: Pematang Siantar, 04 Desember 1999		
Program Studi	: Sistem Informasi		
Semester	: IX (Sembilan)		
Alamat	: Jl. Kemuning Emplasmen PTPN IV Dolok Sinumbah Kelurahan Dolok Sinumbah Kecamatan Huta Bayu Raja		
untuk hal dimaksud kami mohon memberikan Izin dan bantuannya terhadap pelaksanaan Riset di Jl. Kapten Maulana Lubis No.2, Petisah Tengah, Kecamatan Medan Petisah, guna memperoleh informasi/keterangan dan data-data yang berhubungan dengan Skripsi (Karya Ilmiah) yang berjudul:			
<i>Sistem Informasi Geografis Sekolah Di Kota Medan Menggunakan Algoritma Haversine</i>			
Demikian kami sampaikan, atas bantuan dan kerjasamanya diucapkan terima kasih.			
		Medan, 28 Mei 2021 a.n. DEKAN Wakil Dekan Bidang Akademik dan Kelembagaan	
		 <i>Digitally Signed</i>	
		Dr. Abdul Halim Dauday, ST., M.Si NIP. 198111062005011003	
UNIVERSITAS ISLAM SUMATERA UTARA MEDAN			
Tembusan: - Dekan Fakultas Sains dan Teknologi UIN Sumatera Utara Medan			

2. Surat Balasan dari Balitbang ke-Dinas Pendidikan Kota Medan

**PEMERINTAH KOTA MEDAN**
BADAN PENELITIAN DAN PENGEMBANGAN
Jalan Kapten Maulana Lubis Nomor 2 Medan Kode Pos 20112
Telepon (061) 4555693 Faks (061) 4555693
E-mail: balitbangmedan@yahoo.co.id Website: balitbang.pemkotamedan.go.id

SURAT REKOMENDASI RISET
NOMOR : 070/ENV/balitbang/2021

Berdasarkan Surat Keputusan Walikota Medan Nomor: 57 Tahun 2001, Tanggal 13 November 2001 dan Peraturan Walikota Medan Nomor: 55 Tahun 2010, tanggal 24 November 2010 tentang Tugas Pokok dan Fungsi Badan Penelitian dan Pengembangan Kota Medan dan setelah membaca/memperhatikan surat dari Wakil Dekan I Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan. Nomor: B-439/ST.I/ST.V.2/TL.00/5/2021 Tanggal: 28 Mei 2021. Hal: Izin Riset.

Badan Penelitian dan Pengembangan Kota Medan dengan ini memberikan Surat Rekomendasi Riset Kepada :

Nama : Anggi Desisailiya.
NIM : 0702171042.
Prodi : Sistem Informasi.
Lokasi : Dinas Pendidikan Kota Medan.
Judul : " Sistem Informasi Geografis Sekolah Kota Medan Menggunakan Algoritma Haversine "
Lamanya : 1 (Satu) Bulan.
Penanggung Jawab : Wakil Dekan I Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.

Dengan ketentuan sebagai berikut :

1. Sebelum melakukan Penelitian terlebih dahulu harus melapor kepada pimpinan Organisasi Perangkat Daerah lokasi yang ditetapkan.
2. Mematuhi peraturan dan ketentuan yang berlaku di lokasi Penelitian.
3. Tidak dibenarkan melakukan Penelitian atau aktivitas lain di luar lokasi yang telah direkomendasikan.
4. Hasil penelitian diserahkan kepada Kepala Balitbang Kota Medan selambat lambatnya 2 (dua) bulan setelah penelitian dalam bentuk soft copy atau melalui Email / balitbangmedan@yahoo.co.id /.
5. Surat rekomendasi penelitian dinyatakan batal apabila pemegang surat rekomendasi tidak mengindahkan ketentuan atau peraturan yang berlaku pada Pemerintah Kota Medan.
6. Surat rekomendasi penelitian ini berlaku sejak tanggal dikeluarkan.

Demikian Surat ini diperbuat untuk dapat dipergunakan sebagaimana mestinya.

Dikeluarkan di : Medan.
Pada Tanggal : 27 Mei 2021
Plt. Kepala Badan Penelitian dan Pengembangan


Drs. SYARIFUDIN S. DONGGOKAN, M.Si
PENGHUBUNG RUMAH (IV/c)
NIP. 19600401990091001

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

Terdistribusi :
1. Walikota Medan sebagai Laporan,
2. Kepala Dinas Pendidikan Kota Medan.
3. Wakil Dekan I Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan.
4. Bertinggal.

3. Surat Balasan dari Dinas Pendidikan Kota Medan



PEMERINTAH KOTA MEDAN
DINAS PENDIDIKAN
Jalan Pelita IV No.77 Telepon dan Faksimili 061-80089992
E-mail : disdikmedan17@gmail.com
MEDAN -20236

SURAT KETERANGAN
Nomor : 800/ 11242 .Skr/2021

Berdasarkan surat Rekomendasi Penelitian dari Kepala Badan Penelitian dan Pengembangan Kota Medan Nomor : 070/2353/Balithbang/2021 Tanggal 02 Juni 2021, dengan ini menerangkan:

Nama : Anggi Dessisliya
NIM : 0702171042
Program Studi : Sistem Informasi
Lokasi Penelitian : Dinas Pendidikan Kota Medan
Judul : "Sistem Informasi Geografis Sekolah Kota Medan Menggunakan Algoritma Haversine"
Penanggung Jawab : Wakil Dekan I Fakultas Sains Dan Teknologi Universitas Islam Negeri Sumatera Utara Medan

Benar, telah diterima melaksanakan Penelitian di Dinas Pendidikan Kota Medan selama, 1 (Satu) Bulan terhitung dari tanggal 16 Juni 2021

Demikian disampaikan untuk dapat dipergunakan sebagaimana mestinya.

Medan, Juni 2021


KEPALA DINAS PENDIDIKAN
KOTA MEDAN


ADLAN, S.Pd, MM
PEMBINA
NIP.19721208 200701 1 025

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

Lampiran 2. Form Hasil Wawancara

Judul Skripsi : Sistem Informasi Geografis Sekolah Di Kota Medan
Menggunakan Algoritma *Haversine*

Nama Penulis : Anggi Dessisiliya

NIM : 0702171042

Nama Narasumber : Taruli Simanulang, S.Sos, M.Pd

Jabatan : Kepala Sub Bagian Penyelenggara Tugas Pembantuan

Tanggal Wawancara : 19 Juli 2021

No	Pertanyaan	Jawaban
1.	Apakah Dinas Pendidikan Kota Medan saat ini sudah memiliki sistem informasi geografis terkait tentang sekolah di Kota Medan?	Untuk saat ini belum ada sistem terkait lokasi geografis sekolah di Kota Medan pada Dinas Pendidikan Kota Medan.
2.	Bagaimanakah biasanya masyarakat jika ingin mengetahui lokasi-lokasi serta informasi-informasi terkait sekolah di Kota Medan?	Biasanya masyarakat yang ingin mengetahui keberadaan lokasi-lokasi sekolah dan informasinya sering bertanya langsung kepada pihak Dinas Pendidikan Kota Medan, bertanya ke orang - orang sekitar, atau dengan melihat di <i>internet</i> untuk pencarian lokasi sekolah serta informasi terkait sekolah tersebut.
3	Apakah dengan cara tersebut terdapat kendala pada dalam pelaksanaan prosesnya?	Kendala yang dirasakan diantaranya: 1. Data lokasi sekolah serta informasi yang ada di Dinas Pendidikan Kota Medan belum terintegrasi ke dalam sistem informasi geografis. 2. Masyarakat yang ingin mengetahui lokasi harus bertanya secara langsung ke pihak-pihak terkait dan hal itu tidak efektif dan efisien baik dalam segi waktu

		maupun biaya. 3. Sistem manual tidak dapat memberitahukan sekolah terdekat dari posisi pengguna (orang yang mencari sekolah). 4. Tidak memberikan informasi terkait sekolah dengan jelas dan terperinci. 5. Tidak memberikan rute terdekat yang dapat ditempuh dari lokasi pencari sekolah dengan sekolah tujuan.
4.	Apakah ada harapan untuk penyelesaian kendala tersebut misalnya dengan pembuatan sistem informasi geografs yang dapat membantu penyelesaiannya kendala?	Jelas ada harapan untuk menyelesaikan dan lebih mempermudah proses pencarian lokasi sekolah dan terlebih lagi dapat memberikan informasi terkait tiap-tiap sekolah yang ada di Kota Medan.. Sistem tersebut sangat membantu masyarakat yang ingin lebih tahu mengenai informasi-informasi sekolah khususnya lokasi-lokasi sekolah yang ingin dituju.

Medan, 19 Juli 2021

Kepala SubBag Penyelenggara Tugas Pembantuan

UNIVERSITAS SUMATERA UTARA MEDAN

TARUti Cimanullaga, Sks. MPd.

NIP. 19690109 198903 1004

Lampiran 3. Form Black-box Testing

Form Black-box Testing

Tanggal Pengujian : 02 November 2021
 Nama Aplikasi : SIG Pencarian Sekolah Terdekat di Kota Medan
 Penguji : Taruli Simanulang, S.Sos, M.Pd
 Jabatan : Kepala Sub Bagian Penyelenggara Tugas Pembantuan Dinas Pendidikan Kota Medan

1. Pengujian Form Login Admin

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	
				Sesuai	Tidak Sesuai
1	Username tidak diisi kemudian klik login	Username: (Kosong)	Sistem menampilkan pesan username tidak boleh kosong	✓	
2	Password tidak diisi kemudian klik login	Password: (Kosong)	Sistem menampilkan pesan password tidak boleh kosong	✓	
3	Mengisi username dan Password tidak sesuai kemudian klik login	Username: (aaaa) Password: (a123)	Sistem akan menampilkan pesan "login gagal, mohon periksa username atau password anda"	✓	
4	Mengisi username dan Password sesuai kemudian klik login	Nama: (admin) Password : (admin)	Sistem menerima akses login dan menampilkan halaman dashboard user	✓	

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

2. Pengujian Menu Tambah Data Sekolah

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	
				Sesuai	Tidak Sesuai
1	Tambah Data pada Menu Data Sekolah	Button : Tambah Data	Sistem menampilkan Form pengisian data sekolah	✓	
2	Input Data Sekolah pada Menu Data Sekolah	Semua Textbox : Kosong	Sistem menolak dan menampilkan pesan "Tidak Boleh Kosong"	✓	
3	Input Data Sekolah pada Menu Data Sekolah	Sekolah Jenjang Akreditasi Prestasi Info PPDB Info Biaya Alamat Deskripsi Kontak Email Latitude Longitude Gambar	Sistem menerima inputan dan menyimpan data ke dalam database dan data ditampilkan pada menu data sekolah.	✓	
4	Melihat Info Detail Sekolah	Button : Detail	Sistem menampilkan informasi detail tentang sekolah yang dipilih	✓	

3. Pengujian Pencarian Sekolah Terdekat oleh *User*

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	
				Sesuai	Tidak Sesuai
1	<i>User</i> masuk ke dalam halaman web	Halaman beranda <i>user</i>	Sistem menampilkan peta geografis dan marker untuk sekolah.	✓	
2	<i>User</i> menginput data lokasi awal dan tujuan	Lokasi awal: "Lokasi saat ini" Lokasi Tujuan: "Al-Fithriah" Button: "Hitung Jarak"	Sistem menghitung jarak dan menampilkan informasi hasil jarak 0,27 Km dan menampilkan jarak untuk sekolah lainnya dan diurutkan secara Ascending. Sistem juga menampilkan rute menuju lokasi tujuan.	✓	
3	Menampilkan informasi sekolah	Klik Marker Merah Atau Biru	Sistem menampilkan informasi Sekolah, dan terdapat button selengkapnya yang memuat detail informasi sekolah.	✓	

Medan, November 2021

Kepala SubBag Penyelenggara Tugas Pembantuan

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN

(Taruli Simanungkal, S.Sos, M.Pd)

NIP. 19690109 198903 1004

Form Black-box Testing

Tanggal Pengujian : 01 November 2021
 Nama Aplikasi : SIG Pencarian Sekolah Terdekat di Kota Medan
 Penguji : Ilhamsyah, S.Kom
 Jabatan : Data Scientist Business Intelligence Bukalapak

1. Pengujian Form Login Admin

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	
				Sesuai	Tidak Sesuai
1	Username tidak diisi kemudian klik login	Username: (Kosong)	Sistem menampilkan pesan username tidak boleh kosong	✓	
2	Password tidak diisi kemudian klik login	Password: (Kosong)	Sistem menampilkan pesan password tidak boleh kosong	✓	
3	Mengisi username dan Password tidak sesuai kemudian klik login	Username: (aaaaa) Password: (a123)	Sistem akan menampilkan pesan "login gagal, mohon periksa username atau password anda"	✓	
4	Mengisi username dan Password sesuai kemudian klik login	Nama: (admin) Password: (admin)	Sistem menerima akses login dan menampilkan halaman dashboard user	✓	

UNIVERSITAS ISLAM NEGERI
 SUMATERA UTARA MEDAN

2. Pengujian Menu Tambah Data Sekolah

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	
				Sesuai	Tidak Sesuai
1	Tambah Data pada Menu Data Sekolah	Button : Tambah Data	Sistem menampilkan Form pengisian data sekolah	✓	
2	Input Data Sekolah pada Menu Data Sekolah	Semua Textbox : Kosong	Sistem menolak dan menampilkan pesan "Tidak Boleh Kosong"	✓	
3	Input Data Sekolah pada Menu Data Sekolah	Sekolah Jenjang Akreditasi Prestasi Info PPDB Info Biaya Alamat Deskripsi Kontak Email Latitude Longitude Gambar	Sistem menerima inputan dan menyimpan data ke dalam database dan data ditampilkan pada menu data sekolah.	✓	
4	Melihat Info Detail Sekolah	Button : Detail	Sistem menampilkan informasi detail tentang sekolah yang dipilih	✓	

3. Pengujian Pencarian Sekolah Terdekat oleh User

No	Skenario Pengujian	Test Case	Hasil yang Diharapkan	Hasil Pengujian	
				Sesuai	Tidak Sesuai
1	User masuk ke dalam halaman web	Halaman beranda user	Sistem menampilkan peta geografis dan marker untuk sekolah.	✓	
2	User menginput data lokasi awal dan tujuan	Lokasi awal: "Lokasi saat ini" Lokasi Tujuan: "Al-Fithriah" Button: "Hitung Jarak"	Sistem menghitung jarak dan menampilkan informasi hasil jarak 0,27 Km. dan menampilkan jarak untuk sekolah lainnya dan diurutkan secara Ascending. Sistem juga menampilkan rute menuju lokasi tujuan.	✓	
3	Menampilkan informasi sekolah	Klik Marker Merah Atau Biru	Sistem menampilkan informasi Sekolah, dan terdapat button selengkapnya yang memuat detail informasi sekolah.	✓	

Medan, November 2021

Data Scientist Business Intelligence Bukalapak

UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN



(Ilhamsyah, S.Kom)

Lampiran 4. Source Code

login.php

```
<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8" />
<title><?= $title ?></title>
<meta name="viewport"
content="width=device-width, initial-
scale=1.0">
<meta content="Premium Multipurpose
Admin & Dashboard Template"
name="description" />
<meta content="Themesdesign"
name="author" />
<!-- App favicon -->
<link rel="shortcut icon"
href="/images/favicon.ico">
<!-- Bootstrap Css -->
<link href="<?= base_url()
?>/css/bootstrap.min.css"
id="bootstrap-style" rel="stylesheet"
type="text/css" />
<!-- Icons Css -->
<link href="<?= base_url()
?>/css/icons.min.css" rel="stylesheet"
type="text/css" />
<!-- App Css-->
<link href="<?= base_url()
?>/css/app.min.css" id="app-style"
rel="stylesheet" type="text/css" />
</head>
<body class="bg-primary mt-5">
<div class="home-center">
<div class="home-desc-center">
<div class="container">
<div class="row justify-content-center
mt-3">
<div class="col-md-8 col-lg-6 col-xl-
5">
<div class="card">
<div class="card-body">
<div class="px-2 py-3">
<div class="text-center">
<a href="<?= site_url() ?>">
<h5>SISTEM INFORMASI GEOGRAFIS</h5>
<h5>SEKOLAH DI KOTA MEDAN</h5>
</a>
<h5 class="text-primary mb-2 mt-
4">SELAMAT DATANG !!!</h5>
</div>
<form class="form-horizontal mt-4 pt-
2" method="POST" action="<?=
site_url('login') ?>">
<?= csrf_field() ?>
<div class="mb-3">
<label for="username">Username</label>
<input type="text" class="form-control
<?= $validation->hasError('username')
? 'is-invalid' : '' ?>" id="username"
name="username" value="<?=
old('username') ?>"
placeholder="Masukan username">
<div
id="validationServerUsernameFeedback"
class="invalid-feedback">
<?= $validation->getError('username')
?>
</div>
</div>
<div class="mb-3">
<label
for="userpassword">Password</label>
<input type="password" class="form-
control <?= $validation-
>hasError('password') ? 'is-invalid' :
'' ?>" id="userpassword"
name="password" value="<?=
old('password') ?>"
placeholder="Masukan password">
<div
id="validationServerUsernameFeedback"
class="invalid-feedback">
```

```

<?= $validation-
>getError('password') ?>
</div>
</div>
<div>
<button class="btn btn-primary
w-100 waves-effect waves-
light" type="submit">Log
In</button>
</div>
<div class="mt-3">
<?= session()-
>getFlashdata('status') ?>
</div>
</form>
</div>
</div>
</div>
</div>
</div>
</div>
<!-- End Log In page -->
</div>
<!-- JAVASCRIPT -->
<script src="<?= base_url()
?>/libs/jquery/jquery.min.js">
</script>
<script src="<?= base_url()
?>/libs/bootstrap/js/bootstrap
.bundle.min.js"></script>
<script src="<?= base_url()
?>/libs/metismenu/metisMenu.mi
n.js"></script>
<script src="<?= base_url()
?>/libs/simplebar/simplebar.mi
n.js"></script>
<script src="<?= base_url()
?>/libs/node-
waves/waves.min.js"></script>
<script src="<?= base_url()
?>/js/app.js"></script>
</body>
</html>

```

Admin>DashboardController.php

```

<?php
namespace
App\Controllers\Admin;
use
App\Controllers\BaseController
;
use App\Models\SekolahModel;
use App\Models\UserModel;

```

```

use CodeIgniter\HTTP\URI;
class DashboardController
extends BaseController
{
protected $uri;
protected $sekolahModel;
protected $userModel;
public function __construct()
{
$this->sekolahModel = new
SekolahModel();
$this->userModel = new
UserModel();
$this->uri = new
URI(current_url());
}
/**
* index
*
* @return void
*/
public function index()
{
$data = [
'countSekolah' => count($this-
>sekolahModel->find()),
'countUser' => count($this-
>userModel->find()),
];
return
view('admin/dashboard/index',
[
'title' => 'Dashboard',
'data' => $data
]);
}
}

```

Admin>HaversineController.php

```

<?php
namespace
App\Controllers\Admin;
use
App\Controllers\BaseController
;
use App\Models\SekolahModel;
use
CodeIgniter\Config\Services;
class HaversineController
extends BaseController
{
protected $sekolahModel;
public function __construct()
{

```

```

$this->sekolahModel = new
SekolahModel();
}
public function index()
{
return
view('admin/haversine/index',
[
'title' => 'Haversine',
'sekolahRow' => $this-
>sekolahModel->findAll()
]);
}
/**
 * getMarkerList
 * untuk mendapatkan semua list
marker sekolah
 * @return void
 */
public function
getMarkerList()
{
if ($this->request->isAJAX())
{
echo json_encode($this-
>sekolahModel->findAll());
}
}
/**
 * getDistance
 * untuk mendapatkan jarak
menggunakan algo haversine
 * @desc const R = 6371e3; //
metres
const  $\phi_1$  = lat1 * Math.PI/180;
//  $\phi$ ,  $\lambda$  in radians
const  $\phi_2$  = lat2 * Math.PI/180;
const  $\phi$  = (lat2-lat1) *
Math.PI/180;
const  $\lambda$  = (lon2-lon1) *
Math.PI/180;
const a = Math.sin( $\phi/2$ ) *
Math.sin( $\phi/2$ ) +
Math.cos( $\phi_1$ ) * Math.cos( $\phi_2$ ) *
Math.sin( $\lambda/2$ ) *
Math.sin( $\lambda/2$ );
const c = 2 *
Math.atan2(Math.sqrt(a),
Math.sqrt(1-a));
const d = R * c; // in metres
 * @return void
 */
public function getDistance()
{

```

```

if ($this->request->isAJAX())
{
$lok_awal = $this->request-
>getVar('lokasi_awal');
$lok_akhir = $this->request-
>getVar('lokasi_akhir');
if (is_array($lok_awal)) {
$toRows = $this->sekolahModel-
>findAll();
$form_lat = $lok_awal['lat'];
$form_lng = $lok_awal['lng'];
} else {
$toRows = $this->sekolahModel-
>whereNotIn('id',
[$lok_awal])->findAll();
$formRow = $this-
>sekolahModel-
>find($lok_awal);
$form_lat = $formRow-
>latitude;
$form_lng = $formRow-
>longitude;
}
$toRow = $this->sekolahModel-
>find($lok_akhir);
$distanceFromStart = [];
foreach ($toRows as $to) :
$distanceFromStart[] = [
'id' => $to->id,
'sekolah' => $to->sekolah,
'distance' => $this-
>haversineGreatCircleDistance(
$form_lat, $form_lng, $to-
>latitude, $to->longitude)
];
endforeach;
$column =
array_column($distanceFromStar
t, 'distance');
array_multisort($column,
SORT_ASC, $distanceFromStart);
echo json_encode([
'res' => $this-
>haversineGreatCircleDistance(
$form_lat, $form_lng, $toRow-
>latitude, $toRow->longitude),
'coordinates' => [
[$form_lat, $form_lng],
[$toRow->latitude, $toRow-
>longitude]
],
'koordinat_awal' => ['lat' =>
$form_lat, 'lng' =>
$form_lng],

```

```

'koordinat_akhir' => ['lat' =>
$toRow->latitude, 'lng' =>
$toRow->longitude],
'distanceFromStart' =>
$distanceFromStart]);
}
}/**
 * haversineGreatCircleDistance
 *
 * @param mixed $latitudeFrom
 * @param mixed $longitudeFrom
 * @param mixed $latitudeTo
 * @param mixed $longitudeTo
 * @param mixed $earthRadius
 * @return void
 */
public function
haversineGreatCircleDistance($
latitudeFrom, $longitudeFrom,
$latitudeTo, $longitudeTo,
$earthRadius = 6371)
{
    $latFrom =
deg2rad($latitudeFrom);
    $lonFrom =
deg2rad($longitudeFrom);
    $latTo = deg2rad($latitudeTo);
    $lonTo =
deg2rad($longitudeTo);
    $lat = $latTo - $latFrom;
    $long = $lonTo - $lonFrom;
    $a = pow(sin($lat / 2), 2) +
cos($latFrom) * cos($latTo) *
pow(sin($long / 2), 2);
    $c = 2 * asin(sqrt($a));
    return round($c *
$earthRadius, 2);
}
}

```

Admin>RuteController.php

```

<?php
namespace
App\Controllers\Admin;
use
App\Models\DetailRuteModel;
use App\Models\RuteModel;
use
CodeIgniter\Config\Services;
use
CodeIgniter\RESTful\ResourceCo
ntroller;
class RuteController extends
ResourceController

```

```

{
protected $ruteModel;
protected $detailRuteModel;
public function __construct()
{
    $this->ruteModel = new
RuteModel();
    $this->detailRuteModel = new
DetailRuteModel();
}
/**
 * Return an array of resource
objects, themselves in array
format
 *
 * @return mixed
 */
public function index()
{
    $pager =
\Config\Services::pager();
    $keyword = $this->request-
>getVar('keyword');
    if ($keyword) {
        $ruteModel = $this->ruteModel-
>search($keyword);
    } else {
        $ruteModel = $this->ruteModel;
    }
    return
view('admin/rute/index', [
'title' => 'Rute',
'table_desc' => 'Berikut
adalah data rute yang
terdaftar',
'ruteRows' => $ruteModel-
>orderBy('id', 'desc')-
>paginate(10, 'rute'),
'page' => ($this->request-
>getVar('page_rute') ? $this-
>request->getVar('page_rute')
: 1),
'pager' => $ruteModel->pager
]);
}
/**
 * Return the properties of a
resource object
 *
 * @return mixed
 */
public function show($id =
null)
{
    //
}
}

```

```

/**
 * Return a new resource
 * object, with default
 * properties
 *
 * @return mixed
 */
public function new()
{
    return view('admin/rute/new',
    [
        'title' => 'Rute',
        'validation' =>
        Services::validation()
    ]);
    // https://www.movable-
    // type.co.uk/scripts/latlong.htm
    //
    // Haversine
    // formula: a =
    //  $\sin^2(\phi/2) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2(\lambda/2)$ 
    //  $c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$ 
    //
    // d = R · c
    // where  $\phi$  is latitude,
    //  $\lambda$  is longitude, R is earth's
    // radius (mean radius =
    // 6,371km);
    // note that angles need to be
    // in radians to pass to trig
    // functions!
    // JavaScript:
    // const R = 6371e3; // metres
    // const  $\phi_1$  = lat1 *
    // Math.PI/180; //  $\phi$ ,  $\lambda$  in
    // radians
    // const  $\phi_2$  = lat2 *
    // Math.PI/180;
    // const  $\phi$  = (lat2-lat1) *
    // Math.PI/180;
    // const  $\lambda$  = (lon2-lon1) *
    // Math.PI/180;
    // const a = Math.sin( $\phi/2$ )
    // * Math.sin( $\phi/2$ ) +
    // Math.cos( $\phi_1$ ) *
    // Math.cos( $\phi_2$ ) *
    // Math.sin( $\lambda/2$ )
    // * Math.sin( $\lambda/2$ );
    // const c = 2 *
    // Math.atan2(Math.sqrt(a),
    // Math.sqrt(1-a));
    // const d = R * c; // in
    // metres
    // $R = 6371000;

```

```

// $lat1 = -0.493005 * M_PI /
// 180;
// dd($lat1);
}
/**
 * Create a new resource
 * object, from "posted"
 * parameters
 *
 * @return mixed
 */
public function create()
{
    function haversine()
    {
        // https://www.movable-
        // type.co.uk/scripts/latlong.htm
        //
        // Haversine
        // formula: a =
        //  $\sin^2(\phi/2) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2(\lambda/2)$ 
        //  $c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$ 
        //
        // d = R · c
        // where  $\phi$  is latitude,
        //  $\lambda$  is longitude, R is earth's
        // radius (mean radius =
        // 6,371km);
        // note that angles need to be
        // in radians to pass to trig
        // functions!
        // JavaScript:
        // const R = 6371e3; // metres
        // const  $\phi_1$  = lat1 *
        // Math.PI/180; //  $\phi$ ,  $\lambda$  in
        // radians
        // const  $\phi_2$  = lat2 *
        // Math.PI/180;
        // const  $\phi$  = (lat2-lat1) *
        // Math.PI/180;
        // const  $\lambda$  = (lon2-lon1) *
        // Math.PI/180;
        // const a = Math.sin( $\phi/2$ )
        // * Math.sin( $\phi/2$ ) +
        // Math.cos( $\phi_1$ ) *
        // Math.cos( $\phi_2$ ) *
        // Math.sin( $\lambda/2$ )
        // * Math.sin( $\lambda/2$ );
        // const c = 2 *
        // Math.atan2(Math.sqrt(a),
        // Math.sqrt(1-a));
        // const d = R * c; // in
        // metres
        $R = 6371000;

```

```

}
if ($this->request->isAJAX())
{
    $rute_id = $this->ruteModel-
    >insert([
        'jalan' => $this->request-
        >getVar('jalan')
    \]);
    $graf = $this->request-
    >getVar('graf');
    for ($i = 0; $i <
    count($graf); $i++) {
        $this->detailRuteModel-
        >insert([
            'rute_id' => $rute_id,
            'latitude_awal' =>
            $graf[$i]['latitude_awal'],
            'longitude_awal' =>
            $graf[$i]['longitude_awal'],
            'latitude_akhir' =>
            $graf[$i]['latitude_akhir'],
            'longitude_akhir' =>
            $graf[$i]['longitude_akhir'],
        \]);
    }
}
/**
 * Return the editable
 * properties of a resource
 * object
 *
 * @return mixed
 */
public function edit($id =
null)
{
    //
}

/**
 * Add or update a model
 * resource, from "posted"
 * properties
 *
 * @return mixed
 */
public function update($id =
null)
{
    //
}

/**
 * Delete the designated
 * resource object from the model
 *

```

```

* @return mixed
*/
public function delete($id =
null)
{
    //
}
}

```

Admin>SekolahController.php

```

<?php
namespace
App\Controllers\Admin;
use
Ozdemir\Datatables\Datatables;
use
App\Controllers\BaseController
;
use
CodeIgniter\Config\Services;
use
Ozdemir\Datatables\DB\Codeigni
terAdapter;
class SekolahController
extends BaseController
{
    protected $sekolahModel;
    public function __construct()
    {
        $this->sekolahModel = new
        \App\Models\SekolahModel();
    }
    /**
     * Return an array of resource
     * objects, themselves in array
     * format
     *
     * @return mixed
     */
    public function index()
    {
        $pager =
        \Config\Services::pager();
        $keyword = $this->request-
        >getVar('keyword');
        if ($keyword) {
            $sekolahModel = $this-
            >sekolahModel-
            >search($keyword);
        } else {
            $sekolahModel = $this-
            >sekolahModel;
        }
    }
}

```

```

return
view('admin/sekolah/index', [
    'title' => 'Sekolah',
    'table_desc' => 'Berikut
adalah data sekolah yang
terdaftar',
    'sekolahRow' => $sekolahModel-
>orderBy('id', 'desc')-
>paginate(10, 'sekolah'),
    'page' => ($this->request-
>getVar('page_sekolah') ?
$this->request-
>getVar('page_sekolah') : 1),
    'pager' => $sekolahModel-
>pager
]);
/**
 * Return the properties of a
resource object
 *
 * @return mixed
 */
public function show($id =
null)
{
    return
view('admin/sekolah/show', [
    'title' => 'Detail Sekolah',
    'sekolahRow' => $this-
>sekolahModel->find($id)
]);
}
/**
 * Return a new resource
object, with default
properties
 *
 * @return mixed
 */
public function new()
{
    return
view('admin/sekolah/new', [
    'title' => 'Sekolah',
    'validation' =>
Services::validation()
]);
}
/**
 * Create a new resource
object, from "posted"
parameters
 *
 * @return mixed
 */

```

```

public function create()
{
    if (!$this->validate([
        'sekolah' => [
            'label' => 'Sekolah',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
boleh kosong'
            ]
        ],
        'jenjang' => [
            'label' => 'Jenjang',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
boleh kosong'
            ]
        ],
        'akreditasi' => [
            'label' => 'Akreditasi',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
boleh kosong'
            ]
        ],
        'prestasi' => [
            'label' => 'Prestasi',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
boleh kosong'
            ]
        ],
        'info_PPDB' => [
            'label' => 'Info PPDB',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
boleh kosong'
            ]
        ],
        'info_biaya' => [
            'label' => 'Info Biaya',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
boleh kosong'
            ]
        ],
        'deskripsi' => [
            'label' => 'Deskripsi',
            'rules' => 'required',
            'errors' => [

```

```

'required' => '{field} tidak
boleh kosong'
],
'alamat' => [
'label' => 'Alamat',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'kontak' => [
'label' => 'Kontak',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'email' => [
'label' => 'Email',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'latitude' => [
'label' => 'Latitude',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'longitude' => [
'label' => 'Longitude',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
])) {
$validation =
Services::validation();
return redirect()-
>to('admin/sekolah/new')-
>withInput();
}
$file = $this->request-
>getFile('gambar');
if ($file->getError() == 4) {
$nama_sampul = 'default.png';
} else {

```

```

$nama_sampul = $file-
>getRandomName();
$file->move('images/sekolah',
$nama_sampul);
}
$this->sekolahModel->save([
'sekolah' => $this->request-
>getVar('sekolah'),
'jenjang' => $this->request-
>getVar('jenjang'),
'akreditasi' => $this-
>request-
>getVar('akreditasi'),
'prestasi' => $this->request-
>getVar('prestasi'),
'info_PPDB' => $this->request-
>getVar('info_PPDB'),
'info_biaya' => $this-
>request-
>getVar('info_biaya'),
'alamat' => $this->request-
>getVar('alamat'),
'deskripsi' => $this->request-
>getVar('deskripsi'),
'kontak' => $this->request-
>getVar('kontak'),
'email' => $this->request-
>getVar('email'),
'gambar' => $nama_sampul,
'latitude' => $this->request-
>getVar('latitude'),
'longitude' => $this->request-
>getVar('longitude'),
]);
session()-
>setFlashdata('status',
alert('success', 'Data
berhasil disimpan'));
return redirect()-
>to('admin/sekolah');
}
/**
 * Return the editable
 * properties of a resource
 * object
 * @return mixed
 */
public function edit($id =
null)
{
return
view('admin/sekolah/edit', [
'title' => 'Sekolah',
'validation' =>
Services::validation(),

```

```

'sekolahRow' => $this->sekolahModel->find($id)
]);
}
/**
 * Add or update a model
 * resource, from "posted"
 * properties
 *
 * @return mixed
 */
public function update($id = null)
{
    if (!$this->validate([
        'sekolah' => [
            'label' => 'Sekolah',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak boleh kosong'
            ]
        ],
        'jenjang' => [
            'label' => 'Jenjang',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak boleh kosong'
            ]
        ],
        'akreditasi' => [
            'label' => 'Akreditasi',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak boleh kosong'
            ]
        ],
        'prestasi' => [
            'label' => 'Prestasi',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak boleh kosong'
            ]
        ],
        'info_PPDB' => [
            'label' => 'Info PPDB',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak boleh kosong'
            ]
        ],
        'info_biaya' => [

```

```

'label' => 'Info Biaya',
'rules' => 'required',
'errors' => [
    'required' => '{field} tidak boleh kosong'
],
'deskripsi' => [
    'label' => 'Deskripsi',
    'rules' => 'required',
    'errors' => [
        'required' => '{field} tidak boleh kosong'
    ]
],
'kontak' => [
    'label' => 'Kontak',
    'rules' => 'required',
    'errors' => [
        'required' => '{field} tidak boleh kosong'
    ]
],
'latitude' => [
    'label' => 'Latitude',
    'rules' => 'required',
    'errors' => [
        'required' => '{field} tidak boleh kosong'
    ]
],
'longitude' => [
    'label' => 'Longitude',
    'rules' => 'required',
    'errors' => [
        'required' => '{field} tidak boleh kosong'
    ]
]) {
    $validation = Services::validation();
    return redirect()->to('admin/sekolah/' . $id . '/edit')->withInput();
}
$file = $this->request->getFile('gambar');
if ($file->getError() == 4) {
    $nama_sampul = $this->request->getVar('_gambar');
} else {
    $nama_sampul = $file->getRandomName();
    $file->move('images/sekolah', $nama_sampul);
}

```

```

$_gambar = $this->request-
>getVar('_gambar');
if ($_gambar != 'default.png'
&& $_gambar != '') {
@unlink('images/sekolah/' .
$_gambar);
}
}
$this->sekolahModel-
>update($id, [
'sekolah' => $this->request-
>getVar('sekolah'),
'jenjang' => $this->request-
>getVar('jenjang'),
'akreditasi' => $this-
>request-
>getVar('akreditasi'),
'prestasi' => $this->request-
>getVar('prestasi'),
'info_PPDB' => $this->request-
>getVar('info PPDB'),
'info_biaya' => $this-
>request-
>getVar('info biaya'),
'alamat' => $this->request-
>getVar('alamat'),
'deskripsi' => $this->request-
>getVar('deskripsi'),
'kontak' => $this->request-
>getVar('kontak'),
'email' => $this->request-
>getVar('email'),
'gambar' => $nama_sampul,
'latitude' => $this->request-
>getVar('latitude'),
'longitude' => $this->request-
>getVar('longitude'),
]);
session()-
>setFlashdata('status',
alert('success', 'Data
berhasil diperbarui'));
return redirect()-
>to('admin/sekolah');
}
/**
 * Delete the designated
resource object from the model
 *
 * @return mixed
 */
public function delete($id =
null)
{
$sekolah = $this-
>sekolahModel->find($id);

```

```

if ($sekolah->gambar !=
'default.png' && $sekolah-
>gambar != '') {
    unlink('images/sekolah/' .
    $sekolah->gambar);
}
$this->sekolahModel-
>delete($id);
session()-
>setFlashdata('status',
alert('success', 'Sekolah
berhasil dihapus'));
return redirect()-
>to('/admin/sekolah');
}

```

Admin>UserController.php

```

<?php
namespace
App\Controllers\Admin;
use
CodeIgniter\Config\Services;
use
CodeIgniter\RESTful\ResourceCo
ntroller;
class UserController extends
ResourceController
{
protected $userModel;
public function __construct()
{
    helper(['alert helper']);
    session();
    $this->userModel = new
    \App\Models\UserModel();
}
/**
 * Return an array of resource
objects, themselves in array
format
 *
 * @return mixed
 */
public function index()
{
    $pager =
    \Config\Services::pager();
    $keyword = $this->request-
    >getVar('keyword');
    if ($keyword) {
        $userModel = $this->userModel-
        >search($keyword);
    } else {
        $userModel = $this->userModel;
    }
}

```

```

}
return
view('admin/user/index', [
'title' => 'Admin',
'table_desc' => 'Berikut
adalah data admin yang
terdaftar',
'userRows' => $userModel-
>orderBy('id', 'desc')-
>paginate(10, 'user'),
'page' => ($this->request-
>getVar('page_user') ? $this-
>request->getVar('page_user')
: 1),
'pager' => $userModel->pager
]);
/**
 * Return the properties of a
resource object
 *
 * @return mixed
 */
public function show($id =
null)
{
return view('admin/user/show',
[
'title' => 'Detail Admin',
'userRow' => $this->userModel-
>find($id)
]);
}
/**
 * Return a new resource
object, with default
properties
 *
 * @return mixed
 */
public function new()
{
return view('admin/user/new',
[
'title' => 'Admin',
'validation' =>
Services::validation()
]);
}
/**
 * Create a new resource
object, from "posted"
parameters
 *
 * @return mixed
 */

```

```

public function create()
{
if (!$this->validate([
'username' => [
'label' => 'username',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'password' => [
'label' => 'Password',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
]
])) {
$validation =
Services::validation();
return redirect()-
>to('admin/user/new')-
>withInput();
}
$file = $this->request-
>getFile('gambar');
if ($file->getError() == 4) {
$gambar = 'default.png';
} else {
$gambar = $file-
>getRandomName();
$file->move('images/users',
$gambar);
}
$this->userModel->save([
'username' => $this->request-
>getVar('username'),
'password' => $this->request-
>getVar('password'),
'gambar' => $gambar,
]);
session()-
>setFlashdata('status',
alert('success', 'Data
berhasil disimpan'));
return redirect()-
>to('admin/user');
}
/**
 * Return the editable
properties of a resource
object
 *
 * @return mixed
 */

```

```

*/
public function edit($id =
null)
{
return view('admin/user/edit',
[
'title' => 'Sekolah',
'validation' =>
Services::validation(),
'userRow' => $this->userModel-
>find($id)
]);
}
/**
* Add or update a model
resource, from "posted"
properties
*
* @return mixed
*/
public function update($id =
null)
{
if (!$this->validate([
'username' => [
'label' => 'username',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'password' => [
'label' => 'Password',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
])) {
$validation =
Services::validation();
return redirect()-
>to('admin/user/' . $id .
'/edit')->withInput();
}
$file = $this->request-
>getFile('gambar');
if ($file->getError() == 4) {
$gambar = $this->request-
>getVar('_gambar');
} else {
$gambar = $file-
>getRandomName();

```

```

$file->move('images/users',
$gambar);
$_gambar = $this->request-
>getVar('_gambar');
if ($_gambar != 'default.png'
&& $_gambar != '') {
@unlink('images/users/' .
$_gambar);
}
$this->userModel->update($id,
[
'username' => $this->request-
>getVar('username'),
'password' => $this->request-
>getVar('password'),
'gambar' => $gambar,
]);
session()-
>setFlashdata('status',
alert('success', 'Data
berhasil diperbarui'));
return redirect()-
>to('admin/user');
}
/**
* Delete the designated
resource object from the model
*
* @return mixed
*/
public function delete($id =
null)
{
$userRow = $this->userModel-
>find($id);
if ($userRow->gambar !=
'default.png' && $userRow-
>gambar != '') {
unlink('images/users/' .
$userRow->gambar);
}
$this->userModel->delete($id);
session()-
>setFlashdata('status',
alert('success', 'Sekolah
berhasil dihapus'));
return redirect()-
>to('/admin/user');
}
}

```

User>HaversineController.php

<?php

```

namespace
App\Controllers\Admin;
use
App\Controllers\BaseController
;
use App\Models\SekolahModel;
use
CodeIgniter\Config\Services;
class HaversineController
extends BaseController
{
protected $sekolahModel;
public function __construct()
{
$this->sekolahModel = new
SekolahModel();
}
public function index()
{
return
view('admin/haversine/index',
[
'title' => 'Haversine',
'sekolahRow' => $this-
>sekolahModel->findAll()
]);
}
/**
* getMarkerList
* untuk mendapatkan semua list
marker sekolah
* @return void
*/
public function
getMarkerList()
{
if ($this->request->isAJAX())
{
echo json_encode($this-
>sekolahModel->findAll());
}
}
/**
* getDistance
* untuk mendapatkan jarak
menggunakan algo haversine
* @desc const R = 6371e3; //
metres
const  $\phi_1$  = lat1 * Math.PI/180;
//  $\phi$ ,  $\lambda$  in radians
const  $\phi_2$  = lat2 * Math.PI/180;
const  $\phi$  = (lat2-lat1) *
Math.PI/180;
const  $\lambda$  = (lon2-lon1) *
Math.PI/180;

```

```

const a = Math.sin( $\phi/2$ ) *
Math.sin( $\phi/2$ ) +
Math.cos( $\phi_1$ ) * Math.cos( $\phi_2$ ) *
Math.sin( $\lambda/2$ ) *
Math.sin( $\lambda/2$ );
const c = 2 *
Math.atan2(Math.sqrt(a),
Math.sqrt(1-a));
const d = R * c; // in metres
* @return void
*/
public function getDistance()
{
if ($this->request->isAJAX())
{
$lok_awal = $this->request-
>getVar('lokasi_awal');
$lok_akhir = $this->request-
>getVar('lokasi_akhir');
if (is_array($lok_awal)) {
$toRows = $this->sekolahModel-
>findAll();
$form_lat = $lok_awal['lat'];
$form_lng = $lok_awal['lng'];
} else {
$toRows = $this->sekolahModel-
>whereNotIn('id',
[$lok_awal])->findAll();
$formRow = $this-
>sekolahModel-
>find($lok_awal);
$form_lat = $formRow-
>latitude;
$form_lng = $formRow-
>longitude;
}
$toRow = $this->sekolahModel-
>find($lok_akhir);
$distanceFromStart = [];
foreach ($toRows as $sto) :
$distanceFromStart[] = [
'id' => $sto->id,
'sekolah' => $sto->sekolah,
'distance' => $this-
>haversineGreatCircleDistance(
$form_lat, $form_lng, $sto-
>latitude, $sto->longitude)
];
endforeach;
$column =
array_column($distanceFromStar
t, 'distance');
array_multisort($column,
SORT_ASC, $distanceFromStart);
echo json_encode([

```

```

'res' => $this->haversineGreatCircleDistance(
    $form_lat, $form_lng, $toRow->latitude, $toRow->longitude),
'coordinates' => [
    [$form_lat, $form_lng],
    [$toRow->latitude, $toRow->longitude]
],
'koordinat_awal' => ['lat' => $form_lat, 'lng' => $form_lng],
'koordinat_akhir' => ['lat' => $toRow->latitude, 'lng' => $toRow->longitude],
'distanceFromStart' => $distanceFromStart
]);
}
}
/**
 * haversineGreatCircleDistance
 *
 * @param mixed $latitudeFrom
 * @param mixed $longitudeFrom
 * @param mixed $latitudeTo
 * @param mixed $longitudeTo
 * @param mixed $earthRadius
 * @return void
 */
public function haversineGreatCircleDistance($latitudeFrom, $longitudeFrom, $latitudeTo, $longitudeTo, $earthRadius = 6371000)
{
    $latFrom = deg2rad($latitudeFrom);
    $lonFrom = deg2rad($longitudeFrom);
    $latTo = deg2rad($latitudeTo);
    $lonTo = deg2rad($longitudeTo);
    $latDelta = $latTo - $latFrom;
    $lonDelta = $lonTo - $lonFrom;
    $angle = 2 *
        asin(sqrt(pow(sin($latDelta / 2), 2) + cos($latFrom) *
            cos($latTo) *
            pow(sin($lonDelta / 2), 2)));
    return round($angle *
        $earthRadius);
}
}

```

User>HomeController.php

```

<?php
namespace
App\Controllers\User;
use CodeIgniter\HTTP\URI;
use
App\Controllers\BaseController;
;
use App\Models\SekolahModel;
class HomeController extends
BaseController
{
    protected $uri;
    protected $sekolahModel;
    public function __construct()
    {
        $this->sekolahModel = new
        SekolahModel();
    }
    /**
     * index
     *
     * @return void
     */
    public function index()
    {
        return view('user/home/index',
            [
                'title' => 'Beranda',
                'sekolahRow' => $this->sekolahModel->findAll()
            ]);
    }
    /**
     * getMarkerList
     * untuk mendapatkan semua list
     * marker sekolah
     * @return void
     */
    public function getMarkerList()
    {
        if ($this->request->isAJAX())
        {
            echo json_encode($this->sekolahModel->findAll());
        }
    }
    /**
     * getDistance
     * untuk mendapatkan jarak
     * menggunakan algo haversine
     * @desc const R = 6371e3; //
     * metres
     * const φ1 = lat1 * Math.PI/180;
     * // φ, λ in radians

```

```

const  $\phi$ 2 = lat2 * Math.PI/180;
const  $\phi$  = (lat2-lat1) *
Math.PI/180;
const  $\lambda$  = (lon2-lon1) *
Math.PI/180;
const a = Math.sin( $\phi$ /2) *
Math.sin( $\phi$ /2) +
Math.cos( $\phi$ 1) * Math.cos( $\phi$ 2) *
Math.sin( $\lambda$ /2) *
Math.sin( $\lambda$ /2);
const c = 2 *
Math.atan2(Math.sqrt(a),
Math.sqrt(1-a));
const d = R * c; // in metres
* @return void
*/
public function getDistance()
{
    if ($this->request->isAJAX())
    {
        $lok_awal = $this->request-
        >getVar('lokasi_awal');
        $lok_akhir = $this->request-
        >getVar('lokasi_akhir');
        if (is_array($lok_awal)) {
            $toRows = $this->sekolahModel-
            >findAll();
            $form_lat = $lok_awal['lat'];
            $form_lng = $lok_awal['lng'];
        } else {
            $toRows = $this->sekolahModel-
            >whereNotIn('id',
            [$lok_awal])>findAll();
            $formRow = $this-
            >sekolahModel-
            >find($lok_awal);
            $form_lat = $formRow-
            >latitude;
            $form_lng = $formRow-
            >longitude;
        }
        $toRow = $this->sekolahModel-
        >find($lok_akhir);
        $distanceFromStart = 0;
        foreach ($toRows as $to) :
            $distanceFromStart[] = [
                'id' => $to->id,
                'sekolah' => $to->sekolah,
                'distance' => $this-
                >haversineGreatCircleDistance(
                $form_lat, $form_lng, $to-
                >latitude, $to->longitude)
            ];
        endforeach;
    }
}

```

```

$column =
array_column($distanceFromStart,
'distance');
array_multisort($column,
SORT_ASC, $distanceFromStart);
echo json_encode([
    'res' => $this-
    >haversineGreatCircleDistance(
    $form_lat, $form_lng, $toRow-
    >latitude, $toRow->longitude),
    'coordinates' => [
        [$form_lat, $form_lng],
        [$toRow->latitude, $toRow-
        >longitude]
    ],
    'koordinat_awal' => ['lat' =>
    $form_lat, 'lng' =>
    $form_lng],
    'koordinat_akhir' => ['lat' =>
    $toRow->latitude, 'lng' =>
    $toRow->longitude],
    'distanceFromStart' =>
    $distanceFromStart
]);
}
}
/**
 * haversineGreatCircleDistance
 *
 * @param mixed $latitudeFrom
 * @param mixed $longitudeFrom
 * @param mixed $latitudeTo
 * @param mixed $longitudeTo
 * @param mixed $earthRadius
 * @return void
 */
public function
haversineGreatCircleDistance($
latitudeFrom, $longitudeFrom,
$latitudeTo, $longitudeTo,
$earthRadius = 6371)
{
    $latFrom =
deg2rad($latitudeFrom);
    $lonFrom =
deg2rad($longitudeFrom);
    $latTo = deg2rad($latitudeTo);
    $lonTo =
deg2rad($longitudeTo);
    $lat = $latTo - $latFrom;
    $long = $lonTo - $lonFrom;
    $a = pow(sin($lat / 2), 2) +
    cos($latFrom) * cos($latTo) *
    pow(sin($long / 2), 2);
    $c = 2 * asin(sqrt($a));
}

```

```

return round($c *
SearchRadius, 2);
}
}

```

User>SekolahController.php

```

<?php
namespace
App\Controllers\User;
use
Ozdemir\Datatables\Datatables;
use
App\Controllers\BaseController
;
use
CodeIgniter\Config\Services;
use
Ozdemir\Datatables\DB\Codeigni
terAdapter;
class SekolahController
extends BaseController
{
protected $sekolahModel;
public function __construct()
{
$this->sekolahModel = new
\App\Models\SekolahModel();
}
/**
* Return an array of resource
objects, themselves in array
format
*
* @return mixed
*/
public function index()
{
$pagin
=
\Config\Services::pager();
$keyword = $this->request-
>getVar('keyword');
if ($keyword) {
$sekolahModel = $this-
>sekolahModel-
>search($keyword);
} else {
$sekolahModel = $this-
>sekolahModel;
}
return
view('user/sekolah/index', [
'title' => 'Sekolah',

```

```

'table_desc' => 'Berikut
adalah data sekolah yang
terdaftar',
'sekolahRow' => $sekolahModel-
>orderBy('id', 'desc')-
>paginate(10, 'sekolah'),
'page' => ($this->request-
>getVar('page_sekolah') ?
$this->request-
>getVar('page_sekolah') : 1),
'pager' => $sekolahModel-
>pager
]);
}
/**
* Return the properties of a
resource object
*
* @return mixed
*/
public function show($id =
null)
{
return
view('user/sekolah/show', [
'title' => 'Detail Sekolah',
'sekolahRow' => $this-
>sekolahModel->find($id)
]);
}
/**
* Return a new resource
object, with default
properties
*
* @return mixed
*/
public function new()
{
return
view('admin/sekolah/new', [
'title' => 'Sekolah',
'validation' =>
Services::validation()
]);
}
/**
* Create a new resource
object, from "posted"
parameters
*
* @return mixed
*/
public function create()
{
if (!$this->validate([

```

```

'sekolah' => [
'label' => 'Sekolah',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'jenjang' => [
'label' => 'Jenjang',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'akreditasi' => [
'label' => 'Akreditasi',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'prestasi' => [
'label' => 'Prestasi',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'info_PPDB' => [
'label' => 'Info PPDB',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'info_biaya' => [
'label' => 'Info Biaya',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'deskripsi' => [
'label' => 'Deskripsi',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],

```

```

'alamat' => [
'label' => 'Alamat',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'kontak' => [
'label' => 'Kontak',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'email' => [
'label' => 'Email',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'latitude' => [
'label' => 'Latitude',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'longitude' => [
'label' => 'Longitude',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
])) {
$validation =
Services::validation();
return redirect()-
to('admin/sekolah/new')-
withInput();
}
$file = $this->request-
->getFile('gambar');
if ($file->getError() == 4) {
$nama_sampul = 'default.png';
} else {
$nama_sampul = $file-
->getRandomName();
$file->move('images/sekolah',
$nama_sampul);
}
}

```

```

}
$this->sekolahModel->save([
'sekolah' => $this->request-
>getVar('sekolah'),
'jenjang' => $this->request-
>getVar('jenjang'),
'akreditasi' => $this-
>request-
>getVar('akreditasi'),
'prestasi' => $this->request-
>getVar('prestasi'),
'info_PPDB' => $this->request-
>getVar('info_PPDB'),
'info_biaya' => $this-
>request-
>getVar('info biaya'),
'alamat' => $this->request-
>getVar('alamat'),
'deskripsi' => $this->request-
>getVar('deskripsi'),
'kontak' => $this->request-
>getVar('kontak'),
'email' => $this->request-
>getVar('email'),
'gambar' => $nama_sampul,
'latitude' => $this->request-
>getVar('latitude'),
'longitude' => $this->request-
>getVar('longitude'),
]);
session()-
>setFlashdata('status',
alert('success', 'Data
berhasil disimpan'));
return redirect()-
>to('admin/sekolah');
}
/**
 * Return the editable
properties of a resource
object
 *
 * @return mixed
 */
public function edit($id =
null)
{
return
view('admin/sekolah/edit', [
'title' => 'Sekolah',
'validation' =>
Services::validation(),
'sekolahRow' => $this-
>sekolahModel->find($id)
]);
}

```

```

/**
 * Add or update a model
resource, from "posted"
properties
 *
 * @return mixed
 */
public function update($id =
null)
{
if (!$this->validate([
'sekolah' => [
'label' => 'Sekolah',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'jenjang' => [
'label' => 'Jenjang',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'akreditasi' => [
'label' => 'Akreditasi',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'prestasi' => [
'label' => 'Prestasi',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'info_PPDB' => [
'label' => 'Info PPDB',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
],
],
'info biaya' => [
'label' => 'Info Biaya',
'rules' => 'required',
'errors' => [

```

```

'required' => '{field} tidak
boleh kosong'
],
'deskripsi' => [
'label' => 'Deskripsi',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'kontak' => [
'label' => 'Kontak',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'latitude' => [
'label' => 'Latitude',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
'longitude' => [
'label' => 'Longitude',
'rules' => 'required',
'errors' => [
'required' => '{field} tidak
boleh kosong'
]
],
])) {
$validation =
Services::validation();
return redirect()-
>to('admin/sekolah/' . $id .
'/edit')->withInput();
}
$file = $this->request-
>getFile('gambar');
if ($file->getError() == 4) {
$nama_sampul = $this->request-
>getVar('_gambar');
} else {
$nama_sampul = $file-
>getRandomName();
$file->move('images/sekolah',
$nama_sampul);
$gambar = $this->request-
>getVar('_gambar');

```

```

if ($gambar != 'default.png'
&& $gambar != '') {
@unlink('images/sekolah/' .
$gambar);
}
}
$this->sekolahModel-
>update($id, [
'sekolah' => $this->request-
>getVar('sekolah'),
'jenjang' => $this->request-
>getVar('jenjang'),
'akreditasi' => $this-
>request-
>getVar('akreditasi'),
'prestasi' => $this->request-
>getVar('prestasi'),
'info_PPDB' => $this->request-
>getVar('info_PPDB'),
'info_biaya' => $this-
>request-
>getVar('info_biaya'),
'alamat' => $this->request-
>getVar('alamat'),
'deskripsi' => $this->request-
>getVar('deskripsi'),
'kontak' => $this->request-
>getVar('kontak'),
'email' => $this->request-
>getVar('email'),
'gambar' => $nama_sampul,
'latitude' => $this->request-
>getVar('latitude'),
'longitude' => $this->request-
>getVar('longitude'),
]);
session()-
>setFlashdata('status',
alert('success', 'Data
berhasil diperbarui'));
return redirect()-
>to('admin/sekolah');
}
/**
 * Delete the designated
resource object from the model
 *
 * @return mixed
 */
public function delete($id =
null)
{
$sekolah = $this-
>sekolahModel->find($id);

```

```

if ($sekolah->gambar !=
'default.png' && $sekolah-
>gambar != '') {
    unlink('images/sekolah/' .
    $sekolah->gambar);
}
$this->sekolahModel->delete($id);
session()-
>setFlashdata('status',
alert('success', 'Sekolah
berhasil dihapus'));
return redirect()-
>to('/admin/sekolah');
}
}

```

app>Controllers>BaseController.php

```

<?php
namespace App\Controllers;
use CodeIgniter\Controller;
use
CodeIgniter\HTTP\CLIRequest;
use
CodeIgniter\HTTP\IncomingRequest;
use
CodeIgniter\HTTP\RequestInterface;
use
CodeIgniter\HTTP\ResponseInterface;
use Psr\Log\LoggerInterface;
/**
 * Class BaseController
 *
 * BaseController provides a
convenient place for loading
components
 * and performing functions
that are needed by all your
controllers.
 * Extend this class in any new
controllers:
 *
 * class Home extends
BaseController
 *
 * For security be sure to
declare any new methods as
protected or private.
 */
class BaseController extends
Controller
{

```

```

/**
 * Instance of the main Request
object.
 *
 * @var
CLIRequest|IncomingRequest
 */
protected $request;
/**
 * An array of helpers to be
loaded automatically upon
 * class instantiation. These
helpers will be available
 * to all other controllers
that extend BaseController.
 *
 * @var array
 */

```

```

protected $helpers =
['alert_helper',
'const_helper'];
 *
 * Constructor.
 */
public function
initController(RequestInterfac
e $request, ResponseInterfac
e $response, LoggerInterface
 $logger)
{
    // Do Not Edit This Line
    parent::initController($request,
    $response, $logger);
    // Preload any models,
libraries, etc, here.
    // E.g.: $this->session =
\Config\Services::session();
    session();
}
}

```

app>Controllers>LoginController.php

```

<?php
namespace App\Controllers;
use App\Models\UserModel;
use
CodeIgniter\RESTful\ResourceCo
ntroller;
use Config\Services;
class LoginController extends
ResourceController
{
    protected $userModel;
}

```

```

protected $helpers =
['alert_helper'];
public function __construct()
{
    session();
    $this->userModel = new
    UserModel();
}
/**
 * Return an array of resource
 objects, themselves in array
 format
 *
 * @return mixed
 */
public function index()
{
    return view('auth/login', [
        'title' => 'Login',
        'validation' =>
        Services::validation()
    ]);
}
/**
 * Create a new resource
 object, from "posted"
 parameters
 *
 * @return mixed
 */
public function create()
{
    if (!$this->validate([
        'username' => [
            'label' => 'Username',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
                boleh kosong'
            ]
        ],
        'password' => [
            'label' => 'Password',
            'rules' => 'required',
            'errors' => [
                'required' => '{field} tidak
                boleh kosong'
            ]
        ]
    ])) {
        $validation =
        Services::validation();
        return redirect()-
        >to('/login')->withInput();
    }
}

```

```

$username = $this->request-
>getVar('username');
$password = $this->request-
>getVar('password');
$userData = $this->userModel-
>where(['username' =>
$username, 'password' =>
$password])->first();
if (!is_null($userData)) {
    session()->set(['userdata' =>
    $userData]);
    return
    redirect('admin/dashboard');
} else {
    session()-
    >setFlashdata('status',
    alert('danger', 'Login gagal,
    mohon periksa username atau
    password anda'));
    return redirect()->back()-
    >withInput();
}
}
}

```

app>Controllers>LogoutController.p
hp

```

<?php
namespace App\Controllers;
use
App\Controllers\BaseController
;
class LogoutController extends
BaseController
{
    public function index()
    {
        session()->remove('userdata');
        return redirect()-
        >to('/login');
    }
}

```

ISI
UNIVERSITAS ISLAM NEGERI
SUMATERA UTARA MEDAN