W

;/--n

**TUGAS AKHIR – KI1502**

**RANCANG BANGUN SISTEM INFORMASI TRANS PADANG BERBASIS WEB**

**Hafiz Nuzal Djufri**

**NRP 5112100170**

**Dosen Pembimbing**

**Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.**

**Dini Adni Navastara, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2016**

*****[Halaman ini sengaja dikosongkan]***

**TUGAS AKHIR – KI1502**

**RANCANG BANGUN SISTEM INFORMASI TRANS PADANG BERBASIS WEB**

**Hafiz Nuzal Djufri**

**NRP 5112100170**

**Dosen Pembimbing**

**Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.**

**Dini Adni Navastara, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2016**

****

***[Halaman ini sengaja dikosongkan]***

**UNDERGRADUATE THESES – KI1502**

**WEB BASED INFORMATION SYSTEM DESIGN TRANS PADANG**

**Hafiz Nuzal Djufri**

**NRP 5112100170**

**Dosen Pembimbing**

**Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.**

**Dini Adni Navastara, S.Kom., M.Sc.**

**JURUSAN TEKNIK INFORMATIKA**

**Fakultas Teknologi Informasi**

**Institut Teknologi Sepuluh Nopember**

**Surabaya 2016**

# LEMBAR PENGESAHAN

**Rancang Bangun Sistem Informasi Trans Padang Berbasis Web**

**TUGAS AKHIR**

Diajukan Guna Memenuhi Salah Satu Syarat   
Memperoleh Gelar Sarjana Komputer  
pada   
Bidang Studi Manajemen Informasi  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh :  
**Hafiz Nuzal Djufri**NRP : 5112 100 170

Disetujui oleh Dosen Pembimbing Tugas Akhir :

|  |  |
| --- | --- |
| Dr. Tech. Ir. R. V. Hari Ginardi, M.Sc.  NIP: 196505181992031003 | ................................ (pembimbing 1) |
|  |  |
| Dini Adni Navastara, S.Kom., M.Sc. NIP: 197302191998021101 | ................................ (pembimbing 2) |
|  |  |

**Surabaya  
JUNI 2016**

***[Halaman ini sengaja dikosongkan]***

**RANCANG BANGUN SISTEM INFORMASI TRANS PADANG BERBASIS WEB**

|  |  |  |
| --- | --- | --- |
| **Nama Mahasiswa** | **:** | **Hafiz Nuzal Djufri** |
| **NRP** | **:** | **5112100170** |
| **Jurusan** | **:** | **Teknik Informatika FTIF-ITS** |
| **Dosen Pembimbing 1** | **:** | **Dr. tech. Ir. R.V. Hari Ginardi., M.Sc** |
| **Dosen Pembimbing 2** | **:** | **Dini Adni Navastara, S.Kom, M.Sc.** |

# *Abstrak*

*Trans Padang merupakan layanan angkutan massal bus rapid transit (BRT) di Kota Padang yang mulai beroperasi pada Januari 2014. Trans Padang saat ini baru mengoperasikan koridor pertama yang kedepannya akan dikembangkan sehingga dapat memenuhi kebutuhan angkutan untuk masyarakat Kota Padang. Akan tetapi informasi seputar Trans Padang seperti rute, halte dan waktu kedatangan bus menjadi masalah utama bagi masyarakat yang ingin menikmati transportasi massal kebanggaan Kota Padang.*

*Untuk mengatasi permasalahan tersebut, maka dibutuhkan suatu sistem informasi yang menyediakan berbagai solusi dari masalah tersebut. Adapun sistem informasi yang baik dalam mengakomodasi kepentingan umum ini berbasis web. Pada sistem informasi ini akan ditampilkan peta yang dapat memvisualisasikan halte, terminal serta rute yang akan ditempuh pengguna untuk mencapai tujuan.*

*Hasil dari sistem informasi ini dapat melakukan pencarian rute bedasarkan halte kedatangan dan keberangkatan. Disamping itu juga menampilkan informasi rute dan halte yang pada sistem ini berjumlah 5 koridor. Sistem ini sudah mengakomodir pengembangan karena pada faktanya koridor yang baru berjalan hanya 1, sedangkan sistem ini sudah mengakomodir koridor pengembangan bedasarkan acuan data akurat dari dinas perhubungan kota Padang.*

***Kata kunci: Halte, Rute, Koridor, Trans Padang***

**SOFTWARE DESIGN AND IMPLEMENTATION OF TRANS PADANG WEBSITE BASED**

|  |  |  |
| --- | --- | --- |
| **Student’s Name** | **:** | **Hafiz Nuzal Djufri** |
| **Student’s ID** | **:** | **5112100170** |
| **Department** | **:** | **Teknik Informatika FTIF-ITS** |
| **First Advisor** | **:** | **Dr. tech. Ir. Hari Ginardi., M.Sc** |
| **Second Advisor** | **:** | **Dini Adni Navastara, S.Kom, M.Sc.** |

# *Abstract*

*- -*

# KATA PENGANTAR

Puji syukur kepada Allah SWT atas segala karunia dan rahmat-Nya penulis dapat menyelesaikan tugas akhir yang berjudul:

**“*Rancang Bangun Sistem Informasi Trans Padang Berbasis Web*”**

Melalui lembar ini, penulis hanya ingin menyampaikan ucapan terima kasih dan penghormatan yang sebesar-besarnya kepada:

1. Orang tua saya yakni Bapak Drs. Djufri Mustafa,M.Sc. (Dosen Universitas Andalas) dan Ibu Syarlinawati Akbar, M.M. (Komisaris PT. Semen Padang 2011-2014) yang selalu memberikan dukungan serta merupakan semangat utama penulis dalam mengerjakan tugas akhir.
2. Kakak Penulis yakni Ademulia Djufri,M.B.A., Wiradana Djufri S.T., dr.Teguh Risesa Djufri serta Faiz Iman Djufri,S.T. yang selalu memberikan dukungan penulis dalam mengerjakan tugas akhir.
3. Paman penulis yakni Dr. Patrialis Akbar,S.H.,M.H. (Menteri Hukum dan HAM RI 2009-2011) dan saat ini selaku Hakim Konstitusi Mahkamah Konstitusi.
4. Bapak Dr.Tech. R.V. Hari Ginardi,M.Sc. dan Ibu Dini Adni Navastara,M.Sc. yang bersedia meluangkan waktu untuk memberikan petunjuk dari awal hingga selesai pengerjaan tugas akhir.
5. Rekan organisasi penulis saat ini, Yth. Bapak dan Ibu Majelis Wali Amanat ITS 2016-2021.
6. Civitas Akademika Jurusan Teknik Informatika ITS yang telah banyak memberikan ilmu dan bimbingan bagi penulis.
7. Sahabat penulis, Thiar Hasbiya Ditanaya yang selalu membantu penulis dalam memecahkan permasalahan studi kasus pada tugas akhir.
8. Sahabat Penulis, Asyatul Wellya Rahmi, Hazevi Atila Y.A., Janra Isram serta Marchelli Dwi Muthith yang membantu penulis dalam studi lapangan pencarian data halte.
9. Sahabat penulis, Rahma Fida Fadhillah, Risyanggi Azmi Faizin, Raga Krilido, Achmad Saiful dan Kania Amalia yang bersedia membantu penulis dalam proses pengerjaan tugas akhir.
10. Bapak H. Mahyeldi, S.P. selaku Walikota Padang, Kepala Divisi Angkutan dan Kepala Divisi lalu lintas Dinas Perhubungan Kota Padang yakni Bapak Jovi dan Bapak Indra serta Kepala Divisi Infoduk Dinas Pencatatan Sipil yakni Ibu Dr. Mayulnita, S.H.,M.H.
11. Bapak Elfinur, S.H., M.H. (Kepala Kantor Imigrasi Tanjung Perak 2013-2015) saat ini selaku Direktur Akademi Imigrasi yang selalu membantu segala kesulitan penulis selama di Surabaya.
12. Seluruh teman-teman penulis yang ikut membantu penulis dalam pengerjaan tugas akhir.

Bagaimanapun juga penulis telah berusaha sebaik-baiknya dalam menyusun tugas akhir ini, namun penulis mohon maaf apabila terdapat kekurangan, kesalahan maupun kelalaian yang telah penulis lakukan. Kritik dan saran yang membangun dapat disampaikan sebagai bahan perbaikan selanjutnya.

Surabaya, Juni 2016

Hafiz Nuzal Djufri

# DAFTAR ISI

[LEMBAR PENGESAHAN v](#_Toc453463521)

[*Abstrak* vii](#_Toc453463522)

[*Abstract* ix](#_Toc453463523)

[KATA PENGANTAR xi](#_Toc453463524)

[DAFTAR ISI xiii](#_Toc453463525)

[DAFTAR GAMBAR xvi](#_Toc453463526)

[DAFTAR TABEL xvii](#_Toc453463527)

[DAFTAR KODE SUMBER xix](#_Toc453463528)

[BAB I PENDAHULUAN 1](#_Toc453463529)

[1.1. Latar Belakang 1](#_Toc453463530)

[1.2. Rumusan Permasalahan 1](#_Toc453463531)

[1.3. Batasan Permasalahan 2](#_Toc453463532)

[1.4. Tujuan 2](#_Toc453463533)

[1.5. Metodologi 3](#_Toc453463534)

[1.6. Sistematika Penulisan 4](#_Toc453463535)

[BAB II TINJAUAN PUSTAKA 7](#_Toc453463536)

[2 7](#_Toc453463537)

[2.1. Mapbox API untuk Rute dan Pencarian Jalur Trans Padang 7](#_Toc453463538)

[2.2. Kerangka Kerja Laravel 8](#_Toc453463539)

[2.3. Structure Query Language (SQL) 8](#_Toc453463540)

[2.4. PHP 8](#_Toc453463541)

[2.5. Jquery 8](#_Toc453463542)

[2.6. JSON (JavaScript Object Notation) 9](#_Toc453463543)

[BAB III ANALISIS DAN PERANCANGAN SISTEM 10](#_Toc453463544)

[3.1 Analisis Perangkat Lunak 10](#_Toc453463547)

[3.1.1 Deskripsi Umum Perangkat Lunak 10](#_Toc453463548)

[3.1.2 Spesifikasi Kebutuhan Perangkat Lunak 10](#_Toc453463549)

[3.1.2.1 Kebutuhan Fungsional 11](#_Toc453463550)

[3.1.2.2 Aktor 11](#_Toc453463551)

[3.1.2.3 Kasus Penggunaan 11](#_Toc453463552)

[3.1.2.4 Kasus PenggunaanMencari Rute 12](#_Toc453463553)

[3.1.2.5 Kasus Penggunaan Melihat Seluruh Halte dan Rute Trans Padang 14](#_Toc453463554)

[3.1.2.6 Kasus Penggunaan Melihat halte dan rute per koridor 15](#_Toc453463555)

[3.2 Perancangan 17](#_Toc453463556)

[3.2.1 Perancangan Umum Sistem 17](#_Toc453463557)

[3.2.2 Perancangan Umum Arsitektur Sistem 18](#_Toc453463558)

[3.2.2.1 Perancangan Arsitektur Sistem (Pengguna – Server) 19](#_Toc453463559)

[3.2.2.2 Perancangan Arsitektur Sistem (Pengguna – Mapbox API) 20](#_Toc453463560)

[3.2.3 Perancangan Antarmuka Pengguna 21](#_Toc453463561)

[3.2.3.1 Rancangan Halaman Antarmuka Halaman Utama 21](#_Toc453463562)

[3.2.3.2 Melihat Keseluruhan Rute dan Halte 25](#_Toc453463563)

[3.2.3.3 Rancangan Halaman Antarmuka Melihat Rute dan Halte Perkoridor 28](#_Toc453463564)

[3.3 Perancangan Basis Data 30](#_Toc453463565)

[BAB IV IMPLEMENTASI 31](#_Toc453463566)

[4.1 Lingkungan Implementasi 31](#_Toc453463568)

[4.2 Implementasi Pendataan Kordinat Halte 32](#_Toc453463569)

[4.3 Implementasi Normalisasi Data 35](#_Toc453463570)

[4.4 Implementasi Basis Data 39](#_Toc453463571)

[4.4.1 Implementasi Tabel Koridor 39](#_Toc453463572)

[4.4.2 Implementasi Tabel Halte 41](#_Toc453463573)

[4.4.3 Implementasi Tabel Point 43](#_Toc453463574)

[4.4.4 Implementasi Tabel Rute 45](#_Toc453463575)

[4.5 Implementasi Arsitektur Sistem 47](#_Toc453463576)

[4.5.1 Implementasi Model 47](#_Toc453463577)

[4.5.2 Implementasi View 50](#_Toc453463578)

[4.5.3 Implementasi Controller 61](#_Toc453463579)

[4.6 Implementasi Antarmuka 67](#_Toc453463580)

[4.6.1 Antarmuka Halaman Utama 68](#_Toc453463581)

[4.6.2 Antarmuka Informasi Perkoridor 69](#_Toc453463582)

[4.6.3 Antarmuka Informasi Semua Koridor 70](#_Toc453463583)

[BAB V PENGUJIAN DAN EVALUASI 71](#_Toc453463584)

[5.1 Lingkungan Pengujian 71](#_Toc453463586)

[5.2 Skenario Pengujian 71](#_Toc453463587)

[5.2.1 Kasus Pengujian Melakukan Pencarian Rute 72](#_Toc453463588)

[5.2.2 Kasus Pengujian Informasi Semua Koridor 75](#_Toc453463589)

[5.2.3 Kasus Pengujian Melihat Informasi Perkoridor 78](#_Toc453463590)

[BAB VI KESIMPULAN DAN SARAN 80](#_Toc453463591)

[6.3 Kesimpulan 80](#_Toc453463593)

[6.4 Saran 80](#_Toc453463594)

[DAFTAR PUSTAKA 82](#_Toc453463595)

[LAMPIRAN 84](#_Toc453463596)

[BIODATA PENULIS 85](#_Toc453463597)

# DAFTAR GAMBAR

[Gambar 3.1 Use Case Diagram 11](#_Toc453470912)

[Gambar 3.2 Diagram Aktivitas Mencari Rute 12](#_Toc453470913)

[Gambar 3.3 Diagram Aktivitas Melihat Seluruh Rute dan Halte 14](#_Toc453470914)

[Gambar 3.4 Diagram Aktivitas Melihat Rute dan Halte Perkoridor 15](#_Toc453470915)

[Gambar 3.5 Flow Chart Perancangan Sistem 17](#_Toc453470916)

[Gambar 3.6 Arsitektur Sistem 18](#_Toc453470917)

[Gambar 3.7 Arsitektur MVC 19](#_Toc453470918)

[Gambar 3.8 Arsitektur Client-Server 20](#_Toc453470919)

[Gambar 3.9 Rancangan Antarmuka Halaman Utama 22](#_Toc453470920)

[Gambar 3.10 Rancangan AntarmukaHasil Pencarian Rute 23](#_Toc453470921)

[Gambar 3.11 Rancangan Antarmuka Menampilkan Informasi Seluruh Koridor 25](#_Toc453470922)

[Gambar 3.12 Rancangan Antarmuka Hasil Informasi Seluruh Koridor 26](#_Toc453470923)

[Gambar 3.13 Perancangan Basis Data 29](#_Toc453470924)

.

[Gambar 4.1 Contoh Screenshot Hasil Studi Lapangan 31](#_Toc453470902)

[Gambar 4.2 Contoh Data Tidak Normal 35](#_Toc453470903)

[Gambar 4.3 Contoh Data Setelah Normalisasi 35](#_Toc453470904)

[Gambar 4.4 Cuplikan Basis Data Tabel Koridor 40](#_Toc453470905)

[Gambar 4.5 Cuplikan Basis Data Tabel Halte 42](#_Toc453470906)

[Gambar 4.6 Cuplikan Basis Data Tabel Point 44](#_Toc453470907)

[Gambar 4. 7 Cuplikan Basis Data Tabel Rute 46](#_Toc453470908)

[Gambar 4.8 Impelmentasi Antarmuka Halaman Utama 67](#_Toc453470909)

[Gambar 4. 9 Implementasi Antarmuka Informasi Perkoridor 68](#_Toc453470910)

[Gambar 4.10 Implementasi Antarmuka Informasi Semua Koridor 69](#_Toc453470911)

kl

[Gambar 5.1 Uji Coba Form Pengisian Keberangkatan dan Kedatangan 72](#_Toc453470925)

[Gambar 5. 2 Uji Coba Hasil Pencarian 73](#_Toc453470926)

[Gambar 5.3 Uji Coba Halaman Menampilkan Informasi Seluruh Koridor 75](#_Toc453470927)

[Gambar 5.4 Uji Coba Hasil Menampilkan Informasi Seluruh Koridor 76](#_Toc453470928)

[Gambar 5. 5 Uji Coba Melihat Informasi Perkoridor 78](#_Toc453470929)

DAFTAR TABEL

[Tabel 1 Tabel Kebutuhan 11](#_Toc453463423)

[Tabel 2 Mencari Rute 12](#_Toc453463424)

[Tabel 3 Melihat Seluruh Rute dan Halte 14](#_Toc453463425)

[Tabel 4 Melihat Rute dan Halte Per Koridor 15](#_Toc453463426)

[Tabel 5 Rancangan Antarmuka Halaman Utama 22](#_Toc453463427)

[Tabel 6 Melihat Keseluruhan Halte 25](#_Toc453463428)

[Tabel 7 Penjelasan Antarmuka Edit Image Window 28](#_Toc453463429)

[Tabel 8 Cuplikan Data Halte Koridor 1 33](#_Toc453463430)

[Tabel 9 Cuplikan Data Halte Koridor 2 33](#_Toc453463431)

[Tabel 10 Cuplikan Data Halte Koridor 3 34](#_Toc453463432)

[Tabel 11 Cuplikan Data Halte Koridor 5 34](#_Toc453463433)

[Tabel 12 Cuplikan Data Halte Koridor 6 35](#_Toc453463434)

[Tabel 13 Cuplikan Data Halte Kordior 1 Setelah Normalisasi 37](#_Toc453463435)

[Tabel 14 Cuplikan Data Halte Kordior 2 Setelah Normalisasi 37](#_Toc453463436)

[Tabel 15 Cuplikan Data Halte Kordior 3 Setelah Normalisasi 38](#_Toc453463437)

[Tabel 16 Cuplikan Data Halte Kordior 5 Setelah Normalisasi 38](#_Toc453463438)

[Tabel 17 Cuplikan Data Halte Kordior 6 Setelah Normalisasi 39](#_Toc453463439)

[Tabel 18 Spesifikasi Sistem Pengujian 71](#_Toc453463440)

[Tabel 19 Skenario Pencarian Rute 72](#_Toc453463441)

[Tabel 20 Skenario Melihat Informasi Semua Koridor 75](#_Toc453463442)

[Tabel 21 Uji Coba Melihat Informasi Perkoridor 78](#_Toc453463443)

***[Halaman ini sengaja dikosongkan]***

# DAFTAR KODE SUMBER

[Kode Sumber 1 Membuat Tabel Koridor 40](#_Toc453463395)

[Kode Sumber 2 Menyisipkan Data Pada Tabel Koridor 40](#_Toc453463396)

[Kode Sumber 3 Membuat Tabel Halte 42](#_Toc453463397)

[Kode Sumber 4 Menyisipkan Data Pada Tabel Halte 42](#_Toc453463398)

[Kode Sumber 5 Membuat Tabel Point 44](#_Toc453463399)

[Kode Sumber 6 Menyisipkan Data Pada Tabel Point 45](#_Toc453463400)

[Kode Sumber 7 Membuat Tabel Rute 46](#_Toc453463401)

[Kode Sumber 8 Menyisipkan Data Pada Tabel Rute 46](#_Toc453463402)

[Kode Sumber 9 Model Fungsi Point 48](#_Toc453463403)

[Kode Sumber 10 Model Fungsi Koridor 48](#_Toc453463404)

[Kode Sumber 11 Model Fungsi Point 49](#_Toc453463405)

[Kode Sumber 12 Model Fungsi Rute 50](#_Toc453463406)

[Kode Sumber 13 View Tampilan Dasar 51](#_Toc453463407)

[Kode Sumber 14 View Isian Keberangkatan dan Kedatangan 52](#_Toc453463408)

[Kode Sumber 15 View Pengelompkkan Halte dan Tampilan Peta 53](#_Toc453463409)

[Kode Sumber 16 View Pencarian Js 54](#_Toc453463410)

[Kode Sumber 17 View Checkbox Perkoridor 55](#_Toc453463411)

[Kode Sumber 18 View Klusterisasi Rute Pulang atau Pergi 56](#_Toc453463412)

[Kode Sumber 19 View Rute dan Halte Pergi atau Pulang Json 57](#_Toc453463413)

[Kode Sumber 20 View Checkbox Seluruh Rute dan Halte 58](#_Toc453463414)

[Kode Sumber 21 View Logika Tampil Semua Koridor Js 59](#_Toc453463415)

[Kode Sumber 22 View Tampil Semua Halted an Rute Json 60](#_Toc453463416)

[Kode Sumber 23 Controller Inisialisasi View 61](#_Toc453463417)

[Kode Sumber 24 Controller Tampilan Utama 62](#_Toc453463418)

[Kode Sumber 25 Controller Pencarian Rute 63](#_Toc453463419)

[Kode Sumber 26 Controller Pencarian Halte 64](#_Toc453463420)

[Kode Sumber 27 Controller Halte Perkoridor 65](#_Toc453463421)

[Kode Sumber 28 Controller Rute Perkoridor 66](#_Toc453463422)

***[Halaman ini sengaja dikosongkan]***

# BAB I PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar Tugas Akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan Tugas Akhir, dan sistematika penulisan.

## Latar Belakang

Trans Padang merupakan layanan angkutan massal bus rapid transit (BRT) di Kota Padang yang mulai beroperasi pada Januari 2014. Trans Padang saat ini baru mengoperasikan koridor pertama yang kedepannya akan dikembangkan sehingga dapat memenuhi kebutuhan angkutan untuk masyarakat Kota Padang. Akan tetapi informasi seputar Trans Padang seperti rute, halte dan waktu kedatangan bus menjadi masalah utama bagi masyarakat yang ingin menikmati transportasi massal kebanggaan Kota Padang tersebut.

Untuk mengatasi permasalahan tersebut, maka dibutuhkan suatu sistem informasi yang menyediakan berbagai solusi dari masalah tersebut. Adapun sistem informasi yang baik dalam mengakomodasi kepentingan umum ini berbasis web. Pada sistem informasi ini akan ditampilkan peta yang dapat memvisualisasikan halte, terminal serta rute yang akan ditempuh pengguna untuk mencapai tujuan.

Adapun hasil dari pengerjaan tugas akhir ini ialah sistem informasi 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*.*

## Rumusan Permasalahan

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana membuat sistem informasi yang dapat menampilkan peta jalan yang ada di Kota Padang?
2. Bagaimana membuat sistem informasi yang dapat menyimpan rute Trans Padang?
3. Bagaimana membuat sistem informasi yang dapat memvisualisasikan halte yang ada di Kota Padang?
4. Bagaimana membuat sistem informasi yang dapat menampilkan pencarian halte keberangkatan dan tujuannya?
5. Bagaimana membuat sistem informasi yang dapat menampilkan rute sesuai keberangkatan serta kedatangannya?

## Batasan Permasalahan

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, di antaranya sebagai berikut:

1. Aplikasi ini berbasis web dengan menggunakan bahasa pemrograman PHP serta kerangka kerja Laravell.
2. Data yang digunakan ialah data rute Trans Padang saat ini beserta pengembangannya.
3. Halte serta terminal yang dihimpun ialah data dari Dinas Perhubungan dan Komunikasi Informasi Kota Padang.

## Tujuan

Tujuan dari Tugas Akhir ini adalah sebagai berikut:

1. Memvisualisasi secara digital halte, terminal angkutan Trans Padang.
2. Memvisualisasi secara digital rute dari angkutan Trans Padang.
3. Membuat aplikasi yang dapat mengakomodir kebutuhan pengguna seperti rute yang dilalui bedasarkan halte keberangkatan serta kedatangan, visualisasi halte Trans Padang pada halte yang dilalui.
4. Memudahkan masyarakat mendapatkan informasi seputar Trans Padang.
5. Menyajikan informasi Trans Padang yang online dan dapat diakses dimanapun dan kapanpun.

## Metodologi

Langkah-langkah yang ditempuh dalam pengerjaan Tugas Akhir ini yaitu:

1. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam pembuatan aplikasi yaitu:

1. Mapbox API
2. Pemrograman web
3. Kerangka kerja Laravel.
4. Basis Data
5. Analisis dan Perancangan Sistem

Proses analisis dari perangkat lunak yang akan dibuat ini adalah dengan pemecahan masalah-masalah yang telah dirumuskan dalam bab rumusan masalah. Aktor dari penggunaan sistem informasi ini ialah pengguna Trans Padang. Adapun fitur yang tersedia ialah :

• Melihat Rute dan Halte keseluruhan.

• Mencari rute bedasarkan halte keberangkatan dan kedatangan pengguna.

• Melihat Rute dan Halte Perkoridor

1. Implementasi perangkat lunak

Aplikasi ini akan dibangun menggunakan bahasa pemrograman web dan perangkat kerja Laravell. Aplikasi ini akan dibentuk dengan menggunakan Sublime Text, Mapbox API dan My SQL sebagai Relational Database Management Sytem (RDBMS).

1. Pengujian dan evaluasi

Pengujian dari aplikasi ini dapat dilakukan dengan beberapa cara:

1. Black Box Testing

Black-Box Testing merupakan pengujian yang berfokus pada spesifikasi fungsional dari perangkat lunak, tester dapat mendefinisikan kumpulan kondisi input dan melakukan pengetesan pada spesifikasi fungsional program.

1. Unit Test

Pengujian yang dilakukan hanya satu komponen dari satu sistem. Pengujian ini dilakukan pada fungsi utama produk.

1. Penyusunan buku Tugas Akhir

Pada tahap ini dilakukan pendokumentasian dan pelaporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir.

## Sistematika Penulisan

Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut. Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini.

1. **Pendahuluan**

Bab ini berisi latar belakang masalah, tujuan dan manfaat pembuatan tugas akhir, permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penyusunan tugas akhir.

1. **Dasar Teori**

Bab ini membahas beberapa teori penunjang yang berhubungan dengan pokok pembahasan dan mendasari pembuatan tugas akhir ini.

1. **Analisis dan Perancangan Sistem**

Bab ini membahas mengenai perancangan perangkat lunak. Perancangan perangkat lunak meliputi perancangan data, arsitektur, proses dan perancangan antarmuka pada sistem informasi Trans Padang.

1. **Implementasi**

Bab ini berisi implementasi dari perancangan perangkat lunak sistem informasi Trans Padang dan implementasi fitur-fitur penunjang sistem.

1. **Pengujian dan Evaluasi**

Bab ini membahas pengujian dengan metode pengujian subjektif untuk mengetahui penilaian aspek kegunaan (*usability*)dari perangkat lunak dan pengujian fungsionalitas yang dibuat dengan memperhatikan keluaran yang dihasilkan serta evaluasi terhadap fitur-fitur sistem informasi Trans Padang .

1. **Kesimpulan**

Bab ini berisi kesimpulan dari hasil pengujian yang dilakukan. Bab ini membahas saran-saran untuk pengembangan sistem lebih lanjut.

**Daftar Pustaka**

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

**Lampiran**

Merupakan bab tambahan yang berisi daftar istilah yang penting pada aplikasi ini.

## 

# BAB II TINJAUAN PUSTAKA

Pada bab ini akan dibahas mengenai teori-teori yang menjadi dasar dari pembuatan Tugas Akhir. Teori-teori tersebut meliputi Mapbox API untuk Rute dan Pencarian Jalur Trans Padang, kerangka kerja Laravel, SQL(*Structured Query Language*), dan PHP.



## Mapbox API untuk Rute dan Pencarian Jalur Trans Padang

API (*Application Programming Interface*) atau antarmuka pemrograman aplikasi adalah sekumpulan perintah, fungsi, dan protokol yang dapat digunakan oleh programmer saat membangun perangkat lunak untuk sistem operasi tertentu. API memungkinkan programmer untuk menggunakan fungsi standar untuk berinteraksi dengan sistem operasi. API dapat menjelaskan cara sebuah tugas (task) tertentu dilakukan. Dalam pemrograman prosedural seperti bahasa C, aksi biasanya dilakukan dengan media pemanggilan fungsi. Karena itu, API biasanya menyertakan penjelasan dari fungsi yang disediakannya. Mapbox API merupakan wadah untuk penggunaan kakas serta layanan yang disediakan mapbox [1]. Adapun Mapbox API memiliki sejumlah fungsi yang dapat digunakan agar aplikasi dapat berinteraksi dengan Mapbox khususnya dalam fitur menambahkan rute dan halte.

## Kerangka Kerja Laravel

Laravel adalah aplikasi web dengan sintak yang ekspresif dan elegan dengan membawa ideology “*clean code*” dan “*expressiveness*” [2]. Tugas-tugas umum developer dapat dikurangi pada sebagian besar proyek-proyek web seperti routing, session dan caching. Disamping itu, laravel berusaha menggabungkan pengalaman-pengalaman development dalam bahasa lain, seperti Ruby on Rails, ASP.NET, MVC dan Sinatra. Kerangka kerja ini digunakan untuk membangun aplikasi berbasis website pada Sistem Informasi Trans Padang ini.

## Structure Query Language (SQL)

SQL adalah sebuah bahasa yang digunakan untuk mengakses data dalam basis data relasional [3]. Bahasa ini secara de facto merupakan bahasa standar yang digunakan dalam manajemen basis data relasional. Secara umum, SQL terdiri dari dua bahasa, yaitu Data Definition Language (DDL) dan Data Manipulation Language (DML). Implementasi DDL dan DML berbeda untuk tiap sistem manajemen basis data (SMBD), namun secara umum implementasi tiap bahasa ini memiliki bentuk standar yang ditetapkan ANSI. Artikel ini akan menggunakan bentuk paling umum yang dapat digunakan pada kebanyakan SMBD.

## PHP

PHP adalah bahasa skrip yang dapat ditanamkan atau disisipkan ke dalam HTML. PHP dipakai untuk memrogram situs web dinamis. PHP disebut bahasa pemrograman server side karena PHP diproses pada komputer server [4]. Hal ini berbeda dibandingkan dengan bahasa pemrograman client-side seperti JavaScript yang diproses pada web browser (client).

## Jquery

Jquery adalah bahasa yang didesain untuk menyederhanakan naskah HTML dengan menggunakan konsep “*find something do something*” berbasis klien [5]. Secara spesifik, Jquery mengambil sebagian DOM elemen HTML, dan pengolahan dilakukan menggunakan Jquery. Jquery Kompetible dengan semua versi CSS.

## JSON (JavaScript Object Notation)

JSON adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat oleh komputer.

# BAB III ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini akan membahas analisis perancangan aplikasi desktop untuk mempermudah tahap implementasi aplikasi. Perancangan sistem ini akan direpresentasikan dengan menggunakan diagram *Unified Modelling Language* (selanjutnya disebut UML). Dalam UML nantinya akan terlihat dengan jelas bagiaimana struktur rancangan aplikasi dalam diagram-diagram yang dijelaskan.



## Analisis Perangkat Lunak

Pada subbab ini akan dibahas mengenai analisis perancangan dan arsitektur sistem informasi Trans Padang.

### Deskripsi Umum Perangkat Lunak

Pada Tugas Akhir ini membangun sistem informasi Trans Padang yang meliputi melihat rute dan pencarian rute. Adapun visualisasi halte dan rute merupakan dari sistem ini.

### Spesifikasi Kebutuhan Perangkat Lunak

Sesuai dengan uraian mengenai cakupan sistem informasi yang akan dibangun, dibutuhkan spesifikasi perangkat lunak agar dapat memberikan solusi dari permasalahan yang diberikan dan dapat bekerja dengan baik dalam mengakomodasi kebutuhan. Diharapkan dengan adanya spesifikasi ini dapat menyesuaikan kebutuhan-kebutuhan pengguna.

#### Kebutuhan Fungsional

Tabel 1 Tabel Kebutuhan

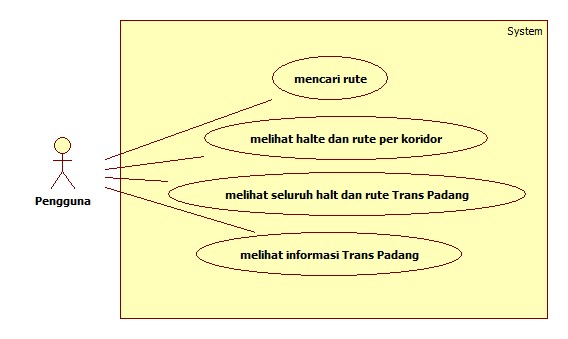
|  |  |  |
| --- | --- | --- |
| Kode Kebutuhan | Kebutuhan Fungsional | Deskripsi |
| UC-001 | Mencari rute | Pengguna dapat mencari rute berdasarkan tempat keberangkatan dan kedatangan yang diinginkan. |
| UC-002 | Melihat seluruh halte dan rute Trans Padang | Pengguna dapat melihat seluruh halte dan rute Trans Padang |
| UC-003 | Melihat halte dan rute per koridor | Pengguna dapat melihat rute dan halte apa saja yang dilalui oleh Bus Trans Padang per koridor |
| UC-004 | Melihat informasi Trans Padang | Pengguna dapat melihat informasi mengenai Trans Padang |

#### Aktor

Aktor adalah pihak-pihak, baik manusia maupun sistem/perangkat lunak lain yang terlibat dan berinteraksi langsung dengan sistem. Dalam sistem untuk tugas akhir ini hanya memiliki 1 aktor yaitu pengguna.

#### Kasus Penggunaan

Bagian ini menjelaskan secara rinci kasus penggunaan yang terdapat pada perangkat lunak. Selain itu, terdapat juga spesifikasi kasus penggunaan, diagram aktivitas dan diagram urutan untuk tiap-tiap kasus penggunaan. Sesuai dengan penjelasan kebutuhan fungsional, maka perangkat lunak memiliki 8 kasus penggunaan yang dapat dilihat pada Gambar3



Gambar 3.1 Use Case Diagram

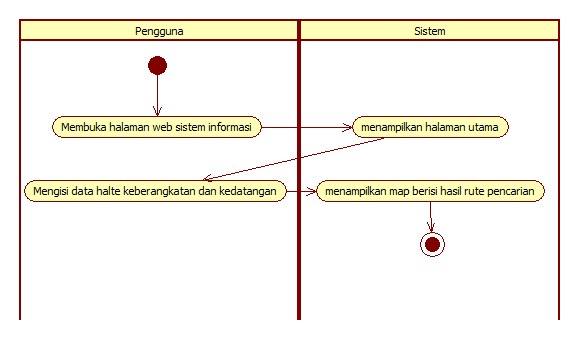
Pada Gambar 3.1

#### Kasus PenggunaanMencari Rute

Pada kasus penggunaan ini, pengguna dapat m. Rincian kasus penggunaan tersebut dapat dilihat pada Tabel 2 dan diagram aktivitas pada

Tabel 2 Mencari Rute

|  |  |
| --- | --- |
| Komponen | Deskripsi |
| Nama | Mencari Rute |
| Kode | UC-001 |
| Deskripsi | Pengguna dapat mencari rute berdasarkan tempat keberangkatan dan kedatangan yang diinginkan. |
| Tipe | Fungsional |
| Pemicu | Pengguna membuka halaman web sistem informasi |
| Aktor | Pengguna |
| Aliran: |  |
| * Kejadian Normal | Pengguna membuka halaman web sistem infromasi  Sistem menampilkan halaman utama  Pengguna mengisi data halte keberangkatan dan halte kedatangan  Sistem menampilkan map berisi rute yang diinginkan |
| * Kejadian Alternatif | Tidak ada |
| Kondisi Akhir | Sistem menampilkan map berisi rute yang diinginakn |
| Kebutuhan Khusus | Tidak ada |



Gambar 3.2 Diagram Aktivitas Mencari Rute

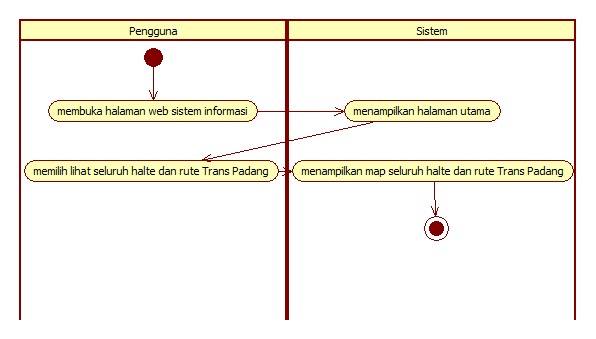
Pada Gambar 3.2 menjelaskan alur aktivitas antara pengguna dan sistem dalam mencari rute yang diinginkan, mulai dari kondisi awal hingga didapatkannya kondisi akhir.

#### Kasus Penggunaan Melihat Seluruh Halte dan Rute Trans Padang

Pada kasus penggunaan ini, pengguna dapat melihat seluruh rute dan halte Trans Padang. Informasi ini menyediakan dan menampilkan semua koridor. Adapun untuk melihat setiap koridornya disediakan pada kasus penggunaan yang berbeda. Rincian kasus penggunaan tersebut dapat dilihat pada Tabel 3 dan diagram aktivitas terdapat pada Gambar 2.

Tabel 3 Melihat Seluruh Rute dan Halte

|  |  |
| --- | --- |
| Komponen | Deskripsi |
| Kode | UC-002 |
| Deskripsi | Pengguna dapat melihat seluruh rute dan halte Trans Padang |
| Tipe | Fungsional |
| Pemicu | Pengguna membuka halaman web sistem informasi |
| Aktor | Pengguna |
| Aliran: |  |
| * Kejadian Normal | Pengguna membuka halaman web sistem infromasi  Sistem menampilkan halaman utama  Pengguna memilih lihat seluruh halte dan rute Trans Padang  Sistem menampilkan map seluruh halte dan rute Trans Padang |
| * Kejadian Alternatif | Tidak ada |
| Kondisi Akhir | Sistem menampilkan map seluruh halte dan rute Trans Padang |



Gambar 3.3 Diagram Aktivitas Melihat Seluruh Rute dan Halte

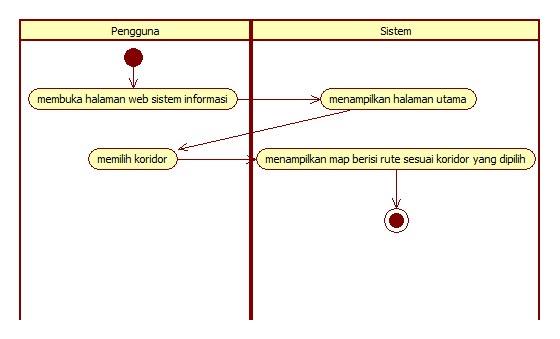
Pada Gambar 3.3 menjelaskan alur aktivitas antara pengguna dan sistem pada aktivitas melihat seluruh rute dan halte dari kondisi awal hingga didapatkannya kondisi akhir.

#### Kasus Penggunaan Melihat halte dan rute per koridor

Pada kasus penggunaan ini, menjelaskan secara lebih detail subbab 3.1.2.6 yakni disajikan data halte dan rute untuk setiap koridornya. Rincian kasus penggunaan tersebut dapat dilihat pada Tabel 4 dan diagram aktivitas terdapat pada Gambar 3.4.

Tabel 4 Melihat Rute dan Halte Per Koridor

|  |  |
| --- | --- |
| Deskripsi | Komponen |
| Nama | Melihat halte dan rute per koridor |
| Kode | UC-003 |
| Deskripsi | Pengguna dapat melihat halte dan rute per koridor. |
| Tipe | Fungsional |
| Pemicu | Pengguna membuka halaman web sistem informasi |
| Aktor | Pengguna |
| Aliran: |  |
| * Kejadian Normal | Pengguna membuka halaman web sistem infromasi  Sistem menampilkan halaman utama  Pengguna memilih koridor yang diinginkan  Sistem menampilkan map berisi rute sesuai koridor yang dipilih |
| * Kejadian Alternatif | Tidak ada |
| Kondisi Akhir | Sistem menampilkan map berisi rute koridor yang dipilih |
| Kebutuhan Khusus | Tidak ada |



Gambar 3.4 Diagram Aktivitas Melihat Rute dan Halte Perkoridor

Pada Gambar 3.4 menjelaskan alur aktivitas antara pengguna dan sistem pada aktivitas melihat rute dan halte perkoridor dari kondisi awal hingga didapatkannya kondisi akhir.

## Perancangan

Pada subbab perancangan akan dijelaskan mengenai arsitektur sistem yang digunakan dan perancangan antarmuka pengguna.

### Perancangan Umum Sistem

Pada perancangan sistem informasi Trans Padang melakukan hal hal sebagai berikut:

* Mendapatkan Data Peta dan Halte

Data peta dan halte diberikan oleh dinas perhubungan kota Padang berupa data peta dan titik secara visual.

* Mendapatkan Data Lapangan

Data lapangan yakni melakukan studi kasus ke halte-halte untuk mendapatkan koordinat peta secara tepat dan akurat

* Normalisasi Data

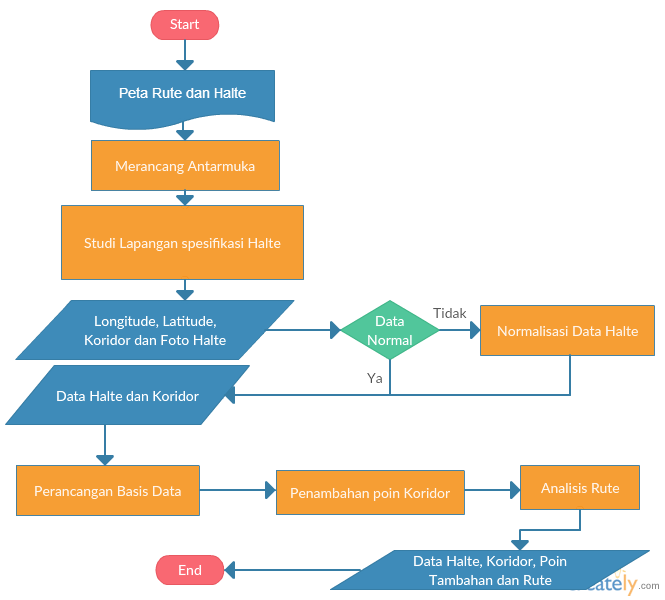
Melakukan perbaikan data yang dirasa kurang berada pada posisi yang normal pada tampilan peta online.

* Merancang Database

Melakukan proses rancangan untuk penyimpanan data-data yang terkait dalam site mini.

* Merancang Antar Muka

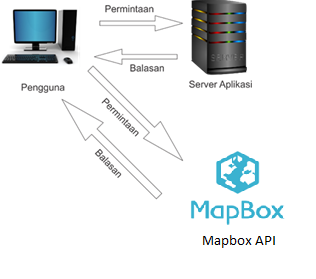
Merancang tampilan dari sistem ini.



Gambar 3.5 Flow Chart Perancangan Sistem

### Perancangan Umum Arsitektur Sistem

Alur aplikasi atau sistem informasi ini ialah pengguna mengakses sistem informasi melalui halaman web. Permintaan dari pengguna diolah dan dibalas oleh server berupa sumber code, kemudian tersebut dijalankan dan sumber kode javascrtipt tersebut melakukan akses / permintaan kepada Mapbox API. Mapbox API memberikan respon sesuai dengan permintaan kepada server aplikasi dan server aplikasi memberikan balasan tampilan kepada pengguna.



Gambar 3.6 Arsitektur Sistem

Gambar 3.6 menjelaskan alur komunikasi pada arsitektur sistem. Mapbox dan Server bertugas memberikan layanan sesuai dengan keinginan pengguna.

#### Perancangan Arsitektur Sistem (Pengguna – Server)

Arsitektur sisten antar pengguna dan sercer yang digunakan ialah Model-View Controller. Model-View-Controller atau MVC adalah salah arsitektur yang digunakan untuk membuat sebuah aplikasi dengan memisahkan data (Model) dari tampilan (View) dan cara bagaimana memproses data untuk ditampilkan Controller). Implementasi kerangka kerja yang ada dalam aplikasi website sebagian besar adalah berbasis arsitektur

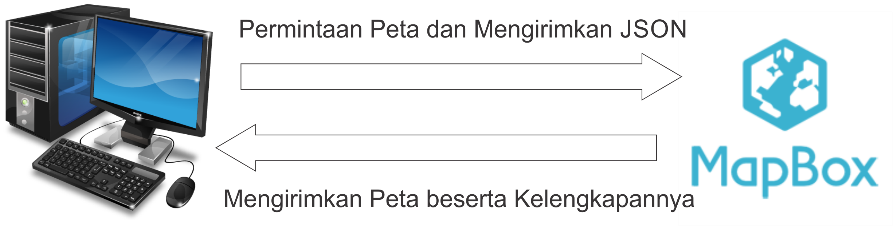


Gambar 3.7 Arsitektur MVC

Pada Gambar 3.7 memperlihatkan bahwa *controller* merupakan muara dari arsitektur ini. Controller menguhubungkan antara model dan view serta kepada pangguna. Controller biasanya berisikan perintah-perintah yang dijalankan sistem.

#### Perancangan Arsitektur Sistem (Pengguna – Mapbox API)

Arstitektur antara pengguna dan Mapbox API menggunakan arsitektur *client-server*. Arsitektur ini merupakan model konektivitas pada jaringan yang membedakan fungsi computer sebagai Client dan Server. Server ini yang bertugas memberikan pelayanan kepada pengguna terhubung dalam sistem.



Gambar 3.8 Arsitektur Client-Server

Gambar 3.8 memperlihatkan bahwa Mapbox API sebagai server. Mapbox API memberikan layanan sesuai dengan permintaan yang diajukan oleh pengguna.

### Perancangan Antarmuka Pengguna

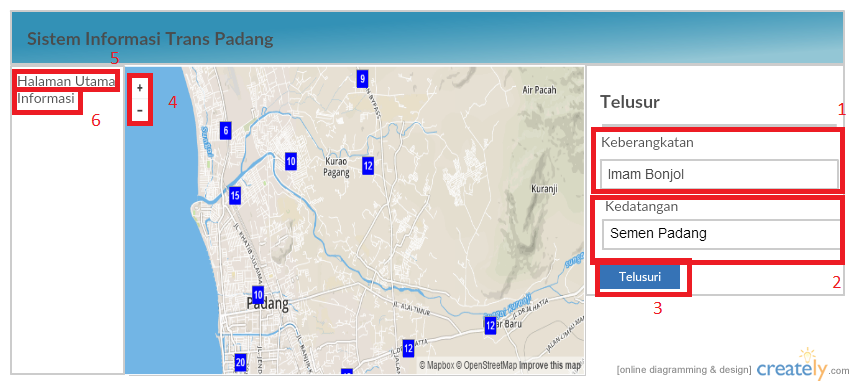
Perancangan antarmuka pengguna merupakan hal yang penting dalam melakukan perancangan aplikasi. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan bagi penggunanya. Sistem memiliki 3 antarmuka pengguna, yaitu halaman Halaman Utama, Informasi Keseluruhan dan Informasi Perkoridor.

#### Rancangan Halaman Antarmuka Halaman Utama

Halaman ini merupakan halaman utama dari aplikasi. Pada halaman ini langsung merujuk kepada halaman pencarian. Pada halaman utama ini disediakan isian untuk halte keberangkan dan halte kedatangan. Selain itu, Pada halaman utama juga menyajikan peta yang menampilkan pengelompokan halte bedasarkan posisi terdekatnya. Rancangan halaman antarmuka dapat dilihat pada Gambar 3.9 dan Tabel 5. Adapun setelah dilakukan pencarian, maka tampilan akan terlihat pada Gambar 3.10

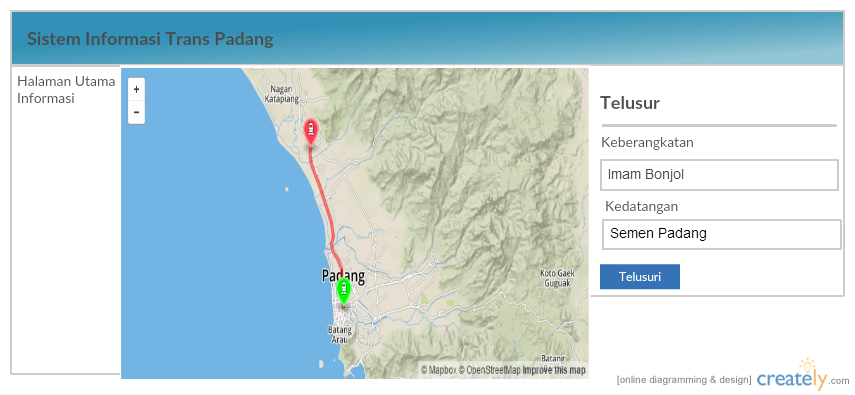
Tabel 5 Rancangan Antarmuka Halaman Utama

|  |  |  |
| --- | --- | --- |
| No. | Nama Atribut Antarmuka | Kegunaan |
| 1 | Keberangkatan | Mencari dan memilih halte keberangkatan / asal yang diinginkan pengguna |
| 2 | Kedatangan | Mencari dan memilih halte kedatangan / tujuan yang diinginkan pengguna |
| 3 | Telusuri | Melakukan pencarian bedasarkan halte kedatangan dan keberangkatan yang telah dimasukkan sebelumnya |
| 4 | +- | Fitur untuk memperbesar atau memperkecil detail peta |
| 5 | Halaman Utama | Untuk menampilkan halaman utama sistem informasi |
| 6 | Informasi | Untuk menampilkan informasi yang akan dilihat. Adapun informasi ini dapat dipilih bedasarkan koridor atau informasi keseluruhan |



Gambar 3.9 Rancangan Antarmuka Halaman Utama

Gambar 3.9 merupakan rancangan antarmuka untuk menampilkan informasi rute dan halte. Bedasarkan gambar, nomor 1, 2 dan 3 berfungsi untuk merupakan form isian halte keberangkatan, kedatangan dan tombol penelusuran atau perncarian. Pada nomor 4 berfungsi untuk memperbesar atau memperkecil detail peta. Untuk nomor 5 dan 6 berfungsi sebagai navigasi menuju halaman utama dan halaman informasi perkoridor atau keseluruhan.



Gambar 3.10 Rancangan AntarmukaHasil Pencarian Rute

Gambar 3.10 merupakan rancangan antarmuka untuk yang menampilkan hasil pencarian rute bedasarkan halte kedatangan dan keberangkatan. Bedasarkan gambar, diberikan variasi visualisasi warna rute dan halte asal dan tujuan.

#### Melihat Keseluruhan Rute dan Halte

Antarmuka ini digunakan untuk melihat visualisasi halte dan rute yang ada di sistem. Adapun pada gambar dapat juga melihat visualisasi halte dan rute sesuai koridor dicentang pada kolom *checkbox* dapat dilihat pada Gambar 3.11 dan 6. Adapun setelah dicentang koridor yang ingin ditampilkan, maka akan ditampilkan seperti pada Gambar 3.12.

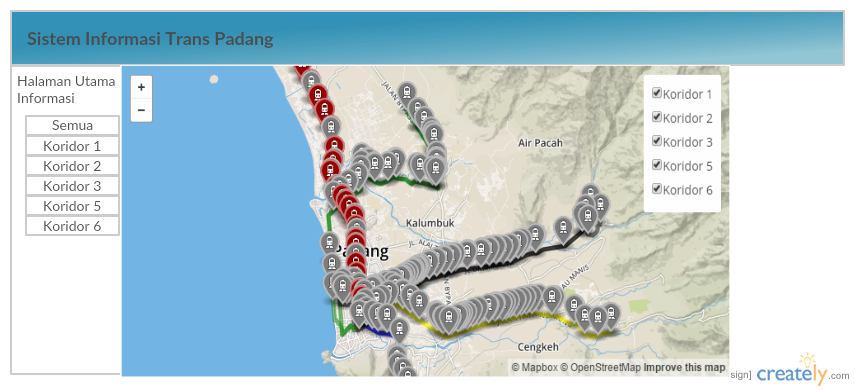
Tabel 6 Melihat Keseluruhan Halte

|  |  |  |
| --- | --- | --- |
| No. | Nama Atribut Antarmuka | Kegunaan |
| 1 | Checkbox | Menampilkan halte dan rute bedasarkan pilihan yang diinginkan pengguna. |
| 2 | +- | Fitur untuk memperbesar atau memperkecil detail peta |
| 3 | Halaman Utama | Untuk menampilkan halaman utama sistem informasi |
| 4 | Informasi | Untuk menampilkan informasi yang akan dilihat. Adapun informasi ini dapat dipilih bedasarkan koridor atau informasi keseluruhan |
|  | Semua | Untuk menampilkan informasi halte dan koridor secara keseluruhan |
|  | Koridor 1 | Untuk menampilkan informasi halte dan koridor 1 |
|  | Koridor 2 | Untuk menampilkan informasi halte dan koridor 2 |
|  | Koridor 3 | Untuk menampilkan informasi halte dan koridor 3 |
|  | Koridor 5 | Untuk menampilkan informasi halte dan koridor 5 |
|  | Koridor 6 | Untuk menampilkan informasi halte dan koridor 6 |



Gambar 3.11 Rancangan Antarmuka Menampilkan Informasi Seluruh Koridor

Gambar 3.11 Menampilkan rancangan antarmuka untuk menampilkan informasi rute dan halte. Bedasarkan gambar, nomor 1 merupakan *checkbox* sebagai isian untuk pengguna dalam menampilkan informasi koridor yang ingin ditampilkan. Nomor 2 berfungsi untuk memperbesar atau memperkecil hasil tampilan peta. Sedangkan nomor 2 dan 3 berfungsi sebagai navigasi menuju halaman utama dan halaman informasi perkoridor atau keseluruhan.



Gambar 3.12 Rancangan Antarmuka Hasil Informasi Seluruh Koridor

Pada Gambar 3.12 Menampilkan rancangan antarmuka yang menampilkan informasi rute dan halte seluruh koridor. Warna rute dan halte akan dibedakan bedasarkan koridor pada saat implementasi.

#### Rancangan Halaman Antarmuka Melihat Rute dan Halte Perkoridor

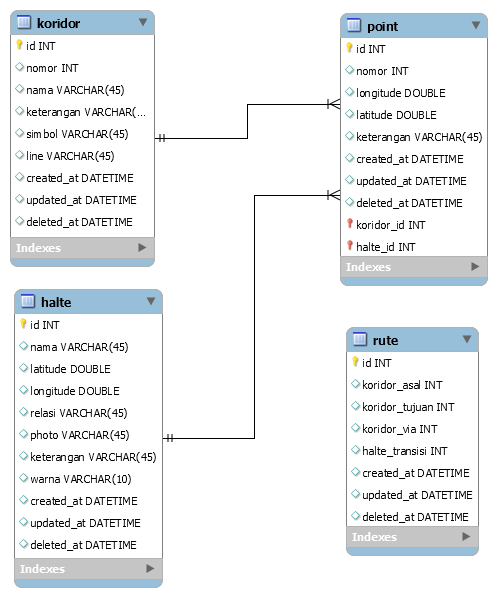
Antarmuka ini digunakan untuk melihat visualisasi halte dan rute yang ada di sistem. Adapun pada gambar dapat juga melihat visualisasi halte dan rute sesuai koridor yang diingiinkan. Rute dan halte yang dilihat dibagi menjadi 2 yakni rute pergi dan rute pulang. Halaman antarmuka dapat dilihat Gambar dan Tabel 7

Tabel 7 Penjelasan Antarmuka Edit Image Window

|  |  |  |
| --- | --- | --- |
| No | Nama Atribut Antarmuka | Kegunaan |
| 1 | Checkbox | Menampilkan halte dan rute bedasarkan pilihan yang diinginkan pengguna. |
| 2 | +- | Fitur untuk memperbesar atau memperkecil detail peta |
| 3 | Halaman Utama | Untuk menampilkan halaman utama sistem informasi |
| 4 | Informasi | Untuk menampilkan informasi yang akan dilihat. Adapun informasi ini dapat dipilih bedasarkan koridor atau informasi keseluruhan |
|  | Semua | Untuk menampilkan informasi halte dan koridor secara keseluruhan |
|  | Koridor 1 | Untuk menampilkan informasi halte dan koridor 1 |
|  | Koridor 2 | Untuk menampilkan informasi halte dan koridor 2 |
|  | Koridor 3 | Untuk menampilkan informasi halte dan koridor 3 |
|  | Koridor 5 | Untuk menampilkan informasi halte dan koridor 5 |
|  | Koridor 6 | Untuk menampilkan informasi halte dan koridor 6 |

# 3.3 Perancangan Basis Data

Pada perancangan basis data membutuhkan analisis peta dan rute Trans Padang. Adapun peta Trans Padang yang berjumlah 5 koridor dapat dilihat pada Lampiran….. Setelah melihat dan menganalisis Peta Trans Padang, maka penulis merancang basis data yang dapat dilihat pada Gambar



Gambar 3.13 Perancangan Basis Data

Gambar 3.13 menjelaskan bahwa tabel halte berhubungan dengan tabel point dengan relasi *one to many*. Selain itu, tabel koridor memiliki relasi *one to many* dengan tabel point.

# BAB IV IMPLEMENTASI

Pada bab ini akan dibahas mengenai implementasi sistem sesuai dengan analisis dan perancangan sistem secara umum pada aplikasi editor album foto digital yang dijabarkan pada bab sebelumnya.

Implementasi yang dijelaskan meliputi lingkungan pembangunan perangkat lunak, kode sumber utama dan implementasi antarmuka perangkat lunak. Bahasa pemrograman yang digunakan adalah Bahasa pemrograman Web, *framework* untuk laravel dan menggunakan My SQL untuk basis data



## Lingkungan Implementasi

Lingkungan implementasi yang akan digunakan untuk melakukan implementasi pada tugas akhir ini dilakukan pada lingkungan dengan kakas sebagai berikut.

1. Sistem operasi Windows 10 Enterprise 64 bit.
2. My SQL Workbench 6.3 CE.
3. Heidi SQl 9.3.
4. XAMPP 3.2.2 Apache Web server.
5. StarUML 2.7 digunakan untuk membuat diagram kasus penggunaan dan diagram aktivitas.

Adapun untuk implementasi, maka dibutuhkan perkakas bantuan untuk menunjang pengerjaan sebagai berikut:

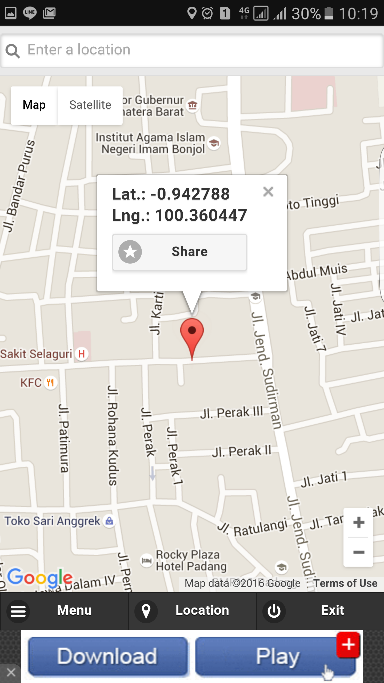
Telepon genggam dengan sistem operasi Android dalam hal ini menggunakan Samsung Galaxy S7 Edge.

GPS Coord. Finder untuk mendapatkan longitude dan latitude halte pada saat studi lapangan.

Normalisasi data, penambahan poin menggunakan fitur pada aplikasi website www.gps-coordinate.net.

## Implementasi Pendataan Kordinat Halte

Sistem yang dibuat memiliki banyak data halte serta rute yang akan disimpan didalamnya. Data yang diberikan oleh Dinas Perhubungan kota Padang ialah data yang disajikan dalam bentuk peta yang didalamnya meliputi rute dan titik halte. Maka data halte yang akan diperolah dengan pendataan halte. Pada Gambar.. dapat dilihat contoh perolehan data koordinat yang menggunakan aplikasi GPS Coord Finder



Gambar 4.1 Contoh Screenshot Hasil Studi Lapangan

Gambar 4.1 Merupakan contoh koordinat dari salah satu halte yang ada di Trans Padang. Koordinat tersebut didapat dengan studi lapangan menuju lokasi halte dan hasil tersebut disimpan untuk dilakukan pencatatan kordinat pada tahap selanjutnya. Cuplikan dari pencatatan dapat dilihat pada Tabel 8, 9, 10, 11 dan 12.

Tabel 8 Cuplikan Data Halte Koridor 1

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

Tabel 9 Cuplikan Data Halte Koridor 2

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | LONGITUDE |
| 84 | Pasar Gaung | -0.991894 | 100.383657 |
| 85 | Toko Gani | -0.995099 | 100.385328 |
| 86 | Masjid Al Qohar | -1.001715 | 100.387856 |
| 87 | Pabrik Karet Sel Beremas | -1.007585 | 100.390633 |
| 88 | Pantai Nirwana | -1.008343 | 100.392117 |
| 89 | PPS Bungus | -1.02745 | 100.399217 |
| 90 | Pelabuhan Penyebrangan | -1.028218 | 100.403146 |
| 91 | Kampus II MAN 1 Padang | -1.027809 | 100.412248 |
| 92 | SPBU Bungus | -1.031516 | 100.415207 |
| 93 | SMP 19 | -1.034248 | 100.412842 |

Tabel 10 Cuplikan Data Halte Koridor 3

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | LONGITUDE |
| 139 | Pusat Pemetintahan Aia Pacah | -0.877145 | 100.386874 |
| 140 | FKG Baiturrahmah | -0.880518 | 100.389661 |
| 141 | Toko Bangunan Anifa | -0.884852 | 100.392238 |
| 142 | Universitas Terbuka | -0.88702 | 100.393668 |
| 143 | Toyota By Pass | -0.894116 | 100.394905 |
| 144 | RM Arafah | -0.896173 | 100.39484 |
| 145 | Hendra Auto Service | -0.899682 | 100.395188 |
| 146 | PNM Ulam Syariah | -0.900207 | 100.394571 |
| 147 | Masjid Anwar | -0.899612 | 100.393013 |
| 148 | Pagang | -0.898713 | 100.387881 |
| 149 | Kantor Camat Nanggalo | -0.896519 | 100.377463 |

Tabel 11 Cuplikan Data Halte Koridor 5

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | LONGITUDE |
| 215 | Semen Padang | -0.956325 | 100.46702 |
| 216 | Semen Padang II | -0.956401 | 100.467459 |
| 217 | Audio Mobil | -0.957958 | 100.462299 |
| 218 | Warung Dhesie | -0.957095 | 100.456429 |
| 219 | Muncak Motor | -0.95346 | 100.452309 |
| 220 | TK Adzkia V | -0.950022 | 100.444322 |
| 221 | Seberang Toko H Bahar | -0.949046 | 100.437419 |
| 222 | SDN 03 dan 05 | -0.949683 | 100.434411 |
| 223 | Seberang BNI Syariah Bandar Buat | -0.949786 | 100.432286 |
| 224 | CV Tisati Persada | -0.95147 | 100.452902 |
| 225 | PT. Sumatex Subur | -0.952185 | 100.422308 |

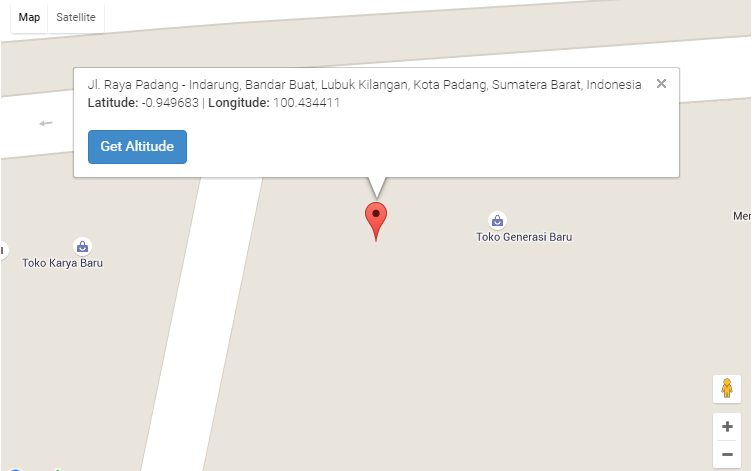
Tabel 12 Cuplikan Data Halte Koridor 6

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | LONGITUDE |
| 269 | Damar Plaza | -0.944274 | 100.35455 |
| 270 | Panin | -0.945818 | 100.355621 |
| 271 | Bumi Putera | -0.945817 | 100.357408 |
| 272 | McD | -0.943091 | 100.355963 |
| 273 | Taman Sari | -0.942797 | 100.358699 |
| 274 | Rumah Walikota | -0.942788 | 100.360477 |
| 275 | SMA Bukit Barisan | -0.944948 | 100.36232 |
| 276 | SMP 1 | -0.946852 | 100.362645 |
| 277 | Masjid Mubarak | -0.947476 | 100.364709 |
| 278 | BCA | -0.947587 | 100.364612 |
| 279 | PT. Pembina Citra Utama | -0.945763 | 100.368565 |

## Implementasi Normalisasi Data

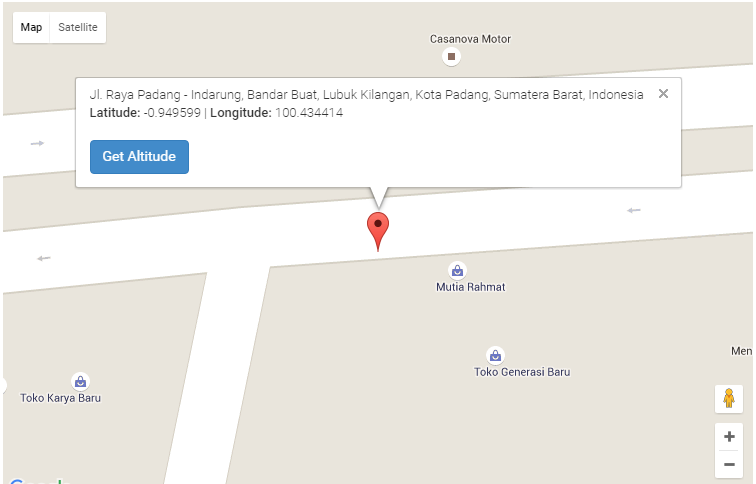
Normalisasi data pada sistem informasi dilakukan untuk memperbaiki data atau titik halte yang dirasa kurang tepat disaat pencocokan pada peta *online*. Langkah normalisasi digunakan karena pengambilan koordinat yang kurang cocok antara kakas pendukung studi lapangan dengan peta yang akan digunakan pada sistem informasi Trans Padang. Selain itu, Normalisasi memliki fungsi utama yakni untuk memperbaiki tampilan posisi halte pada sistem informasi. Adapun data yang dipermasalahka hanya data yang berposisi sangat jauh dari posisi normal.

Halte SDN 03 dan 05 adalah salah satu contoh kekurang cocokan koordinat hasil studi lapangan yang dapat dilihat pada Gambar 4.2.



Gambar 4.2 Contoh Data Tidak Normal

Pada Gambar 4.2 terlihat bahwa posisi koordinat sangat jauh dari jalan raya yang semestinya sebuah halte berposisis tepat disisi jalan raya. Oleh karena itu, maka dilakukan normalisasi data dengan menarik posisi kordinat mendekati sisi jalan dengan hasil seperti pada Gambar 4.3



Gambar 4.3 Contoh Data Setelah Normalisasi

Adapun untuk cuplikan hasil normalisasi data dapat dilihat pada Tabel 13, 14, 15 dan 17.

Tabel 13 Cuplikan Data Halte Kordior 1 Setelah Normalisasi

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

Tabel 14 Cuplikan Data Halte Kordior 2 Setelah Normalisasi

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | | LONGITUDE | |
| 84 | Pasar Gaung | | -0.991894 | | 100.383657 | |
| 85 | Toko Gani | | -0.995099 | | 100.385328 | |
| 86 | Masjid Al Qohar | | -1.001715 | | 100.387856 | |
| 87 | Pabrik Karet Sel Beremas | | -1.007213 | | 100.390632 | |
| 88 | Pantai Nirwana | | -1.008343 | | 100.392117 | |
| 89 | PPS Bungus | | -1.02745 | | 100.399217 | |
| 90 | Pelabuhan Penyebrangan | | -1.028218 | | 100.403146 | |
| 91 | Kampus II MAN 1 Padang | | -1.027809 | | 100.412248 | |
| 92 | SPBU Bungus | | -1.031516 | | 100.415207 | |
| 93 | SMP 19 | | -1.034248 | | 100.412842 | |

Tabel 15 Cuplikan Data Halte Kordior 3 Setelah Normalisasi

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | LONGITUDE |
| 139 | Pusat Pemerintahan Aia Pacah | -0.877145 | 100.386874 |
| 140 | FKG Baiturrahmah | -0.880588 | 100.389529 |
| 141 | Toko Bangunan Anifa | -0.884852 | 100.392238 |
| 142 | Universitas Terbuka | -0.887055 | 100.393584 |
| 143 | Toyota By Pass | -0.894122 | 100.394862 |
| 144 | RM Arafah | -0.896167 | 100.394953 |
| 145 | Hendra Auto Service | -0.899682 | 100.395231 |
| 146 | PNM Ulam Syariah | -0.900276 | 100.394545 |
| 147 | Masjid Anwar | -0.899717 | 100.392997 |
| 148 | Pagang | -0.898677 | 100.387892 |

Tabel 16 Cuplikan Data Halte Kordior 5 Setelah Normalisasi

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | LONGITUDE |
| 215 | Semen Padang II | -0.956401 | 100.467459 |
| 216 | Audio Mobil | -0.957958 | 100.462299 |
| 217 | Warung Dhesie | -0.95707 | 100.456443 |
| 218 | Muncak Motor | -0.95346 | 100.452309 |
| 219 | TK Adzkia V | -0.950022 | 100.444322 |
| 220 | Seberang Toko H Bahar | -0.949158 | 100.437382 |
| 221 | SDN 03 dan 05 | -0.949599 | 100.434414 |
| 222 | Seberang BNI Syariah Bandar Buat | -0.94986 | 100.432314 |
| 223 | CV Tisati Persada | -0.9509 | 100.428055 |
| 224 | PT. Sumatex Subur | -0.952185 | 100.422308 |

Tabel 17 Cuplikan Data Halte Kordior 6 Setelah Normalisasi

|  |  |  |  |
| --- | --- | --- | --- |
| ID | NAMA HALTE | LATITUDE | LONGITUDE |
| 269 | BCA | -0.947529 | 100.364584 |
| 270 | PT. Pembina Citra Utama | -0.945812 | 100.368584 |
| 271 | Stasiun Kereta | -0.943926 | 100.374307 |
| 272 | Bank Nagari Andalas | -0.943125 | 100.376815 |
| 273 | Dalas Motor | -0.941588 | 100.380761 |
| 274 | Depot Air Skalau Krul | -0.940183 | 100.383199 |
| 275 | Swalayan Aqlan | -0.938905 | 100.385508 |
| 276 | SMP 31 | -0.938337 | 100.386562 |
| 277 | Nusantara Poultry Shop | -0.938001 | 100.387168 |
| 278 | Masjid Kebenaran Anduring | -0.93594 | 100.391681 |

## Implementasi Basis Data

Implementasi pembuatan basis data menggunakan bantuan SQL. Pada bagian ini, terdapat sub bagian yang menjelaskan setiap implementasi setiap tabel beserta hasilnya. Impelemntasi basis data ini diperuntukkan untuk menyimpan dan mengelola data yang dibutuhkan dalam pembentukan sistem informasi ini

### Implementasi Tabel Koridor

Implementasi pembuatan table koridor diperuntukkan menyimpan data koridor pada sistem informasi. Tabel koridor terdiri dari beberapa kolom yakni id, nomor, nama, keterangan, simbol, line, created\_at, updated\_at dan deleted\_at. Untuk implementasi serta hasilnya dapat dilihat pada Kode Sumber 1 dan 2 serta Gambar 4.4.

CREATE TABLE `trans\_padang`.`koridor` (

`id` INT NOT NULL,

`nomor` INT NULL,

`nama` VARCHAR(45) NULL,

`keterangan` VARCHAR(45) NULL,

`simbol` VARCHAR(45) NULL,

`line` VARCHAR(45) NULL,

`created\_at` DATETIME NULL,

`updated\_at` DATETIME NULL,

`deleted\_at` DATETIME NULL,

PRIMARY KEY (`id`))

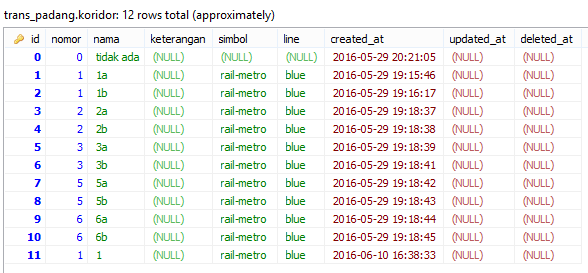
Kode Sumber 1 Membuat Tabel Koridor

Pada Kode Sumber 1 dapat dilihat bahwa untuk id dan nomor memiliki tipe data integer. Pada keterangan, simbol dan line bertipe data varchar serta crated \_at, updated\_at dan deleted\_at bertipe data datetime. Untuk *primary key* table ini ialah id.

Kode Sumber 2 Menyisipkan Data Pada Tabel Koridor

**INSERT** **INTO** `trans\_padang`.`koridor` (`id`, `nomor`, `nama`, `simbol`, `line`, `created\_at`) **VALUES** (1, 1, '1a', 'rail-metro', 'blue', '2016-06-10 16:38:33');

Pada Kode Sumber 2 merupakan salah satu contoh data yang disimpan dalam tabel koridor. Kolom updated\_at dan kolom deleted\_at belum terisi atau secara langsung terdefinisi dengan nilai null.



Gambar 4.4 Cuplikan Basis Data Tabel Koridor

Pada Gambar 4.4 memperlihatkan hasil table setelah dimasukkan sejumlah data yang diperlukan pada tabel koridor. Tabel koridor berfungsi untuk mengelompokkan titik-titik halte bedasarkan koridornya.

### Implementasi Tabel Halte

Implementasi pembuatan tabel halte diperuntukkan menyimpan data halte pada sistem informasi. Tabel halte terdiri dari beberapa kolom yakni id, nama, latitude, longitude, relasi, photo, keterangan, warna, created\_at, updated\_at dan deleted\_at. Untuk implementasi serta hasilnya dapat dilihat pada Kode Sumber… dan Gambar….

CREATE TABLE `trans\_padang`.`halte` (

`id` INT NOT NULL,

`nama` VARCHAR(45) NULL,

`latitude` DOUBLE NULL,

`longitude` DOUBLE NULL,

`relasi` VARCHAR(45) NULL,

`photo` VARCHAR(45) NULL,

`keterangan` VARCHAR(45) NULL,

`warna` VARCHAR(10) NULL,

`created\_at` DATETIME NULL,

`updated\_at` DATETIME NULL,

`deleted\_at` DATETIME NULL,

PRIMARY KEY (`id`))

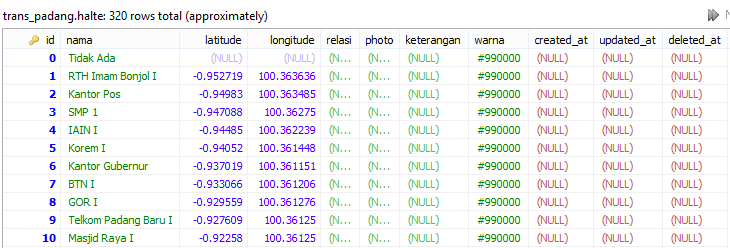
Kode Sumber 3 Membuat Tabel Halte

Pada Kode Sumber 3 dapat dilihat bahwa untuk id bertipe data integer serta latitude dan longitude bertipe data double. Pada warna, relasi, photo, keterangan bertipe data varchar serta crated \_at, updated\_at dan deleted\_at bertipe data datetime. Untuk *primary key* table ini ialah id.

Kode Sumber 4 Menyisipkan Data Pada Tabel Halte

**INSERT** **INTO** `trans\_padang`.`halte` (`id`, `nama`, `latitude`, `longitude`) **VALUES** (319, 'RS Siti Rahmah', -0.871082, 100.382369);

Pada Kode Sumber 4 merupakan salah satu contoh data yang disimpan dalam tabel halte. Kolom created\_at, updated\_at dan kolom deleted\_at belum terisi atau secara langsung terdefinisi dengan nilai null.



Gambar 4.5 Cuplikan Basis Data Tabel Halte

Pada Gambar… memperlihatkan hasil tabel setelah dimasukkan sejumlah data yang diperlukan pada tabel halte. Tabel halte berfungsi untuk menyimpan titik-titik halte dan kelengkapan yang dibutuhkan pada saat menampilkan hasil pada peta.

### Implementasi Tabel Point

Implementasi pembuatan tabel point diperuntukkan menyimpan data poin pada sistem informasi. Tabel poin membantu untuk menyempurnakan tampilan pada rute yang akan dihasilkan peta. Tabel poin terdiri dari beberapa kolom yakni id, nomor, latitude, longitude, keterangan, created\_at, updated\_at, deleted\_at, koridor\_id dan halte\_id. Untuk implementasi serta hasilnya dapat dilihat pada Kode Sumber… dan Gambar….

CREATE TABLE `trans\_padang`.`point` (

`id` INT NOT NULL,

`nomor` INT NULL,

`longitude` DOUBLE NULL,

`latitude` DOUBLE NULL,

`keterangan` VARCHAR(45) NULL,

`created\_at` DATETIME NULL,

`updated\_at` DATETIME NULL,

`deleted\_at` DATETIME NULL,

`koridor\_id` INT NOT NULL,

`halte\_id` INT NOT NULL,

PRIMARY KEY (`id`, `koridor\_id`, `halte\_id`),

INDEX `fk\_point\_rute1\_idx` (`koridor\_id` ASC),

INDEX `fk\_point\_halte1\_idx` (`halte\_id` ASC),

CONSTRAINT `fk\_point\_rute1`

FOREIGN KEY (`koridor\_id`)

REFERENCES `trans\_padang`.`koridor` (`id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_point\_halte1`

FOREIGN KEY (`halte\_id`)

REFERENCES `trans\_padang`.`halte` (`id`)

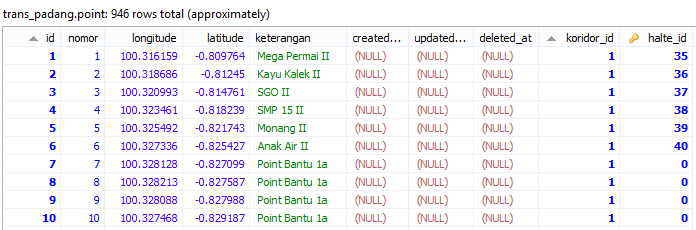
Kode Sumber 5 Membuat Tabel Point

Pada Kode Sumber 5 terlihat bahwa untuk id dan nomor bertipe data integer serta latitude dan longitude bertipe data double. Pada keterangan bertipe data varchar serta created \_at, updated\_at dan deleted\_at bertipe data datetime. Pada tabel ini terdapat *foreign key* yakni halte\_id dan koridor\_id yang bereferensi kepada tabelnya. Untuk *primary key* tabel ini ialah id.

Kode Sumber 6 Menyisipkan Data Pada Tabel Point

**INSERT** **INTO** `trans\_padang`.`point` (`id`, `nomor`, `longitude`, `latitude`, `keterangan`, `created\_at`, `koridor\_id`, `halte\_id`) **VALUES** (977, 57, 100.364475, -0.954066, 'Poin Bantu 1a', '2016-06-10 17:11:57', 1, 0);

Pada Kode Sumber 6 merupakan salah satu contoh data poin yang disimpan dalam tabel. Kolom created\_at, updated\_at dan kolom deleted\_at belum terisi atau secara langsung terdefinisi dengan nilai null.



Gambar 4.6 Cuplikan Basis Data Tabel Point

Pada Gambar 4.6 memperlihatkan hasil tabel setelah dimasukkan sejumlah data yang diperlukan pada tabel point. Tabel poin berfungsi untuk menyimpan titik-titik poin dan kelengkapan untuk penyempurnaan tampilan rute.

### Implementasi Tabel Rute

Implementasi pembuatan tabel rute diperuntukkan menyimpan data rute pada sistem informasi. Tabel rute merupakan proses pencarian rute secara statis karena rute pada studi kasus sistem ini dapat ditentukan dan tidak berubah. Adapun Tabel poin terdiri dari beberapa kolom yakni id, koridor\_asal, koridor\_tujuan, koridor via, halte transisi created\_at, updated\_at dan deleted\_at. Untuk implementasi serta hasilnya dapat dilihat pada Kode Sumber… dan Gambar….

CREATE TABLE `trans\_padang`.`rute` (

`id` INT NOT NULL,

`koridor\_asal` INT NULL,

`koridor\_tujuan` INT NULL,

`koridor\_via` INT NULL,

`halte\_transisi` INT NULL,

`created\_at` DATETIME NULL,

`updated\_at` DATETIME NULL,

`deleted\_at` DATETIME NULL,

PRIMARY KEY (`id`))

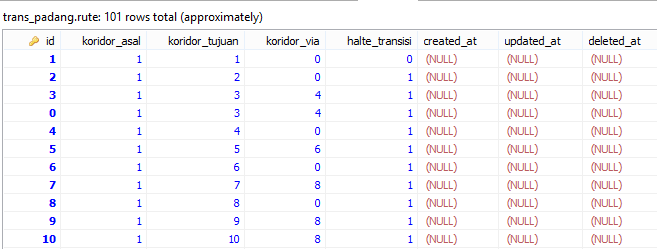
Kode Sumber 7 Membuat Tabel Rute

Pada Kode Sumber 7 dapat dilihat bahwa untuk id, koridor\_asal, koridor\_tujuan, koridor\_via dan halte\_transisi bertipe data integer. Adapun created \_at, updated\_at dan deleted\_at bertipe data datetime. Pada tabel ini id sebagai *primary key*.

Kode Sumber 8 Menyisipkan Data Pada Tabel Rute

**INSERT** **INTO** `trans\_padang`.`rute` (`id`, `koridor\_asal`, `koridor\_tujuan`, `koridor\_via`, `halte\_transisi`) **VALUES** (1, 1, 3, 4, 1);

Pada Kode Sumber8 merupakan salah satu contoh data rute yang disimpan dalam tabel. Kolom created\_at, updated\_at dan kolom deleted\_at belum terisi atau secara langsung terdefinisi dengan nilai null.



Gambar 4. 7 Cuplikan Basis Data Tabel Rute

Pada Gambar 4.7 memperlihatkan hasil tabel setelah dimasukkan sejumlah data yang diperlukan pada tabel rute. Tabel rute berfungsi menyimpan koridor asal dan tujuan serta koridor yang menghubungkannya. Adapun halte perpindahan antar koridor juga disimpan untuk pencarian rute yang akan dilewati.

## Implementasi Arsitektur Sistem

Implementasi arsitektur sistem merupakan langka untuk membangun sebuah sistem. Untuk detail implementasi arsitektur sistem ini akan dijelaksan pada sub bab berikutnya.

### Implementasi Model

Implementasi model merupakan fungsi-fungsi yang membantu dalam pengelolaan basis data seperti memasukkan data ke basis data, pembaruan data dan lain-lain. Adapun implementasi model pada sistem ini meliputi basis data yang terhubung dan keterikatan antara satu tabel dengan tabel lainnya dalam satu basis data. Untuk pendefinisian dan implementasi relasi antar tabel pada sistem ini dapat dilihat pada Kode Sumber…..

class Halte extends Model

{

use SoftDeletes;

protected $table = 'halte';

public $timestamps = true;

protected $dates = ['deleted\_at'];

public function Point()

{

return $this->hasMany('App\Model\Point');

}

}

Kode Sumber 9 Model Fungsi Point

Kode Sumber 9 merupakan fungsi dari tabel halte. Pada fungsi ini menjelaskan bahwa halte hanya berhubungan dengan tabel point dengan relasi *one to many* dimana halte menjadi tabel induk.

class Koridor extends Model

{

use SoftDeletes;

protected $table = 'koridor';

public $timestamps = true;

protected $dates = ['deleted\_at'];

public function Point()

{

return $this->hasMany('App\Model\Point');

}

public function Rute()

{

return $this->hasMany('App\Model\Rute');

}

}

Kode Sumber 10 Model Fungsi Koridor

Kode Sumber 10 Merupakan fungsi dari tabel koridor. Pada fungsi ini menjelaskan bahwa koridor berhubungan dengan tabel point dan tabel rute dengan relasi *one to many* dimana koridor menjadi tabel induk.

class Point extends Model

{

use SoftDeletes;

protected $table = 'point';

public $timestamps = true;

protected $dates = ['deleted\_at'];

public function Halte()

{

return $this->belongsTo('App\Model\Halte');

}

public function Koridor()

{

return $this->belongsTo('App\Model\Koridor');

}

}

Kode Sumber 11 Model Fungsi Point

Kode Sumber 11 merupakan fungsi dari tabel point. Pada fungsi ini menjelaskan bahwa point berhubungan dengan tabel halte dan koridor rute dengan relasi *one to many* dimana point menjadi tabel anak.

class Rute extends Model

{

use SoftDeletes;

protected $table = 'rute';

public $timestamps = true;

protected $dates = ['deleted\_at'];

public function koridor\_asal()

{

return $this->belongsTo('App\Model\Koridor');

}

public function koridor\_tujuan()

{

return $this->belongsTo('App\Model\Koridor');

}

}

Kode Sumber 12 Model Fungsi Rute

Sumber 12 merupakan fungsi dari tabel rute. Pada fungsi ini menjelaskan bahwa rute berhubungan dengan tabel koridor dengan relasi *one to many* dimana rute menjadi tabel anak.

### Implementasi View

Implementasi view merupakan fungs-fungsi bagian yang mengatur tampilan ke pengguna. Fungsi fungsi yang terdapat pada view ini akan menjadi sebuah antarmuka yang dijelaskan pada Bab 4.6. Untuk fungsi-fungsi view pada sistem ini dapat dilihat pada Kode Sumber 13

……….

<div class="main-content">

<div class="main-content-inner">

<div class="breadcrumbs" id="breadcrumbs">

<ul class="breadcrumb">

<li>

<i class="ace-icon fa fa-home home-icon"></i>

<a href="#">Home</a>

</li>

@yield('breadcrumb')

</ul>

</div>

<div class="page-content">

<div class="page-header">

<h1>

@yield('page-header')

</h1>

</div>

<div class="row">

<div class="col-xs-12"> @yield('content')

</div>

</div>

</div>

</div>

</div>

…….

Kode Sumber 13 View Tampilan Dasar

Kode Sumber 13 merupakan potongan kode HTML dan PHP pada tampilan dasar yang digunakan untuk tampilan lainnya. Pada tampilan dasar terdapat fungsi @yield yang dapat diisikan sesuai dengan kebutuhan dari tampilan turunannya. Pada @yield('page-header') diisikan berupa *header* dari halaman turunan yang akan ditampilkan, sedangkan @yield('content') diisikan berupa konten yang akan ditampilkan dari halaman turunan.

@extends('maintemplate')

……….

@section('content')

<div class="col-sm-4">

<label class="control-label no-padding-right" for="form-field-1"> Keberangkatan </label>

<br>

<label class="control-label no-padding-right" for="form-field-1"> Kedatangan </label>

</div>

<div class="default text">Keberangkatan

</div>

<div class="menu">

<?php foreach ($halte as $key => $value) {?>

<div class="item" data-value="<?php echo $value->id?>"></i> <?php echo $value->nama?> </div> <?php } ?>

</div>

@endsection

…….

@endsection

Kode Sumber 14 View Isian Keberangkatan dan Kedatangan

Kode Sumber 14 merupakan potongan kode HTML dan PHP yang menampilkan isian untuk halte keberangkatan dan kedatangan. Adapun potongan kode ini juga melakukan pencocokan isian dengan pemanggilan data halte yang ada pada sistem ini.

@extends('maintemplate')

……….

@section(‘js’)

…..

<script>

var fL = []

var groups= {};

var map = L.mapbox.map('map')

.setView([-0.908667,100.3872087], 13)

.addLayer(L.mapbox.tileLayer('mapbox.streets'));

L.mapbox.featureLayer()

.loadURL('/TransPadang/public/halte')

.on('ready', function(e) {

function makeGroup(color) {

return new L.MarkerClusterGroup({

iconCreateFunction: function(cluster) {

return new L.DivIcon({

iconSize: [20, 20],

html: '<div style="text-align:center;color:#fff;background:' +

color + '">' + cluster.getChildCount() + '</div>'

});

}

}).addTo(map);

}

…………</script>

……

});@endsection

…….

@endsection

Kode Sumber 15 View Pengelompkkan Halte dan Tampilan Peta

Kode Sumber 15 merupakan potongan kode Javascript untuk menampilkan peta serta melakukan klusterisasi halte. Untuk Potongan kode klusterisasi halte, digunakan pada halaman utama saat pertama kali ditampilkan.

@extends('maintemplate')

……….

@section(‘js’)

…..

<script>

var layer = []

function berhenti\_telusur(){

groups.blue.addTo(map); }

function telusuri()

{

var temp\_berangkat= document.getElementById("keberangkatan");

var halte\_berangkat = temp\_berangkat.value;

var temp\_datang= document.getElementById("kedatangan");

var halte\_datang = temp\_datang.value;

if(halte\_datang=='' || halte\_berangkat==''){return;}

for (var i = 0; i < layer.length; i++) {

map.removeLayer(layer[i])}

map.removeLayer(groups.blue);

…….

$.get( "/TransPadang/public/pencarian\_halte/"+halte\_berangkat+"/"+halte\_datang, function( data ) {

var geojson = JSON.parse(data);

var mark = L.mapbox.featureLayer(geojson);

mark.addTo(map);

layer.push(mark);

}); …………</script>

……

});@endsection

…….

@endsection

Kode Sumber 16 View Pencarian Js

Kode Sumber 16 merupakan potongan kode Javascript untuk menampilkan hasil pencarian rute. Potongan kode ini juga melakukan pemanggilan kode JSON yang berisikan sejumlah koordinat yang membentuk rute dari hasil eksekusi *controller* terkait.

@extends('maintemplate')

……….

@section('content')

<div id='filters' class='ui-select'>

<div>

<input type='checkbox' class='filter'

name='filter' id='Pergi' value='Pergi' onclick="change()"/><label for='Pergi'>Pergi (Pusat Kota)</label>

</div>

<div>

<input type='checkbox' class='filter'

name='filter' id='Pulang' value='Pulang' onclick="change()"/><label for='Pulang'>Pulang</label></div>

</div>

<div id="map" width="600" height="450" frameborder="0" style="border:0" allowfullscreen>

</div>

@endsection

…….

@endsection

Kode Sumber 17 View Checkbox Perkoridor

Kode Sumber 17 merupakan potongan kode HTML *checkbox* yang digunakan saat menampilkan rute dan halte perkoridor. Tampilan ini membantu untuk memperlihatkan rute dan halte pergi ataupun pulang sesuai dengan masukan dari pengguna.

@extends('maintemplate')

……….

@section('js')

…..

<script>

function change() {

for (var i = 0; i < layer.length; i++) {

map.removeLayer(layer[i])

}

var on = [];

for (var i = 0; i < checkboxes.length; i++) {

if (checkboxes[i].checked) on.push(checkboxes[i].value);

}

if (on.length == 2)

{ @yield('rute\_js\_pergi');

@yield('rute\_js\_pulang');

}

else if (on[0] == "Pergi")

{ @yield('rute\_js\_pergi'); }

else if (on[0] == "Pulang")

{ @yield('rute\_js\_pulang'); }

return false;

}

</script>@endsection

…….

@endsection

Kode Sumber 18 View Klusterisasi Rute Pulang atau Pergi

Kode Sumber 18 merupakan potongan kode Javascript yang melakukan penyaringan rute yang akan ditampilkan pada halaman turunan. Penyaringan rute ini dilakukan pada halaman tampilan rute dan halte perkoridor yang memungkinkan pengguna memilih menampilkan rute dan halte pergi atau pulang atau keduanya.

@extends('test\_ungroup')

….

@section('rute\_js\_pulang')

$.get( "/TransPadang/public/rute1a", function( data ) {

var line\_points = JSON.parse(data);

var polyline\_options = {

color: 'red'

};

var polyline = L.polyline(line\_points, polyline\_options).addTo(map);

layer.push(polyline);

});

$.get( "/TransPadang/public/halte\_ka1a", function( data ) {

var geojson = JSON.parse(data);

var mark = L.mapbox.featureLayer(geojson);

mark.addTo(map);

layer.push(mark);

});

@endsection

@section('rute\_js\_pergi')

……..

@endsection

Kode Sumber 19 View Rute dan Halte Pergi atau Pulang Json

Kode Sumber 19 merupakan potongan kode Javascript turunan dari kode Sumber 18. Potongan kode ini berfungsi pengambilan data dari *controller* dalam format JSON yang berisikan sejumlah koodrinat untuk menampilkan rute dan halte perkoridor. Untuk potongan kode pergi dan pulang hampir sama yang membedakan ialah variable pengambilan data dari *controller*.

@extends('maintemplate')

……….

@section('content')

……

<div id='filters' class='ui-select'>

<div><input type='checkbox' class='filter'

name='filter' id='Pergi' value='js\_koridor\_1' onclick="change()"/><label for='Koridor1'>Koridor 1</label></div>

<div><input type='checkbox' class='filter'

name='filter' id='Pulang' value='js\_koridor\_2' onclick="change()"/><label for='Koridor2'>Koridor 2</label></div>

<div><input type='checkbox' class='filter'

name='filter' id='Pergi' value='js\_koridor\_3' onclick="change()"/><label for='Koridor3'>Koridor 3</label></div>

<div><input type='checkbox' class='filter'

name='filter' id='Pulang' value='js\_koridor\_5' onclick="change()"/><label for='Koridor5'>Koridor 5</label></div>

<div><input type='checkbox' class='filter'

name='filter' id='Pergi' value='js\_koridor\_6' onclick="change()"/><label for='Koridor6'>Koridor 6</label></div>

</div> @endsection

…….

@endsection

Kode Sumber 20 View Checkbox Seluruh Rute dan Halte

Kode Sumber 20 merupakan potongan kode HTML *checkbox* yang digunakan saat menampilkan rute dan halte keseluruhan. Tampilan ini membantu untuk memperlihatkan koridor pilihan dari pengguna yang ingin ditampilkan.

@extends('maintemplate')

……….

@section('js')

…..

<script>

function change() {

for (var i = 0; i < layer.length; i++) { map.removeLayer(layer[i]) }

var on = [];

for (var i = 0; i < checkboxes.length; i++) {

if (checkboxes[i].checked) on.push(checkboxes[i].value); }

var temp = [];

for (var i = 0; i < on.length; i++)

{ if(on[i] == 'js\_koridor\_1')

{ temp[0] = 1; }

else if (on[i] == 'js\_koridor\_2')

{ temp[1] = 1; }

else if(on[i] == 'js\_koridor\_3')

{ temp[2] = 1; }

else if (on[i] == 'js\_koridor\_5')

{ temp[3] = 1; }

else

{ temp[4] = 1; }

}

if (temp[0] == 1)

{ @yield('js\_koridor\_1'); }

if (temp[1] == 1)

{ @yield('js\_koridor\_2'); }

if (temp[2] == 1)

{ @yield('js\_koridor\_3'); }

if (temp[3] == 1)

{ @yield('js\_koridor\_5'); }

if (temp[4] == 1)

{ @yield('js\_koridor\_6'); }

return false;

}</script>@endsection

…….

@endsection

Kode Sumber 21 View Logika Tampil Semua Koridor Js

Kode Sumber 21 merupakan potongan kode Javascript yang melakukan penyaringan rute yang akan ditampilkan pada halaman turunan. Penyaringan rute ini dilakukan pada halaman informasi halte dan rute keseluruhan yang memungkinkan pengguna memilih menampilkan koridor-koridor tertentu secara bersamaan.

@extends('koridor\_all')

@section('js\_koridor\_1')

$.get( "/TransPadang/public/rute1a", function( data ) {

var line\_points = JSON.parse(data);

var polyline\_options = {

color: 'red'

};

var polyline = L.polyline(line\_points, polyline\_options).addTo(map);

layer.push(polyline);

});

$.get( "/TransPadang/public/rute1b", function( data ) {

var line\_points = JSON.parse(data);

var polyline\_options = {

color: '#red'

};

var polyline = L.polyline(line\_points, polyline\_options).addTo(map);

layer.push(polyline);

});

$.get( "/TransPadang/public/halte\_ka1a", function( data ) {

……..

});

$.get( "/TransPadang/public/halte\_k1b", function( data ) {

……

});

@endsection

@section('js\_koridor\_1') ….. @endsection

…..

@endsection

Kode Sumber 22 View Tampil Semua Halted an Rute Json

Kode Sumber 22 merupakan potongan kode Javascript turunan dari kode Sumber 21 Potongan kode ini berfungsi pengambilan data dari *controller* dalam format JSON yang berisikan sejumlah koodrinat untuk menampilkan rute dan halte sesuai koridor yang ingin ditampilkan. Untuk potongan setiap koridornya hampir sama yang membedakan ialah variable pengambilan data dari *controller*.

### Implementasi Controller

Controller merupakan bagian yang menjembatani model dan view. Controller berisi perintah-perintah yang berfungsi untuk memproses suatu [data](https://id.wikipedia.org/wiki/Data) dan mengirimkannya ke halaman web. Oleh karena itu, controller mencakupi logika sistem serta proses pengambilan data dari basis data yang kemudian akan dicocokkan, ditampilkan atau diproses sistem dan tampilan

….

Class HomeController extends Controller

{

public function index()

{ return view('index'); }

public function koridor()

{ return view('information-koridor'); }

public function k\_all()

{ return view('k\_all'); }

public function k1()

{ return view('k1'); }

………….

}

Kode Sumber 23 Controller Inisialisasi View

Kode Sumber 23 merupakan potongan kode yang berfungsi mendefinisikan seluruh view yang akan ditampilkan pada sistem ini.

….

public function halte\_form()

{

$halte = Halte::where('id','>',0)->get();

return view("index")->with('halte',$halte);

// return json\_encode($halte);

}

………….

Kode Sumber 24 Controller Tampilan Utama

Kode Sumber 24 merupakan potongan kode yang berfungsi menampilkan view halaman utama. Selain itu, potongan kode sumber 24 ini juga mengirimkan hasil *query* berupa data halte yang difungsikan sebagai masukkan dalam pencarian rute.

public function rute1a()

{

$FeatureCollection = array();

$where = array('koridor\_id' => 1);

$rute = Point::with('Koridor')->where($where)->get();

foreach ($rute as $key => $value) {

$coordinate = array();

array\_push($coordinate, $value->latitude);

array\_push($coordinate, $value->longitude);

array\_push($FeatureCollection, $coordinate);

}

return json\_encode($FeatureCollection);

}

Kode Sumber 25 Controller Rute Perkoridor

Kode Sumber 25 merupakan potongan kode yang berisikan logika pencarian rute yang ada pada koridor tertentu (dalam hal ini pada rute 1a). Rute 1a merupakan rute menuju arah pulang atau mendekati pusat kota. Logika pencarian rute dilakukan dengan parameter id koridor dan dilakukan pencocokan id terhadap inputan dan data yang ada. Setelah dilakukan pencocokan, dibutuhkan koordinat. Setelah mendapatkan nilai koordinat, maka dilakukan pengiriman nilai yang selanjutnya diklasifikasikan dalam bentuk JSON

….

public function pencarian($awal,$akhir)

{

…..

while(1)

{

$halte\_transisi = Point::where('koridor\_id',$koridor\_awal)->where('halte\_id',$temp\_start)->first();

if($koridor\_awal==$koridor\_akhir){

$halte\_akhir = Point::where('koridor\_id',$koridor\_awal)->where('halte\_id',$akhir)->first();

$poin = Point::where('koridor\_id',$koridor\_awal)->whereBetween('nomor',array($halte\_transisi->nomor,$halte\_akhir->nomor))->get();

}

else $poin = Point::where('koridor\_id',$koridor\_awal)->where('nomor','>=',$halte\_transisi->nomor)->get();

foreach ($poin as $key => $value) {

$coordinates = array();

array\_push($coordinates, $value->latitude);

array\_push($coordinates, $value->longitude);

array\_push($poins,$coordinates);

}

if($koridor\_awal == $koridor\_akhir){ break; }

$next = Rute::where('koridor\_asal',$koridor\_awal)->where('koridor\_tujuan',$koridor\_akhir)->first();

$temp\_start = $next->halte\_transisi;

$koridor\_awal = $next->koridor\_via;

if($koridor\_awal==0){

$koridor\_awal = $next->koridor\_tujuan;

}

}

return json\_encode($poins);

}………….

Kode Sumber 26 Controller Pencarian Rute

Kode Sumber 26 merupakan potongan kode yang berisikan logika pencarian rute. Logika pencarian rute dilakukan dengan parameter id halte kedatangan dan keberangkatan yang kemudian diterjemahkan dalam koridor. Setelah penerjemahan ke dalam koridor, dilakukan pencarian rute menggunakan tabel rute. Perpindahan halte juga didefinisikan didalam logika ini sampai menemukan halte kedatangannya.

public function pencarian\_halte($awal,$akhir)

{

$FeatureCollection = array();

$halte = Point::with('Halte')->where('halte\_id',$awal)->orWhere('halte\_id',$akhir)->get();

foreach ($halte as $key => $value) {

$feature = array();

$feature['type'] = "Feature";

$feature['geometry'] = array();

$feature['geometry']['type'] = "Point";

$feature['geometry']['coordinates'] = array();

array\_push($feature['geometry']['coordinates'], $value->longitude);

array\_push($feature['geometry']['coordinates'], $value->latitude);

$feature['properties'] = array();

$feature['properties']['title'] = $value->nama;

$feature['properties']['description'] = $value->keterangan;

if($value->halte\_id==$akhir)

$feature['properties']['marker-color'] = '#fc4353';

else $feature['properties']['marker-color'] = '#00ff00';

$feature['properties']['marker-size'] = "medium";

$feature['properties']['marker-symbol'] = $value->Koridor->simbol;

array\_push($FeatureCollection, $feature);

}

return json\_encode($FeatureCollection);

}

Kode Sumber 27 Controller Pencarian Halte

Kode Sumber 27 merupakan potongan kode yang berisikan logika pencarian halte. Logika pencarian halte dilakukan dengan parameter id halte kedatangan dan dilakukan pencocokan id terhadap inputan dan data yang ada. Setelah dilakukan pencocokan, dibutuhkan pengambilan posisi berupa longitude dan latitude. Setelah mendapatkan nilai koordinat, maka dilakukan pengiriman nilai yang selanjutnya akan diklasifikasikan dalam bentuk JSON.

public function halte\_k1b()

{

$FeatureCollection = array();

$where = array('koridor\_id' => 2);

$halte = Point::with('Halte')-> where($where)->where('halte\_id','>',0)->get();

foreach ($halte as $key => $value) {

$feature = array();

$feature['type'] = "Feature";

$feature['geometry'] = array();

$feature['geometry']['type'] = "Point";

$feature['geometry']['coordinates'] = array();

array\_push($feature['geometry']['coordinates'], $value- >longitude);

array\_push($feature['geometry']['coordinates'], $value->latitude);

$feature['properties'] = array();

$feature['properties']['title'] = $value->nama;

$feature['properties']['description'] = $value->keterangan;

$feature['properties']['marker-color'] = $value->Halte->warna;

$feature['properties']['marker-size'] = "medium";

$feature['properties']['marker-symbol'] = $value->Koridor->simbol;

array\_push($FeatureCollection, $feature);

}

return json\_encode($FeatureCollection);

}

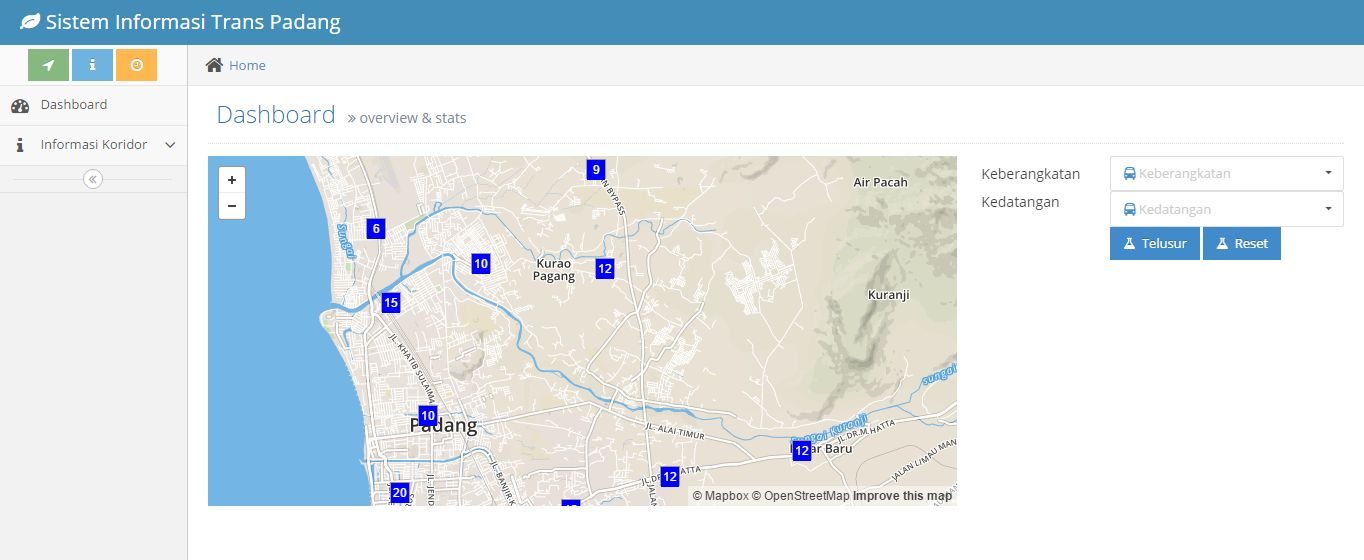
Kode Sumber 28 Controller Halte Perkoridor

Kode Sumber 28 merupakan potongan kode yang berisikan logika pencarian halte yang ada pada koridor tertentu (dalam hal ini pada koridor 1b). Koridor 1b merupakan koridor dengan rute menuju arah pergi atau menjauhi pusat kota. Logika pencarian halte dilakukan dengan parameter id koridor dan dilakukan pencocokan id terhadap inputan dan data yang ada. Setelah dilakukan pencocokan, dibutuhkan koordinat. Setelah mendapatkan nilai koordinat, maka dilakukan pengiriman nilai yang selanjutnya diklasifikasikan dalam bentuk JSON.

## Implementasi Antarmuka

Antarmuka pengguna merupakan hal yang penting dalam sebuah sistem. Antarmuka pengguna yang berhubungan langsung dengan aktor harus memiliki kemudahan-kemudahan bagi penggunanya. Sistem memiliki 3 antarmuka pengguna, yaitu halaman Halaman Utama, Informasi Keseluruhan dan Informasi Perkoridor

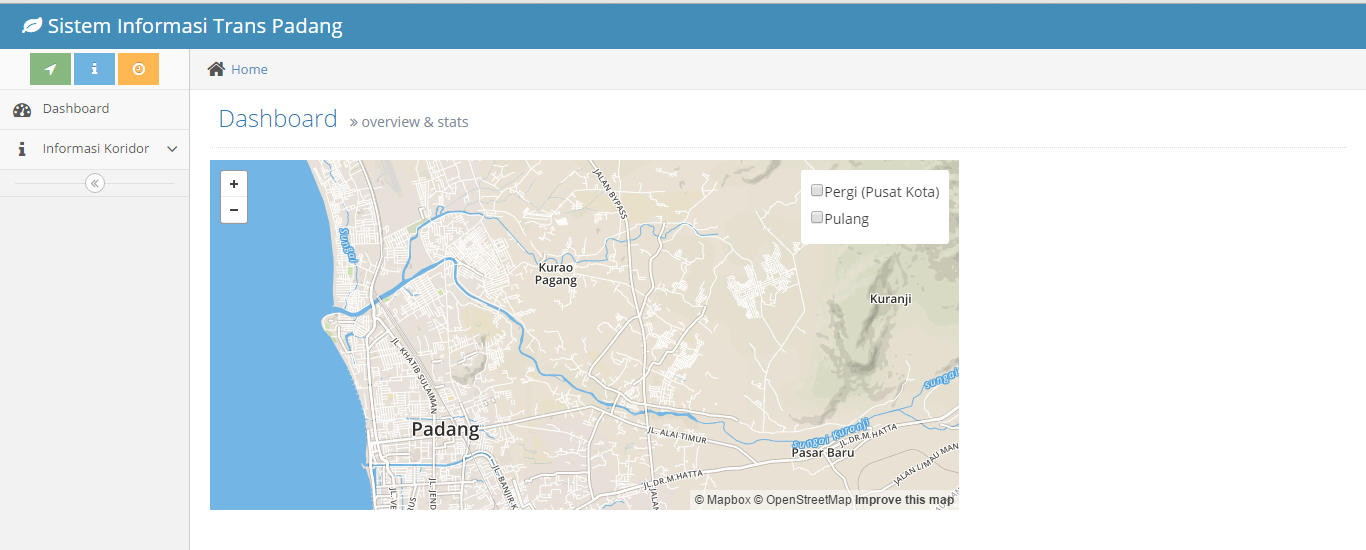
### Antarmuka Halaman Utama



Gambar 4.8 Impelmentasi Antarmuka Halaman Utama

Gambar 4.8 terdapat antarmuka navigasi yang menampilkan informasi rute dan halte. Bedasarkan gambar, juga terdapat form isian halte keberangkatan, kedatangan dan tombol penelusuran atau pencarian. Form isian berupa isian dan *dropdown* pencocokan dengan data yang ada pada sistem.

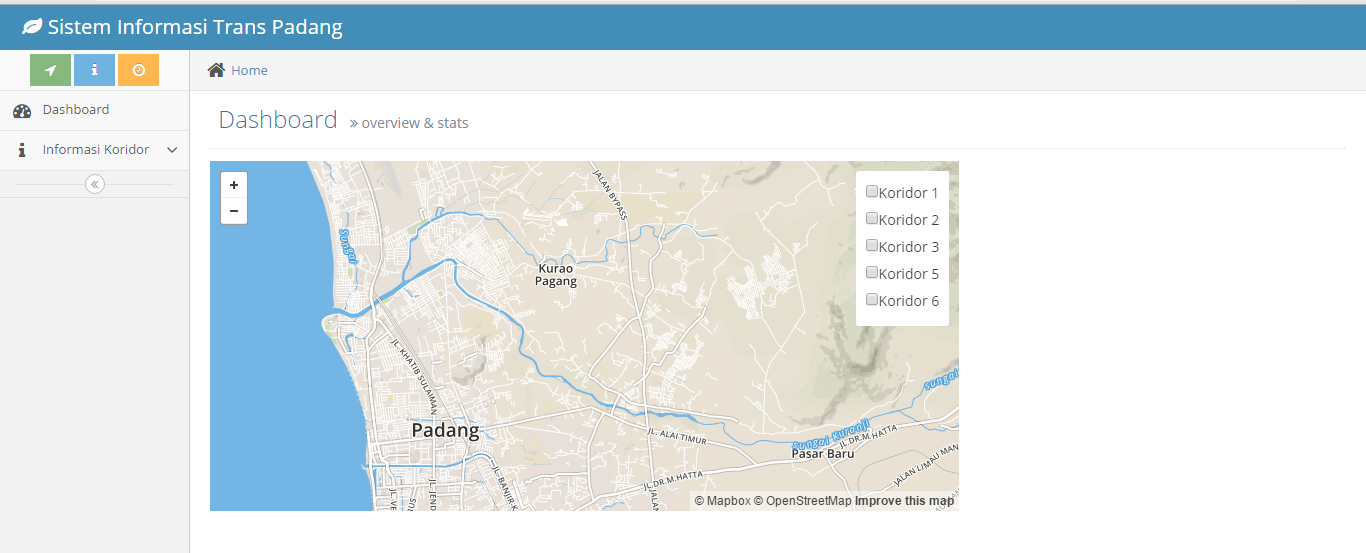
### Antarmuka Informasi Perkoridor



Gambar 4. 9 Implementasi Antarmuka Informasi Perkoridor

Gambar 4.9 merupakan antarmuka informasi rute dan halte perkoridor. Bedasarkan gambar, juga terdapat form isian berupa *checkbox* yang menjadi pilihan apaka ingin menampilkan rute dan halte pergi atau pulang atau keduanya.

### Antarmuka Informasi Semua Koridor



Gambar 4.10 Implementasi Antarmuka Informasi Semua Koridor

Gambar 4.10 Menampilkan antarmuka informasi rute dan halte semua koridor. Bedasarkan gambar, terdapat *checkbox* sebagai isian untuk pengguna dalam menampilkan informasi koridor yang ingin ditampilkan.

# BAB V PENGUJIAN DAN EVALUASI

Pada bab ini akan dijelaskan pengujian dan evaluasi dari aplikasi sistem akademik untuk modul mata kuliah ini. Pengujian yang dilakukan adalah pengujian dalam bentuk *control test* yang diselenggarakan pada waktu yang ditentukan.



## Lingkungan Pengujian

Lingkungan pengujian adalah lingkungan, baik perangkat keras maupun perangkat lunak tempat pengujian sistem dilakukan. Pengujian untuk modul mata kuliah dilakukan pada komputer dengan spesifikasi pada Tabel 18.

Tabel 18 Spesifikasi Sistem Pengujian

|  |  |
| --- | --- |
| Spesifikasi | Deskripsi |
| CPU | Intel ® Core ™ i3-3240 CPU @ 3.40 GHz |
| RAM | 4.00 GB |
| Sistem Operasi | Windows 8.1 Professional 64 bit |
| Koneksi Internet | Jaringan intranet dengan IP 10.151.31.33 dan port 8083 |

Pengujian dilakukan pada tanggal 4 Juni 2016 pukul 13.00-selesai bertempat di Laboratorium Manajemen Informasi Teknik Informatika ITS.

## Skenario Pengujian

Pada subbab ini dijelaskan mengenai skenario pengujian yang digunakan. Seperti yang sudah dijelaskan pada pengantar di bab 5, bahwa pengujian merupakan pengujian fungsionalitas yang menggunakan metode *blackbox.* Metode *blacbox* merupakan metode dimana pengujian ditekankan pada pola *input* dan *output* yang sesuai dengan skenario. Pengujian dilakukan dengan mengacu kasus penggunaan yang dijelaskan pada bab 3, analisis dan perancangan sistem.

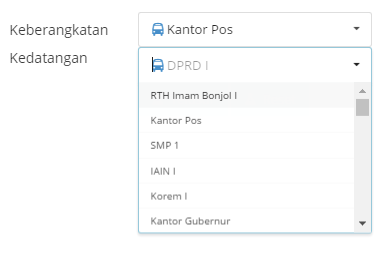
### Kasus Pengujian Melakukan Pencarian Rute

Pada kasus uji ini, pengguna dapat mencari rute bedasarkan halte keberangkatan dan kedatangan. Detail pengujian berupa skenario, kondisi awal, data uji dan kondisi akhir yang terdapat pada Tabel 19.

Tabel 19 Skenario Pencarian Rute

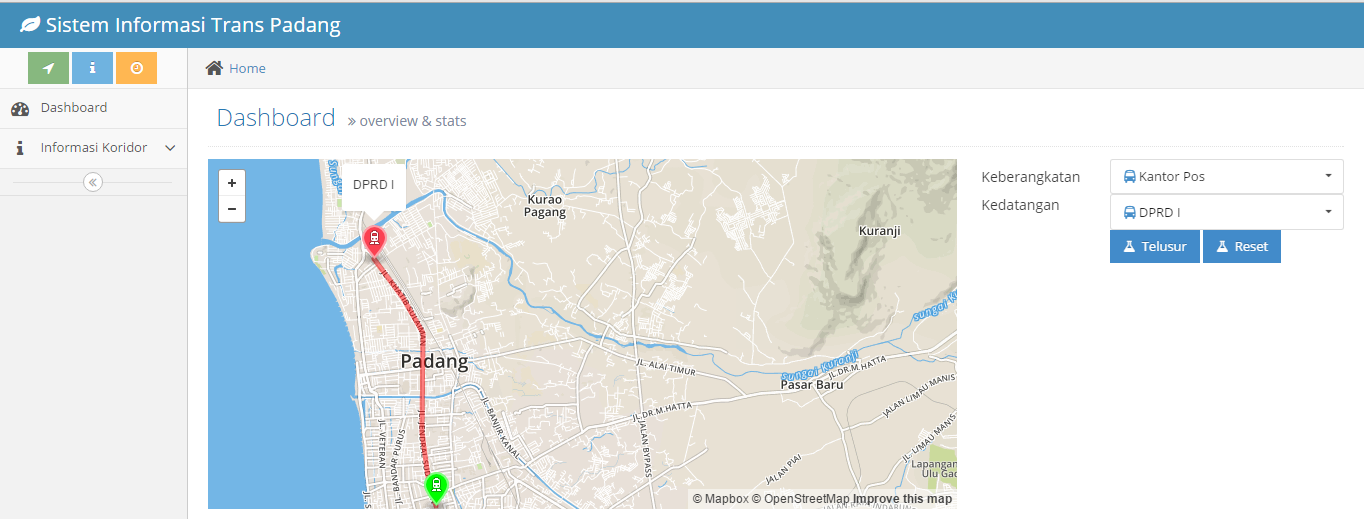
|  |  |
| --- | --- |
| ID | UJ-001 |
| Kasus Penggunaan | Pencarian Rute |
| Sub Kasus | - |
| Nama | Pengujian pencarian rute |
| Tujuan Pengujian | Menguji apakah sistem dapat melakukan pencarian dengan benar. |
| *Skenario 1* | Pengguna melakukan pencarian halte*.* |
| Kondisi Awal | Pencarian belum dilakukan |
| Data Uji | * Nama Halte Keberangkatan: * Nama Halte Kedatangan: |
| Langkah Pengujian | 1. Pengguna membuka halaman utama. 2. Pengguna memasukkan halte keberangkatan dan kedatangan. 3. Pengguna mengklik penanda halte untuk pengecekan nama halte |
| Hasil yang Diharapkan | Terlihat rute serta penanda halte keberangkatan dan kedatangan. |
| Hasil yang Didapat | Program menampilkan rute serta halte pencarian |
| Kondisi Akhir | Tampil hasil pencarian. |

Bentuk pengisian dilakukan dengan *input* pada sistem berupa teks dan *dropdown*. Pengujian dan hasil dari skenario adalah tampil rute serta halte keberangkatan dan kedatangan yang dapat dilihat pada Gambar 5. 1 dan Gambar 5. 2.



Gambar 5.1 Uji Coba Form Pengisian Keberangkatan dan Kedatangan

Pada Gambar 5.1 melihatkan formulir yang akan diuji coba. Adapaun data yang digunakan ialah halte Kantor Pos sebagai keberangakatan dan Halte DPRD I sebagai kedatangan.



Gambar 5. 2 Uji Coba Hasil Pencarian

Gambar 5.2 uji coba yang menampilkan hasil pencarian rute bedasarkan halte kedatangan dan keberangkatan. Bedasarkan gambar, diberikan variasi visualisasi warna rute dan halte asal dan tujuan serta dapat mengetahui nama halte dengan mengklik pada penanda halte.

### Kasus Pengujian Informasi Semua Koridor

Pada kasus uji ini pengguna dapat melihat informasi terkait halted an rute pada koridor yang diinginkan secara bersamaan dalam satu tampilan. Disediakan isian centang untuk menampilkan hasil. Detail pengujian berupa skenario, kondisi awal, data uji dan kondisi akhir yang terdapat pada Tabel 20.

Tabel 20 Skenario Melihat Informasi Semua Koridor

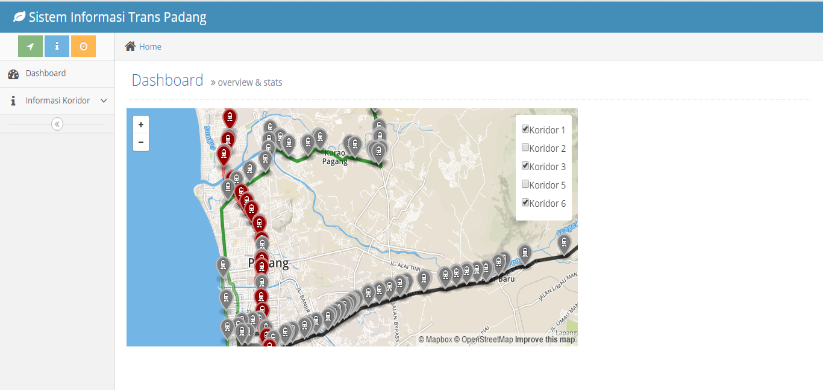
|  |  |
| --- | --- |
| ID | UJ-002 |
| Kasus Penggunaan | Melihat Informasi Semua Koridor |
| Sub Kasus | - |
| Nama | Pengujian melihat informasi semua koridor |
| Tujuan Pengujian | Menguji apakah informasi koridor berupa rute dan halte dapat ditampilkan. |
| *Skenario 1* | Pengguna melihat informasi rute dan halte |
| Kondisi Awal | Rute dan halte belum ditampilkan |
| Data Uji | - |
| Langkah Pengujian | 1. Pengguna membuka navigasi halaman melihat informasi koridor 2. Pengguna memilih halaman semua koridor. 3. Penggune mencentang koridor 1, 3 dan 6 |
| Hasil yang Diharapkan | Tampil informasi rute dan koridor. |
| Hasil yang Didapat | Program merespon dengan menampilkan informasi rute dan halte |
| Kondisi Akhir | Tampil informasi rute dan halte 1, 3 dan 6 |

Bentuk pengisian dilakukan dengan *input* pada sistem berupa *checkbox*. Hasil dari skenario adalah rute dan halte disetiap koridor yang diklasifikasikan dengan menampilkan rute pulang atau pergi atau keduanya. Tampilan informasi kondisi akhir dari skenario terdapat pada Gambar 5. 5 dan Gambar 5. 6.



Gambar 5.3 Uji Coba Halaman Menampilkan Informasi Seluruh Koridor

Gambar 5.3 merupakan halaman menampilkan informasi rute dan halte semua koridor. Bedasarkan gambar, terdapat *checkbox* sebagai isian untuk pengguna dalam menampilkan informasi koridor yang ingin ditampilkan. Adapun kondisi awalnya ialah kosong. Setelah dilakukan skenario maka akan tampil sesuai keinginan.



Gambar 5.4 Uji Coba Hasil Menampilkan Informasi Seluruh Koridor

Pada Gambar 5.4 merupakan antarmuka yang menampilkan informasi rute dan halte seluruh koridor. Warna rute dan halte akan dibedakan bedasarkan koridor.

### Kasus Pengujian Melihat Informasi Perkoridor

Pada subbab ini akan menjelaskan kasus uji untuk melihat informasi rute dan halte perkoridor. Pada sistem ini terdapat 5 koridor yakni koridor 1, 2, 3, 5 dan 6. Detail pengujian berupa cenario, kondisi awal, data uji dan kondisi akhir yang terdapat pada Tabel 5. 5.

Tabel 21 Uji Coba Melihat Informasi Perkoridor

|  |  |
| --- | --- |
| ID | UJ-003 |
| Kasus Penggunaan | Melihat Informasi Perkoridor |
| Sub Kasus | - |
| Nama | Pengujian melihat informasi perkoridor. |
| Tujuan Pengujian | Menguji *input* data pada penambahan prasyarat mata kuliah. |
| *Skenario 1* | Menguji apakah informasi rute dan halte disetiap koridor dapat ditampilkan. |
| Kondisi Awal | Rute dan halte belum ditampilkan |
| Data Uji | - |
| Langkah Pengujian | 1. Pengguna membuka navigasi halaman melihat informasi koridor 2. Pengguna memilih halaman koridor 1, 2, 3, 5 dan 6. 3. Pengguna mencentang untuk rute dan halte pulang serta pergi |
| Hasil yang Diharapkan | Tampil informasi rute dan koridor. |
| Hasil yang Didapat | Program merespon dengan menampilkan informasi rute dan halte disetiap koridor |
| Kondisi Akhir | Tampil informasi rute dan halte disetiap koridor |

Bentuk pengisian dilakukan dengan *input* pada sistem berupa *checkbox*. Hasil dari cenario adalah rute dan halte disetiap koridor yang diklasifikasikan dengan menampilkan rute pulang atau pergi atau keduanya. Tampilan informasi kondisi akhir dari skenario terdapat pada Gambar 5. 5 dan Gambar 5. 6.



Gambar 5. 5 Uji Coba Melihat Informasi Perkoridor

Gambar 5.5 merupakan halaman menampilkan informasi rute dan halte perkoridor. Bedasarkan gambar, terdapat *checkbox* sebagai isian untuk pengguna dalam menampilkan informasi rute yang akan ditampilkan, apakah pergi atau pulang atau keduanya. Adapun kondisi awalnya ialah kosong. Setelah dilakukan skenario maka akan tampil sesuai keinginan.

BAB VI  
KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan sebagai jawaban dari rumusan masalah yang dikemukakan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.



## Kesimpulan

Dari hasil uji coba yang telah dilakukan terhadap pembuatan sistem informasi Trans Padang, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi telah memiliki hasil yang baik dengan dapat menvisualisasi rute, map dan halte.
2. Aplikasi sudah mampu menampilkan informasi rute dan halte perkoridor ataupun keseluruhan.
3. Aplikasi sudah dapat membedakan tampilan informasi koridor menggunakan variasi warna rute dan halte disetiap koridornya.
4. Aplikasi sudah dapat melakukan pencarian rute yang bedasarkan halte yang diinginkan oleh pengguna.

## Saran

Saran yang diberikan untuk pengembangan aplikasi ini adalah:

1. Diharapkan kedepannya dapat ditampilkan untuk koridor 4 dikarenakan pada tugas akhir ini data koridor 4 dapat divisualisasikan.
2. Pada tugas akhir ini menampilkan jadwal keberangkatan dan kedatangan karena realisasi transportasi ini baru tahap pertama yakni koridor 1 dan kedepannya diharapkan dapat menampilkan jadwal ini.
3. Sistem informasi ini dapat dikembangkan dengan proses bisnis lainnya seperti pembelian tiket dan analisis penjualan tiket untuk pengembangan koridor, penambahan armada dan halte.

# DAFTAR PUSTAKA

|  |  |
| --- | --- |
| [1] | "Mapbox API," [Online]. Available: https://www.mapbox.com/api-documentation/. [Accessed 20 April 2016]. |
| [2] | "Tutorial Laravel Bahasa Indonesia," IDLARAVEL, [Online]. Available: http://id-laravel.com/post/kenapa-memilih-laravel/. [Accessed 20 April 2016]. |
| [3] | A. W. Imam, SQL Server 2000, Yogyakarta: Graha Ilmu, 2005. |
| [4] | N. Bunafit, PHP, Jakarta: Media, 2009. |
| [5] | jQuery Community Experts, jQuery Cookbook, Cambridge: O'Reilly Media, 2009. |

# LAMPIRAN

BIODATA PENULIS

Hafiz Nuzal Djufri pria kelahiran Padang, 19 Februari 1995. Saat ini menjabat sebagai Anggota Majelis Wali Amanat ITS Unsur wakil Mahasiswa 2016-2017. Memulai pendidikan pada tahun 1999 di TK Aisyiyah Padang, dilanjutkan ke SD Adabiah IV, SD Islam Budi Mulia Padang, MTsN Model Padang MAN 2 Padang dan Teknik Informatika ITS Surabaya. Selama perkuliahan, penulis aktif berorganisasi sebagai Staff Dalam Negeri HMTC ITS, Ketua National Big Event Schematics 2014 serta Ketua Mahkamah Mahasiswa ITS 2015-2016, Menteri Koordinator Dinamika Kampus Berani BEM ITS 2015-2016 dan Anggota Majelis Wali Amanat ITS Unsur wakil Mahasiswa 2016-2017. Sebelum masuk dunia kampus, penulis juga memiliki pengalaman organisasi penulis yakni Kepala Bidang Politik dan Kepemimpinan OSIS MAN 2 Padang, Ketua Remaja Masjid Baiturrahman Lapai Padang, Disamping itu, penulis juga sering menjadi pemateri di berbagai pelatihan baik ditingkat jurusan sampai tingkat Institut. Selain itu, penulis juga berpengalaman sebagai Project Manager aplikasi skor online Netlivescore PBSI, Project Manajer Sistem Informasi Keuangan RSU Aisyiyah Padang, Programmer kontrak Sistem Informasi SDM Semen Indonesia, Asisten Praktikum Jaringan Komputer Teknik Informatika ITS 2015 Programmer sistem informasi Perumahan dan Pemukiman dinas PU CKTR Sidoarjo.