

Rancang Bangun Layanan Informasi Trans Padang Berbasis Web

Hafiz Nuzal Djufri, R. V. Hari Ginardi, dan Dini Adni Navastara

Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember (ITS)

Jl. Arief Rahman Hakim, Surabaya 60111 Indonesia

e-mail: hari@if.its.ac.id

Abstrak—Trans Padang merupakan layanan angkutan massal di kota Padang. Untuk informasi seputar Trans Padang seperti rute, halte dan pencarian rute menjadi masalah utama bagi masyarakat yang ingin menikmati transportasi massal kebanggaan kota Padang. Untuk mengatasi hal itu maka dibutuhkan suatu layanan informasi yang menyediakan berbagai solusi dari masalah tersebut. Adapun layanan informasi yang baik dalam mengakomodasi kepentingan umum ini berbasis *website*. Salah satu kegunaan layanan informasi ini adalah pencarian rute berdasarkan halte kedatangan dan keberangkatan. Selain itu, layanan ini sudah mengakomodir kebutuhan manajemen data yang diperlukan dalam pengembangannya bagi Dinas Perhubungan Komunikasi dan Informatika kota Padang. Adapun layanan ini sudah menampilkan informasi rute dan halte sebanyak lima koridor. Informasi tersebut dapat dilihat detail perkoridornya ataupun langsung secara keseluruhan. Aplikasi ini sudah mengakomodir koridor 1, 2, 3, 5, dan 6, sedangkan kondisi terkini Trans Padang baru mengoperasikan koridor 1. Selain itu, layanan informasi ini sudah dapat mengakomodir kebutuhan bagi pengguna yang ingin mengakses menggunakan telepon selular. Untuk acuan data terkini dan pengembangannya, penelitian ini bereferensi kepada data dari Dinas Perhubungan Komunikasi dan Informatika kota Padang.

Kata Kunci—Halte, Rute, Koridor, Trans Padang

I. PENDAHULUAN

TRANS Padang merupakan layanan angkutan massal Bus Rapid Transit (BRT) di kota Padang yang mulai beroperasi pada Januari 2014. Trans Padang beroperasi pada pukul 06.00 s.d. 20.00 WIB dengan frekuensi kedatangan lima sampai sepuluh menit di setiap haltenya [1]. Trans Padang saat ini baru mengoperasikan koridor pertama yang kedepannya akan dikembangkan menjadi enam koridor [2]. Adapun keenam koridor tersebut beroperasi pada rute Pusat Kota menuju Batas Kota, Bungus, Pusat Pemerintahan, Bandara, Unand dan Semen Padang. Seiring berjalannya waktu, pengguna membutuhkan informasi terkait rute, halte, dan pencarian rute untuk mencapai tempat yang akan ditujunya.

Oleh karena itu, dibutuhkan sistem informasi yang menyediakan berbagai solusi untuk mengatasi masalah tersebut. Adapun sistem informasi yang baik dalam mengakomodasi kepentingan umum ini berbasis *website*. Pada sistem informasi ini ditampilkan peta yang dapat memvisualisasikan halte serta rute yang akan ditempuh pengguna untuk mencapai tujuan. Sistem informasi yang dimaksud diimplementasikan dengan layanan informasi *online* Trans Padang.

Hasil dari penelitian ini ialah layanan informasi *online* yang dapat memvisualisasikan rute sesuai dengan keberangkatan dan tujuan pengguna Trans Padang. Selain itu, terdapat visualisasi halte yang ada di kota Padang serta waktu kedatangan bus pada halte-halte yang dilalui Trans Padang.

II. METODE PENELITIAN

A. Analisis Perangkat Lunak

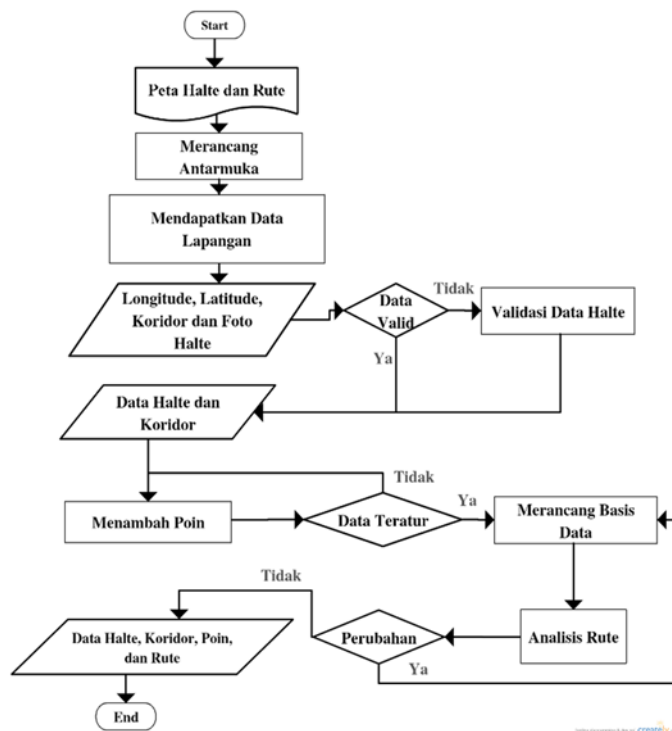
Penelitian ini berupa layanan informasi Trans Padang yang meliputi melihat rute, pencarian rute dan manajemen data. Visualisasi halte dan rute merupakan bagian yang sangat penting dari sistem ini. Selain itu sistem juga menaungi pencarian rute berdasarkan halte kedatangan dan keberangkatan serta manajemen data. Adapun kebutuhan pada sistem ini yaitu:

1. Pengguna dapat melihat seluruh halte dan rute Trans Padang.
2. Pengguna dapat melihat rute dan halte apa saja yang dilalui oleh Trans Padang disetiap koridornya.
3. Pengguna dapat mencari rute berdasarkan halte keberangkatan dan kedatangan.
4. Pengguna dapat mengelola data yang digunakan pada sistem yang meliputi menambah, menyunting dan menghapus data.

B. Perancangan Perangkat Lunak

Perancangan layanan informasi Trans Padang dilakukan melalui berbagai tahapan seperti Gambar 1, yaitu:

1. Mendapatkan Data Peta dan Halte
Data didapatkan dari Dishub Kominfo kota Padang berupa gambar peta halte, rute dan koridor. Data tersebut diperoleh dan diolah untuk kepentingan selanjutnya pada layanan ini.
2. Mendapatkan Data Lapangan
Data lapangan diperoleh dengan tinjauan langsung ke halte-halte untuk mendapatkan koordinat peta secara tepat dan akurat. Selain itu, studi lapangan dibutuhkan untuk mendapatkan *longitude* dan *latitude* dari halte terkait.
3. Validasi Data
Melakukan perbaikan data posisi yang dianggap kurang tepat pada tampilan peta *online*. Pada validasi dilakukan proses penarikan titik menuju posisi idealnya.



Gambar 1. Diagram Alir Perancangan Sistem

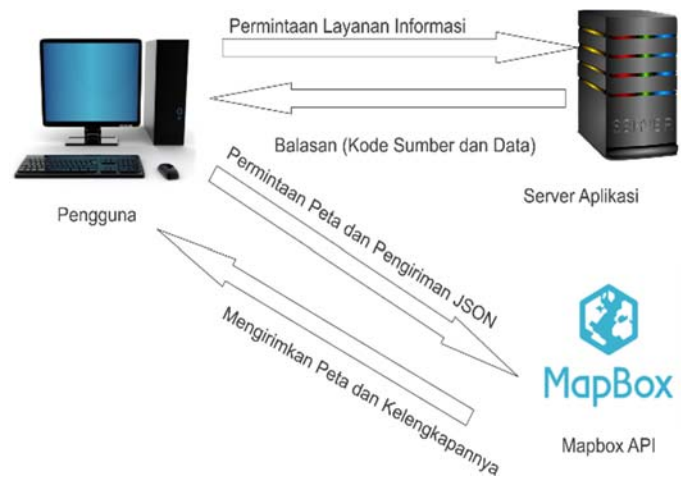
4. Merancang Basis Data

Melakukan proses rancangan untuk penyimpanan data-data yang terkait dalam sistem. Proses perancangan basis data juga meliputi perancangan tipe data atribut yang akan digunakan pada layanan ini. Pembuatan basis data menggunakan SQL. SQL adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional [3].

5. Merancang Antarmuka

Merancang tampilan pada sistem. Perancangan antarmuka meliputi visualisasi peta, halte, rute dan antarmuka manajemen data. Pada proses implementasi antarmuka nantinya menggunakan bahasa pemrograman web diantaranya PHP dan jQuery. PHP dipakai untuk memrogram situs website dinamis. PHP disebut bahasa pemrograman server-side karena PHP diproses pada komputer server [4]. jQuery adalah bahasa yang didesain untuk menyederhanakan naskah HTML dengan menggunakan konsep “find something do something” berbasis klien [5].

Alur layanan informasi diawali dengan pengguna mengakses layanan informasi melalui halaman website. Kerangka kerja yang digunakan pada proses pembuatan website tersebut ialah Laravel. Laravel adalah aplikasi website dengan sintak yang ekspresif dan elegan dengan membawa ideologi “clean code” dan “expressiveness” [6]. Balasan yang dikirimkan oleh server berupa data dalam format JSON yang kemudian dikirimkan kepada Mapbox sebagai penyedia layanan peta online. JSON adalah format pertukaran data berbasis teks yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan oleh client maupun server [7].



Gambar 2. Arsitektur Sistem

Permintaan dari pengguna diolah dan dibalas oleh server berupa kode sumber. Setelah itu, pengguna mengirimkan kode sumber JavaScript untuk melakukan permintaan kepada Mapbox API. Mapbox API merupakan wadah untuk penggunaan kaskas serta layanan yang disediakan Mapbox [8]. Mapbox API memberikan balasan sesuai dengan permintaan kepada server aplikasi yang memberikan balasan tampilan kepada pengguna.

Adapun arsitektur yang digunakan antara pengguna dengan server ialah MVC. MVC terdiri atas Model, View, dan Controller. Model merupakan logika aplikasi yang berhubungan dengan basis data sedangkan View merupakan segala sesuatu yang berhubungan dengan antarmuka aplikasi. Controller merupakan komponen yang menghubungkan Model dan View [9].

Pada proses pencarian rute, dibutuhkan suatu algoritma yang dapat memecahkan permasalahannya. Untuk algoritma pencarian rute seperti pada Gambar 3. Adapun penjelasan tahapan dari algoritma tersebut yaitu:

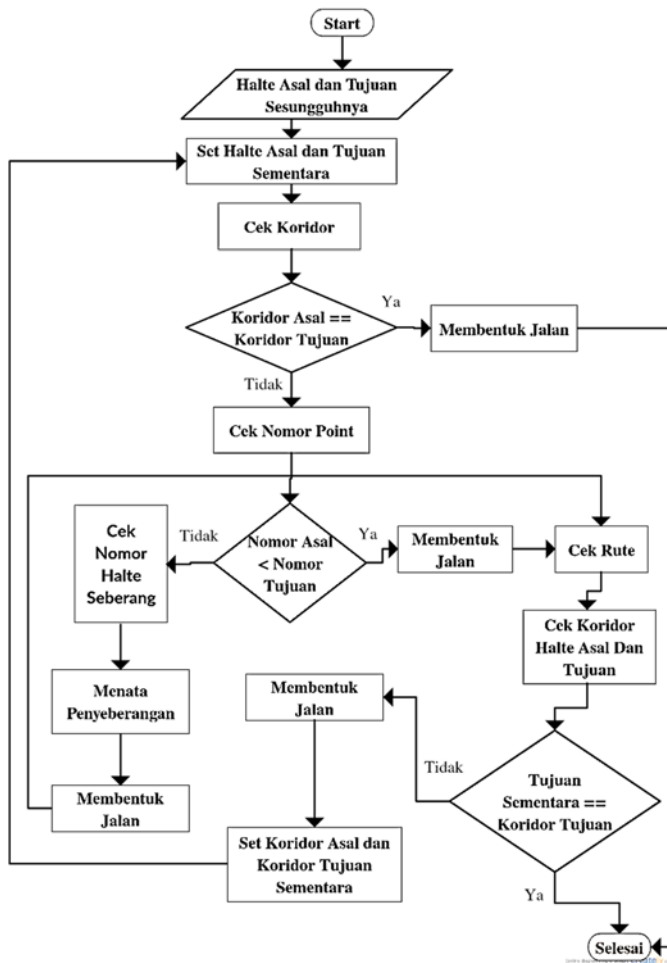
- a. Set Halte Asal dan Tujuan Sementara

Set halte asal dan tujuan sementara merupakan proses menata nilai dari halte asal dan tujuan sementara yang dibutuhkan karena sistem memungkinkan terjadinya perpindahan koridor.
- b. Cek Koridor

Cek koridor merupakan proses pencocokan nilai koridor berdasarkan halte asal dan tujuan. Apabila bernilai sama maka satu lintasan koridor. Jika berbeda, memungkinkan sekoridor yang berbeda lintasan ataupun berbeda koridor.
- c. Membentuk Jalan

Membentuk jalan merupakan proses penyimpanan data-data poin yang akan digunakan untuk menghasilkan jalan yang ditampilkan.
- d. Cek Nomor Point

Cek Nomor Point merupakan proses pengecekan nomor urutan point dari halte yang dimaksud. Proses ini dibutuhkan dalam menentukan apakah dibutuhkan penyeberangan atau tidak. Jika nomor poin kedatangan lebih kecil daripada keberangkatan, maka dibutuhkan proses penyeberangan. Apabila nomor keberangkatan lebih kecil daripada kedatangan, maka rute yang dilalui langsung



Gambar 3. Algoritma Pencarian Rute

(tidak butuh kondisi menyeberang). Adapun nomor poin juga digunakan untuk menata rute yang akan dibentuk.

e. Cek Halte Seberang

Cek halte seberang merupakan proses pengecekan nomor urutan poin pada halte yang berada di seberangnya. Pengecekan halte seberang untuk membentuk jalan yang membutuhkan penyeberangan terlebih dahulu.

f. Menata Penyeberangan

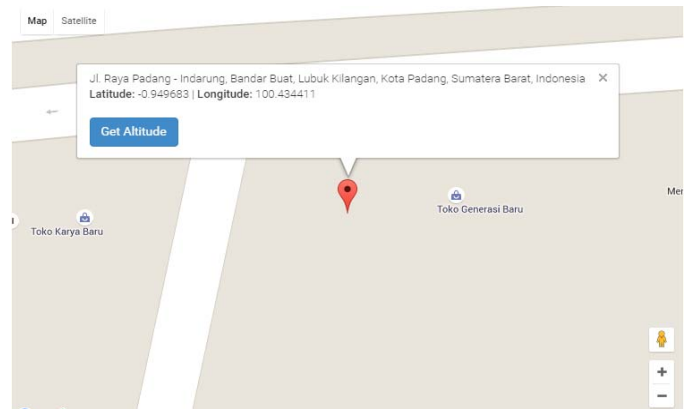
Menata penyeberangan merupakan proses penataan penyeberangan dari halte yang dimaksud menuju posisi halte di seberangnya. Menata penyeberangan dilakukan jika pada pencarian rute membutuhkan proses penyeberangan halte.

g. Set Koridor Asal dan Tujuan Sementara

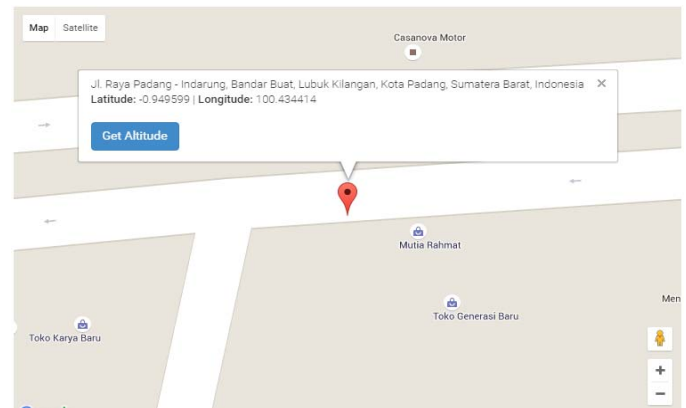
Set koridor asal dan tujuan sementara merupakan proses pencocokan koridor tujuan sementara berfungsi untuk menentukan apakah nilai koridor tujuan sementara telah sama dengan koridor tujuan yang sesungguhnya.

h. Cek Koridor Asal dan Tujuan

Cek koridor asal dan tujuan merupakan proses pencocokan koridor berdasarkan halte asal dan tujuan. Apabila koridor bernilai "id" sama, maka halte tersebut berada selintasan koridor. Jika berbeda, maka kemungkinan halte tersebut berada pada satu koridor yang berbeda lintasan ataupun berbeda koridor.



Gambar 4. Contoh Data Tidak Normal



Gambar 5. Contoh Data Setelah Validasi

III. HASIL UJI COBA DAN EVALUASI

A. Implementasi Umum Sistem

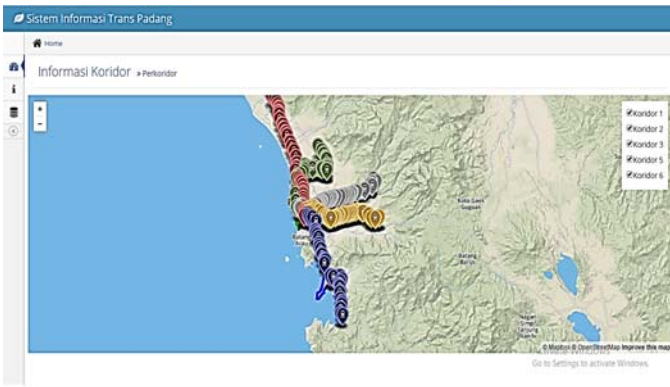
Pada tahap implementasi meliputi pendataan koordinat halte dan validasi data. Data yang diberikan oleh Dishub Kominfo kota Padang ialah data yang disajikan dalam bentuk peta yang didalamnya meliputi rute dan titik halte. Posisi koordinat halte diperoleh dengan pendataan halte. Validasi data pada layanan informasi dilakukan untuk memperbaiki data atau titik halte yang dirasa kurang tepat disaat pencocokan pada peta online. Langkah validasi juga dilakukan karena terjadi perbedaan hasil antara kakas pendukung tinjauan lapangan dengan peta yang digunakan pada layanan informasi Trans Padang.

Validasi data memiliki fungsi utama yakni untuk memperbaiki tampilan posisi halte pada layanan informasi. Adapun data yang dipermasalahkan hanya data yang berposisi sangat jauh dari posisi normal. Untuk data yang berada pada posisi normal tidak diberlakukan proses validasi data tersebut.

Halte SDN 03 dan 05 adalah salah satu contoh data koordinat yang kurang valid dari hasil peninjauan lapangan seperti pada Gambar 8. Data yang kurang valid tersebut dilakukan validasi agar posisi halte mendekati atau berada disisi jalan. Validasi ini dilakukan agar visualisasi halte pada layanan berada pada posisi yang ideal. Untuk visualisasi dan perbedaan hasil pendataan dan setelah validasi dapat dilihat pada Gambar 8 dan Gambar 9. Selain itu, cuplikan hasil data yang telah divalidasi juga dicantumkan pada Tabel 1.

Tabel 1. Cuplikan Data Hasil Pendataan dan Sesudah Validasi

ID	NAMA HALTE	SEBELUM		SESUDAH	
		LATITUDE	LONGITUDE	LATITUDE	LONGITUDE
1	RTH Imam Bonjol	-0.952708	100.363513	-0.952719	100.363636
2	Kantor Pos	-0.949802	100.363544	-0.94983	100.363485
3	SMP 1	-0.947088	100.36275	-0.947088	100.36275
4	IAIN I	-0.94485	100.362239	-0.94485	100.362239
5	Korem I	-0.94052	100.361448	-0.94052	100.361448
6	Kantor Gubernur	-0.937019	100.361151	-0.937019	100.361151
7	BTN I	-0.933072	100.361381	-0.933066	100.361206
8	GOR I	-0.929559	100.361276	-0.929559	100.361276
9	Telkom Padang Baru I	-0.927609	100.36125	-0.927609	100.36125
10	Masjid Raya I	-0.92258	100.36125	-0.92258	100.36125



Gambar 6. Implementasi Antarmuka Informasi Semua Koridor

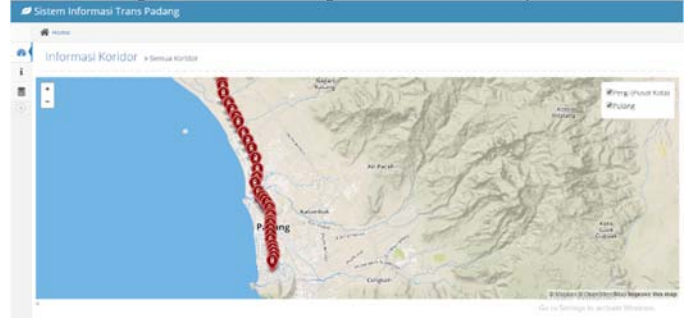
B. Implementasi Visualisasi Peta Rute dan Halte Keseluruhan

Implementasi bagian ini merupakan proses membangun layanan informasi bagian menampilkan dan memproses visualisasi peta rute dan halte keseluruhan. Untuk akses atau penggunaan peta *online* serta marker halte menggunakan bantuan Mapbox. Peta yang ditampilkan diklasifikasikan berdasarkan koridor yang telah didefinisikan oleh *controller*. Data koordinat berserta kelengkapan tersebut disediakan dalam format data JSON yang kemudian diproses dengan bantuan Mapbox untuk ditampilkan visualisasinya. Pada proses pengelompokkan halte, telah didefinisikan pada basis data berupa angka identik yang membedakan antara satu koridor dengan lainnya. Untuk warna perkoridornya telah dibedakan berdasarkan basis data yang diperoleh. Untuk implementasi pewarnaan baik halte ataupun rute menggunakan pustaka yang telah disediakan oleh Mapbox. Adapun perbedaan warna rute dan halte merupakan visualisasi perbedaan disetiap koridornya. Untuk hasil implementasi bagian ini dapat dilihat pada Gambar 6.

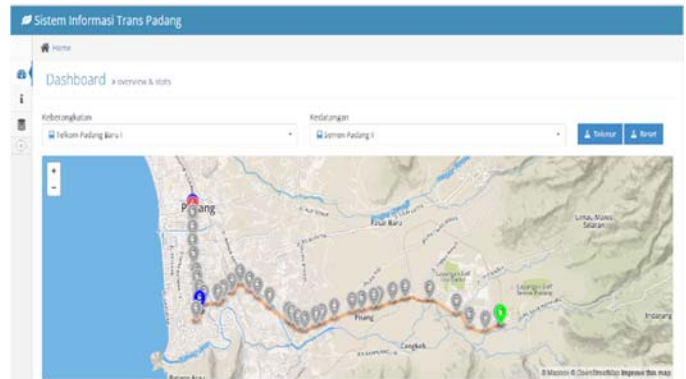
C. Implementasi Visualisasi Peta Rute dan Halte Perkoridor

Implementasi bagian ini merupakan proses membangun layanan informasi bagian menampilkan dan memproses visualisasi peta rute dan halte keseluruhan. Untuk akses atau penggunaan peta *online* serta marker halte menggunakan bantuan Mapbox. Peta yang ditampilkan diklasifikasikan berdasarkan arah rute yang telah didefinisikan oleh *controller*. Data koordinat berserta kelengkapan tersebut telah disediakan dalam format data JSON yang kemudian diproses dengan

bantuan Mapbox untuk ditampilkan visualisasinya. Untuk arah



Gambar 7. Implementasi Antarmuka Informasi Perkoridor



Gambar 8. Implementasi Antarmuka Pencarian Rute

putang merupakan rute yang menuju pusat kota dan pergi arah menjauhi pusat kota. Adapun perbedaan arah juga divisualisasi dengan perbedaan warna antara arah pergi dan pulang. Untuk hasil implementasi bagian ini dapat dilihat pada Gambar 7.

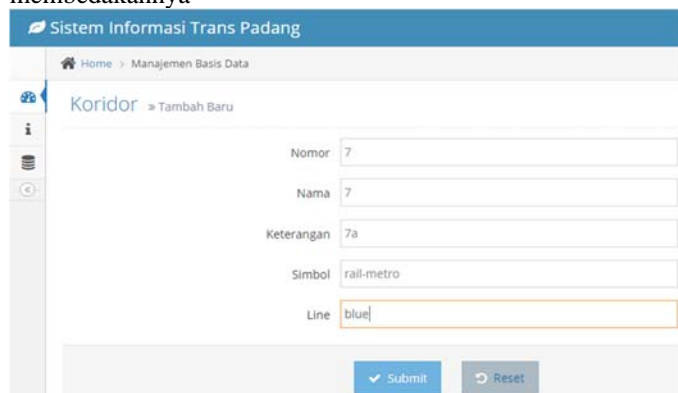
D. Implementasi Visualisasi Pencarian Peta Rute dan Halte

Implementasi bagian ini merupakan proses membangun layanan informasi bagian mencari rute berdasarkan halte keberangkatan dan kedatangan. Pencarian tersebut divisualisasi dengan tampilan berupa peta rute dan halte. Selain itu, layanan informasi juga menampilkan daftar halte yang dilalui beserta keterangan yang hendaknya dilakukan pengguna dalam perjalanan menuju tujuan. Untuk akses atau penggunaan peta *online* serta marker halte menggunakan bantuan Mapbox. Peta yang ditampilkan diklasifikasikan berdasarkan arah rute yang telah didefinisikan oleh *controller*. Setelah itu, data koordinat berserta kelengkapan tersebut disediakan dalam format data JSON yang kemudian diproses dengan bantuan Mapbox untuk ditampilkan visualisasinya. Pewarnaan visualisasi halte dibedakan sesuai tindakan yang hendak dilakukam. Warna biru memiliki arti perpindahan koridor atau penyeberangan halte, sedangkan warna merah dan hijau memiliki arti halte asal dan tujuan. Untuk visualisasi hasil peta rute pencarian dapat dilihat Gambar 8.

E. Implementasi Manajemen Data

Implementasi bagian ini merupakan proses membangun layanan informasi pada bagian menampilkan dan memproses manajemen data sistem. Pada implementasi manajemen data meliputi menambah, menghapus, dan menyunting data yang digunakan sistem. Untuk tampilan konfirmasi hapus data

menggunakan *library sweet-alert*. Tampilan manajemen data halte, poin, koridor dan rute relatif sama, yang membedakannya



Gambar 9. Formulir Tambah Koridor Baru

ialah kolom data tabel beserta isian didalamnya. Proses menyunting dan menghapus data menggunakan parameter atribut "id" dalam pemrosesannya. Untuk penambahan data dilakukan proses pembangkitan nomor "id" secara otomatis.

Untuk posisi dan bentuk tombol aksi seperti menambahkan, menghapus dan menyunting relatif sama. Fitur menyunting dan menghapus data menjadi satu kesatuan pada baris data yang ditampilkan. Untuk proses menghapus data menggunakan fitur *soft delete*. Fitur *soft delete* merupakan pustaka yang didukung kerangka kerja Laravel. Selain itu, bentuk salah satu formulir menambah serta memperbarui data dapat dilihat pada Gambar 9.

F. Hasil Uji Coba

Setelah aplikasi selesai dibangun, maka perlu dilakukan pengujian dengan pengujian fungsionalitas yang menggunakan metode *Black-box*. Metode *Black-box* merupakan metode pengujian yang ditekankan pada pola input dan output yang sesuai dengan skenario yang mengacu kasus penggunaan sistem. Rangkuman hasil uji coba kasus penggunaan dapat dilihat pada Tabel 2.

Berdasarkan hasil pengujian pada Tabel 2 didapatkan bahwa fungsional sistem berhasil dibuat sesuai dengan rancangannya. Sistem memiliki hasil yang baik ditandai dengan dapat memvisualisasikan rute, peta dan halte. Sistem berhasil menampilkan informasi rute dan halte per koridor maupun keseluruhan. Sistem juga dapat membedakan tampilan informasi koridor. Selain itu, sistem dapat menambah, menyunting dan menghapus data.

IV. KESIMPULAN

Layanan informasi Trans Padang yang dibangun telah berhasil merealisasikan kebutuhan sistem. Berdasarkan hasil pengujian, sistem berhasil memvisualisasikan peta halte dan rute yang ada di kota Padang, memvisualisasikan peta halte dan rute yang ada disetiap koridornya dan berhasil menampilkan rute sesuai halte keberangkatan serta kedatangannya baik dalam satu koridor maupun lintas koridor. Selain itu, sistem juga berhasil memvisualisasi dan memanajemen data halte dan rute serta berhasil memvisualisasi tampilan responsif jika diakses menggunakan

telepon selular.

Tabel 2. Hasil Pengujian

No	Kasus Penggunaan	Skenario
1.	Pencarian Rute	Menguji apakah sistem dapat melakukan pencarian dengan benar.
2.	Melihat Informasi Semua Koridor	Menguji apakah informasi koridor berupa rute dan halte dapat ditampilkan.
3.	Melihat informasi Perkoridor	Menguji apakah informasi rute dan halte disetiap koridor dapat ditampilkan.
4.	Menambah Data	<i>Skenario 1</i> Menguji apakah data halte dapat ditambahkan sesuai pengisian formulir. <i>Skenario 2</i> Menguji apakah data poin dapat ditambahkan sesuai pengisian formulir. <i>Skenario 3</i> Menguji apakah data koridor dapat ditambahkan sesuai pengisian formulir. <i>Skenario 4</i> Menguji apakah data rute dapat ditambahkan sesuai pengisian formulir.
5.	Menyunting Data	<i>Skenario 1</i> Menguji apakah data halte dapat disunting <i>Skenario 2</i> Menguji apakah data poin dapat disunting. <i>Skenario 3</i> Menguji apakah data koridor dapat disunting. <i>Skenario 4</i> Menguji apakah data rute dapat disunting.
6.	Menghapus Data	<i>Skenario 1</i> Menguji apakah data halte terkait dapat dihapus. <i>Skenario 2</i> Menguji apakah data poin dapat dihapus. <i>Skenario 3</i> Menguji apakah data koridor dapat dihapus. <i>Skenario 4</i> Menguji apakah data rute dapat dihapus.

DAFTAR PUSTAKA

[1] Dishub Kominfo Padang, "Trans Padang Untuk Kita Semua," Padang, 2016.
 [2] Dishub Kominfo Padang, "Halte Trans Padang," Padang, 2016.
 [3] A. Imam, "SQL Server 2000," in *SQL Server 2000*, Yogyakarta, Graha Ilmu, 2005.
 [4] N. Bunafit, "PHP," in *Membuat Website Sendiri Dengan PHP-MySQL*, Jakarta, Media, 2009.
 [5] jQuery Community Experts, *jQuery Cookbook*, Cambridge: O'Reilly Media, 2009.
 [6] "Tutorial Laravel Bahasa Indonesia," IDLARAVEL, [Online]. Available: <http://id-laravel.com/post/kenapa-memilih-laravel/>. [Accessed 20 April 2016].
 [7] Sriparasa, *JavaScript and JSON Essentials*, Birmingham: Packt Publishing Ltd, 2013.
 [8] "Mapbox API," [Online]. Available: <https://www.mapbox.com/api-documentation/>. [Accessed 20 April 2016].
 [9] C. Pitt, *Pro PHP MVC*, New York: Apress, 2012.