

**RANCANG BANGUN APLIKASI  
SISTEM INFORMASI GEOGRAFIS  
PENENTUAN RUTE TERPENDEK MENUJU TAMBAL BAN  
DAN BENGKEL  
MENGGUNAKAN METODE DIJKSTRA BERBASIS  
WEBSERVICE (*Studi Kasus Kota Mojokerto*)**

**SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**FUAD ALI DAWAMI**

**NIM. 1341180126**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
AGUSTUS 2017**

## **HALAMAN JUDUL**

# **RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS PENENTUAN RUTE TERPENDEK MENUJU TAMBAL BAN DAN BENGKEL MENGGUNAKAN METODE DIJKSTRA BERBASIS WEBSERVICE (*Studi Kasus Kota Mojokerto*)**

## **SKRIPSI**

Digunakan Sebagai Syarat Maju Ujian Diploma IV

Politeknik Negeri Malang

**Oleh:**

**FUAD ALI DAWAMI**

**NIM. 1341180126**



**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG  
AGUSTUS 2017**

## **HALAMAN PENGESAHAN**

# **RANCANG BANGUN APLIKASI SISTEM INFORMASI GEOGRAFIS PENENTUAN RUTE TERPENDEK MENUJU TAMBAL BAN DAN BENGKEL MENGGUNAKAN METODE DIJKSTRA BERBASIS WEBSERVICE (*Studi Kasus Kota Mojokerto*)**

**Disusun oleh:**

**FUAD ALI DAWAMI      NIM. 1341180126**

**Skripsi ini telah diuji pada 30 Agustus 2017**

**Disetujui oleh:**

- |                 |   |                                    |       |
|-----------------|---|------------------------------------|-------|
| 1 Penguji I     | : | <u>Ely Setyo Astuti, ST., MT</u>   | ..... |
|                 |   | NIP.19760515 200912 2 001          |       |
| 2 Penguji II    | : | <u>Ekojono, ST., M.Kom</u>         | ..... |
|                 |   | NIP.19591208 198503 1 004          |       |
| 3 Pembimbing I  | : | <u>Budi Harijanto, ST., M.MKom</u> | ..... |
|                 |   | NIP.19620105 199003 1 002          |       |
| 4 Pembimbing II | : | <u>Usman Nurhasan, S.Kom., MT</u>  | ..... |
|                 |   | NIP. 19860923 201504 1 001         |       |

Mengetahui,

Ketua Jurusan  
Teknologi Informasi

Ketua Program Studi  
Teknik Informatika

Rudy Ariyanto, ST., MCs.  
NIP. 19711110 199903 1 002

Ir. Deddy Kusbianto P., M.Mkom.  
NIP. 19621128 198811 1 001

## **HALAMAN PERNYATAAN**

Dengan ini saya menyatakan bahwa Laporan Skripsi ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Ahli Madya/Kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Malang, 6 Juni 2017

Fuad Ali Dawami

## **ABSTRAK**

**Dawami, Fuad Ali.** “Rancang Bangun Aplikasi Sistem Informasi Geografis Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel Menggunakan Metode *Dijkstra* Berbasis *Webservice* (Studi Kasus Kota Mojokerto)”.

**Pembimbing: (1) Budi Harijanto, ST., M.Mkom. (2) Usman Nurhasan, S.Kom., MT**

**Skripsi, Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang, 2017.**

Salah satu faktor yang dapat mengakibatkan masalah dalam berkendara adalah mendapati ban kendaraan bocor di jalan. Dengan luas wilayah kota Mojokerto 16,465 km<sup>2</sup> dan jumlah tambal ban mencapai 113 maka jarak rata-rata antar tambal ban sekitar 0,145 km<sup>2</sup> dan jumlah tambal ban mencapai 93 maka jarak rata-rata antar bengkel sekitar 0,177 km<sup>2</sup> maka untuk mencari tambal ban dan bengkel dibutuhkan waktu dan tenaga yang tidak sedikit, apalagi jika tidak diketahui tempatnya. Hal tersebut memengaruhi kebutuhan akan layanan kendaraan bermotor seperti bengkel dan tambal ban, sehingga dibutuhkan media informasi untuk membantu mencari lokasi bengkel dan tambal ban berada. Sistem informasi geografis yang dibuat menggunakan algoritma yang ditemukan oleh *Dijkstra* untuk mencari path terpendek merupakan algoritma yang lebih efisien dibandingkan algoritma *Warshall*. Sistem berguna pada *user* yang berada di area kota Mojokerto. Dengan tujuan mempermudah *user* dalam mencari tambal ban dan bengkel terdekat serta dapat mempermudah *user* dalam mencari rute terdekat menuju tambal ban dan bengkel di area Kota Mojokerto.

**Kata Kunci:** Sistem Informasi Geografis, Pencarian Rute Terdekat, Metode *Dijkstra*.

## ***ABSTRACT***

**Dawami, Fuad Ali.** "The Development of Web-service Based Application of Geographic System Information of the Shortest Route Determination of Tire Repairs and Vehicle Service Using Dijkstra Method (Case Study of Mojokerto City) ".

**Advisors:** (1) **Budi Harijanto, ST., M.Mkom.** (2) **Usman Nurhasan, S.Kom., MT**

**Thesis, Informatics Engineering Study Program, Department of Information Technology, State Polytechnic of Malang, 2017.**

*One factor that may lead to problems while driving or riding is flat tyres on the road. With the area of Mojokerto is 16.465 km<sup>2</sup> and the total number of tire repair is 113, the average distance among tire repairs is around 0.145 km<sup>2</sup>. The total number of tire repairs is 93 where the average distance among vehicle service is about 0.177 km<sup>2</sup>. Therefore, to find a tire repair and vehicle service takes more time and efforts, especially if their places are unknown. It affects the needs of information media to help find the location of the vehicle service and the tire repairs. In developing this application, Geographic Information Systems created using the algorithm created by Dijkstra's is applied to find the shortest path. The system is useful for users especially in Mojokerto City to find the nearest tire repairs and vehicle service and to facilitate them in finding the shortest route to those places.*

**Keywords:** *Geographic Information System, Shortest Route Search, Dijkstra Method.*

## KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan laporan skripsi dengan judul “Rancang Bangun Aplikasi Sistem Informasi Geografis Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel Menggunakan Metode *Dijkstra* Berbasis *Webservice* (Studi Kasus Kota Mojokerto)”. Penyusunan laporan skripsi ini merupakan salah satu syarat untuk menyelesaikan studi program Diploma IV Program Studi Teknik Informatika, Jurusan Teknologi Informasi, Politeknik Negeri Malang.

Penulis menyadari bahwa tanpa adanya dukungan dan kerja sama dari berbagai pihak, penyusunan laporan skripsi ini tidak dapat diselesaikan dengan baik, untuk itu penulis menyampaikan rasa terima kasih kepada :

1. Allah SWT yang selalu memberi rahmat dan hidayah-Nya sehingga penulis diberi kelancaran untuk menyelesaikan Skripsi ini.
2. Kedua Orangtua tercinta yang selalu mendoakan demi kelancaran Skripsi.
3. Bapak Rudy Ariyanto,ST.,M.Cs., selaku ketua jurusan Teknologi Informasi.
4. Bapak Ir. Deddy Kusbianto P., M.Mkom. selaku ketua prodi Teknik Informatika.
5. Bapak Budi Harijanto,ST., M.MKom selaku Dosen Pembimbing I.
6. Bapak Usman Nurhasan,S.Kom., MT selaku Dosen Pembimbing II.
7. Dan seluruh pihak yang telah membantu dan mendukung lancarnya pembuatan.

Penulis menyadari bahwa dalam penulisan laporan skripsi ini masih belum sempurna serta masih banyak kekurangan dan kelemahan yang dimiliki oleh penulis baik itu bahasa maupun sistematika penulisan, untuk itu penulis mengharapkan saran dan kritik yang bersifat membangun dari berbagai pihak demi penyempurnaan laporan ini. Akhir kata, penulis ucapkan banyak terima kasih.

Malang, Juni 2017

Penulis

## DAFTAR ISI

	Halaman
HALAMAN PENGESAHAN .....	iii
HALAMAN PERNYATAAN .....	iv
ABSTRAK .....	v
<i>ABSTRACT</i> .....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI .....	viii
DAFTAR GAMBAR .....	xi
DAFTAR TABEL .....	xiv
DAFTAR LAMPIRAN .....	xv
BAB I. PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan dan Manfaat .....	3
1.4.1 Tujuan .....	3
1.4.2 Manfaat .....	3
1.5 Sistematika Penulisan Laporan .....	3
1.5.1 Bab I. Pendahuluan .....	3
1.5.2 Bab II. Landasan Teori .....	4
1.5.3 Bab III. Metodologi Penelitian .....	4
1.5.4 Bab IV. Analisa dan Perancangan Sistem .....	4
1.5.5 Bab V. Implementasi Sistem .....	4
1.5.6 Bab VI. Pengujian dan Pembahasan .....	4
1.5.7 Bab VII. Kesimpulan .....	5
BAB II. LANDASAN TEORI .....	6
2.1 Kota Mojokerto .....	6
2.2 Data Lokasi Tambal Ban dan Bengkel di Area Kota Mojokerto .....	7
2.3 Metode <i>Dijkstra</i> .....	8

2.4	Sistem Informasi .....	13
2.5	<i>Smartphone</i> .....	13
2.6	<i>Web Service</i> .....	13
2.7	Sistem Informasi Geografis.....	14
<b>BAB III.</b>	<b>METODOLOGI PENELITIAN.....</b>	<b>16</b>
3.1	Metodologi Penelitian .....	16
3.1.1	Pengumpulan Data.....	16
3.1.2	Analisa Kebutuhan .....	17
3.1.3	Perancangan Sistem.....	17
3.1.4	Implementasi Sistem .....	18
3.1.5	Pengujian Sistem .....	21
<b>BAB IV.</b>	<b>ANALISIS DAN PERANCANGAN .....</b>	<b>22</b>
4.1	Analisis Kebutuhan .....	22
4.1.1	Deskripsi Umum.....	22
4.1.2	Kebutuhan Perangkat Keras ( <i>Hardware</i> ) .....	23
4.1.3	Kebutuhan Perangkat Lunak ( <i>Software</i> ) .....	23
4.2	Perancangan Sistem.....	23
4.2.1	<i>Work Breakdown Structure (WBS)</i> .....	24
4.2.2	<i>Use Case</i> .....	24
4.2.3	<i>Flowchart</i> .....	32
4.2.4	Desain <i>Database ERD (Entity Relationship Diagram)</i> .....	33
4.2.5	Desain Antarmuka ( <i>Interface</i> ) .....	34
4.2.6	<i>Interface</i> Halaman Menu Utama .....	34
4.2.7	<i>Interface</i> Halaman <i>login</i> Petugas Administrasi .....	35
4.2.8	<i>Interface</i> Halaman Petugas Administrasi .....	36
4.2.9	Perancangan Metode Dijkstra.....	38
<b>BAB V.</b>	<b>IMPLEMENTASI.....</b>	<b>42</b>
5.1	Implementasi Basis Data.....	42
5.1.1	Gambar Tabel Admin .....	42
5.1.2	Gambar Tabel Graph .....	43
5.1.3	Gambar Tabel Bengkel_tambalan .....	44

5.2	Implementasi Antar Muka.....	44
5.2.1	Halaman Utama .....	44
5.2.2	Fitur Tujuan dan <i>RUN</i> .....	47
5.2.3	Fitur Menu Bengkel.....	50
5.2.4	Fitur Menu Tambal Ban .....	50
5.2.5	Halaman <i>Login</i> .....	51
5.2.6	Halaman Utama <i>Admin</i> .....	52
5.2.7	Halaman Menampilkan Data <i>Map</i> .....	56
5.2.8	Halaman Menampilkan Lokasi <i>Map</i> .....	57
5.2.9	Halaman Tambah Data .....	58
BAB VI.	PENGUJIAN DAN PEMBAHASAN .....	61
6.1	Uji Coba .....	61
6.1.1	Pengujian Spesifikasi Perangkat Uji Coba .....	61
6.2	Pengujian Performa Fungsionalitas Sistem.....	62
6.3	Analisis Hasil Uji Coba Sistem .....	63
6.4	Hasil Uji Coba dengan Responden.....	64
6.5	Pengujian Algoritma <i>Dijkstra</i> .....	65
BAB VII.	KESIMPULAN.....	71
7.1	Kesimpulan.....	71
7.2	Saran.....	71
	DAFTAR PUSTAKA .....	72
	DAFTAR LAMPIRAN .....	73

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Peta Kota Mojokerto.....	6
<b>Gambar 2. 2</b> Lokasi Tambal Ban dan Bengkel.....	7
<b>Gambar 2. 3</b> Penerapan metode <i>dijkstra</i> .....	8
<b>Gambar 3. 1</b> Perancangan Sistem .....	18
<b>Gambar 3. 2</b> Jalur perhitungan <i>dijkstra</i> . ....	19
<b>Gambar 3. 3</b> Perhitungan <i>dijkstra</i> .....	19
<b>Gambar 3. 4</b> Perhitungan <i>dijkstra</i> .....	20
<b>Gambar 3. 5</b> Perhitungan <i>dijkstra</i> .....	20
<b>Gambar 3. 6</b> Jalur yang di lewati.....	21
<b>Gambar 4. 1</b> <i>Work Breakdown Structure</i> (WBS). ....	24
<b>Gambar 4. 2</b> Diagram <i>Use case</i> .....	25
<b>Gambar 4. 3</b> <i>Flowchart</i> pencarian lokasi bengkel dan tambal ban. ....	32
<b>Gambar 4. 4</b> <i>Entity Relational</i> Diagram. ....	33
<b>Gambar 4. 5</b> Desain <i>interface</i> halaman menu utama.....	34
<b>Gambar 4. 6</b> Desain <i>interface</i> halaman <i>login</i> petugas administrasi. ....	35
<b>Gambar 4. 7</b> Desain <i>interface</i> halaman utama petugas administrasi menu data bengkel dan tambal ban.....	36
<b>Gambar 4. 8</b> Desain <i>interface</i> halaman utama petugas administrasi menu menampilkan data <i>map</i> . ....	36
<b>Gambar 4. 9</b> Desain <i>interface</i> halaman utama petugas administrasi menu menampilkan lokasi <i>map</i> .....	37
<b>Gambar 4. 10</b> Desain <i>interface</i> halaman utama petugas administrasi menu tambah dan <i>edit</i> data. ....	37
<b>Gambar 4. 11</b> Perancangan metode <i>dijkstra</i> .....	39
<b>Gambar 4. 12</b> Perhitungan metode <i>dijkstra</i> . ....	40
<b>Gambar 4. 13</b> Hasil perhitungan metode <i>dijkstra</i> .....	41
<b>Gambar 5. 1</b> <i>Database</i> . ....	42
<b>Gambar 5. 2</b> Tabel Admin.....	43

<b>Gambar 5. 3</b> Tabel <i>Graph</i> .....	43
<b>Gambar 5. 4</b> Tabel bengkel_tambalban.....	44
<b>Gambar 5. 5</b> Halaman utama.....	45
<b>Gambar 5. 6</b> <i>script</i> untuk menampilkan <i>marker</i> .....	45
<b>Gambar 5. 7</b> <i>script</i> untuk menampilkan peta dan <i>user</i> .....	46
<b>Gambar 5. 8</b> <i>script</i> untuk membedakan warna <i>marker</i> sesuai jenis.....	46
<b>Gambar 5. 9</b> Fitur <i>RUN</i> .....	47
<b>Gambar 5. 10</b> <i>script</i> untuk menampilkan rute terdekat.....	48
<b>Gambar 5. 11</b> <i>script</i> untuk menampilkan rute terdekat.....	48
<b>Gambar 5. 12</b> <i>script</i> untuk menampilkan rute terdekat.....	49
<b>Gambar 5. 13</b> <i>script</i> untuk menampilkan rute terdekat.....	49
<b>Gambar 5. 14</b> Fitur Bengkel.....	50
<b>Gambar 5. 15</b> Fitur Bengkel.....	50
<b>Gambar 5. 16</b> Halaman <i>Login</i> .....	51
<b>Gambar 5. 17</b> <i>script</i> halaman <i>login</i> .....	51
<b>Gambar 5. 18</b> <i>script</i> cek <i>login</i> .....	52
<b>Gambar 5. 19</b> Halaman Utama <i>Admin</i> .....	53
<b>Gambar 5. 20</b> <i>script</i> halaman <i>admin</i> .....	53
<b>Gambar 5. 21</b> <i>script button edit</i> .....	54
<b>Gambar 5. 22</b> <i>script button delete</i> .....	54
<b>Gambar 5. 23</b> Halaman <i>edit</i> .....	55
<b>Gambar 5. 24</b> <i>script</i> untuk menampilkan halaman <i>edit</i> .....	55
<b>Gambar 5. 25</b> Halaman Menampilkan Data <i>Map</i> .....	56
<b>Gambar 5. 26</b> <i>script</i> untuk menampilkan halaman "Menampilkan Data <i>Map</i> ". .....	56
<b>Gambar 5. 27</b> Halaman Menampilkan Lokasi <i>Map</i> .....	57
<b>Gambar 5. 28</b> <i>script</i> untuk menampilkan halaman "Menampilkan Lokasi <i>Map</i> ". ....	57
<b>Gambar 5. 29</b> <i>script</i> untuk menampilkan halaman "Menampilkan Lokasi <i>Map</i> "....	58
<b>Gambar 5. 30</b> Halaman Tambah Data.....	58
<b>Gambar 5. 31</b> <i>script</i> untuk menampilkan halaman "Tambah Data".....	59
<b>Gambar 5. 32</b> <i>script</i> untuk menampilkan halaman "Tambah Data".....	59

<b>Gambar 5.33</b> <i>script</i> untuk menampilkan halaman "Tambah Data".....	60
<b>Gambar 6.1</b> Contoh perhitungan pada program. .....	66
<b>Gambar 6.2</b> Perhitungan metode <i>dijkstra</i> . .....	67
<b>Gambar 6.3</b> Peta pada Aplikasi. .....	68
<b>Gambar 6.4</b> Peta pada <i>google maps</i> .....	68
<b>Gambar 6.5</b> Peta pada <i>google maps</i> .....	69
<b>Gambar 6.6</b> Peta pada <i>google maps</i> .....	69
<b>Gambar 6.7</b> Peta pada <i>google maps</i> .....	70

## DAFTAR TABEL

<b>Tabel 4. 1</b> Tabel fungsi aktor <i>use case</i> .....	25
<b>Tabel 4. 2</b> Definisi <i>use case</i> .....	25
<b>Tabel 4. 3</b> Deskripsi <i>use case "Login"</i> .....	26
<b>Tabel 4. 4</b> Deskripsi <i>use case "Logout"</i> .....	27
<b>Tabel 4. 5</b> Deskripsi <i>use case "Lihat data bengkel dan tambal ban"</i> .....	28
<b>Tabel 4. 6</b> Deskripsi <i>use case "Olah data bengkel dan tambal ban"</i> . .....	29
<b>Tabel 4. 7</b> Deskripsi <i>use case "Marker</i> lokasi bengkel dan tambal ban". .....	30
<b>Tabel 4. 8</b> Deskripsi <i>use case</i> memilih bengkel dan tambal ban sesuai jarak.....	31
<b>Tabel 4. 9</b> Tabel keterangan menu utama.....	34
<b>Tabel 4. 10</b> Tabel keterangan menu <i>login</i> . .....	35
<b>Tabel 4. 11</b> Tabel keterangan halaman menu petugas administrasi. ....	38
<b>Tabel 4. 12</b> Jarak antar titik. ....	39
<b>Tabel 4. 13</b> Daftar Rute. ....	41
<b>Tabel 6. 1</b> Spesifikasi perangkat keras <i>PC / Laptop</i> .....	61
<b>Tabel 6. 2</b> Spesifikasi perangkat keras <i>smartphone</i> . .....	61
<b>Tabel 6. 3</b> Spesifikasi perangkat lunak <i>PC / Laptop</i> . .....	62
<b>Tabel 6. 4</b> Spesifikasi perangkat lunak <i>smartphone</i> .....	62
<b>Tabel 6. 5</b> Fungsionalitas Sistem.....	62
<b>Tabel 6. 6</b> Data hasil kuesioner. .....	64
<b>Tabel 6. 7</b> Grafik presentase kuisioner. .....	65
<b>Tabel 6. 8</b> Daftar rute.....	67

## **DAFTAR LAMPIRAN**

Lampiran 1 Lembar Bimbingan Skripsi I.....	73
Lampiran 2 Lembar Bimbingan Skripsi II .....	74
Lampiran 3 Lembar Revisi Skripsi II.....	75
Lampiran 4 Lembar Revisi Skripsi II.....	76
Lampiran 5 Form Verifikasi Bahasa Inggris dan Tata Tulis Buku Skripsi.....	77
Biodata Penulis.....	78

## **BAB I. PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam bidang teknologi informasi, di era yang modern ini kecepatan akses informasi telah menjadi popularitas, dalam hal ini sebagian besar manusia lebih banyak menggunakan internet sebagai jasa yang utama, tidak hanya mudah diakses namun menghemat waktu dan biaya. Fasilitas dari *software* yang dapat diakses oleh internet tak urung mempermudah konsumen dalam memperoleh informasi dengan mudah.

Berdasarkan observasi awal yang telah dilakukan di tiga tempat Kantor Badan Pusat Statistik Kota Mojokerto, Kantor SAMSAT Kota Mojokerto, dan melakukan observasi di area Kota Mojokerto di peroleh hasil sebagai berikut : Pertama di Kantor Badan Pusat Statistik Kota Mojokerto di peroleh hasil bahwa luas wilayah Kota Mojokerto  $16,465 \text{ Km}^2$ .<sup>[2]</sup> yang kedua di lakukan observasi di Kantor SAMSAT Kota Mojokerto di peroleh hasil bahwa jumlah pembelian kendaraan bermotor baru roda dua di area Kota Mojokerto mencapai 39.389 pada tahun 2012, 50.518 pada tahun 2013, 53.803 pada tahun 2014, 45.486 pada tahun 2015.<sup>[7]</sup> dan yang ketiga di lakukan observasi di area Kota Mojokerto, di peroleh hasil bahwa untuk jumlah tambal ban yang berada di area Kota Mojokerto mencapai 113 dan jumlah bengkel yang berada di area Kota Mojokerto mencapai 93 baik resmi atau tidak resmi, dari hasil observasi di 5 lokasi tambal ban rata-rata jumlah pengguna kendaraan bermotor yang membutuhkan jasa tambal ban 5 sampai 10/hari, dari hasil observasi di 5 lokasi bengkel rata-rata jumlah pengguna kendaraan bermotor yang membutuhkan jasa bengkel 10 sampai 15 sepeda motor/hari.

Dengan data tersebut dapat disimpulkan betapa penting nya kebutuhan jasa tambal ban dan bengkel. Dengan luas wilayah kota Mojokerto  $16,465 \text{ km}^2$  dan jumlah tambal ban mencapai 113 maka jarak rata-rata antar tambal ban sekitar  $0,145 \text{ km}^2$  dan jumlah tambal ban mencapai 93 maka jarak rata-rata antar bengkel sekitar  $0,177 \text{ km}^2$  sehingga untuk mencari tambal ban dan bengkel membutuhkan waktu dan tenaga yang tidak sedikit, jika tidak mengetahui tempatnya. Salah satu faktor yang dapat

mengakibatkan masalah dalam berkendara adalah mendapati ban kendaraan bocor di jalan karena tertusuk paku, terkena jebakan, ataupun sebab lain. Hal tersebut memengaruhi kebutuhan akan layanan kendaraan bermotor seperti bengkel dan tambal ban, sehingga dibutuhkan media informasi untuk membantu mencari lokasi bengkel dan tambal ban berada.

Sistem Informasi Geografis sebagai salah satu bidang ilmu yang sedang berkembang pesat, saat ini telah mampu menyelesaikan permasalahan *routing*, contohnya masalah pencarian rute terpendek (*shortest path*). Sistem Informasi Geografis sudah banyak dikembangkan menjadi aplikasi-aplikasi *GIS* mulai dari berbasis desktop, berbasis *web* maupun berbasis aplikasi *mobile* dalam berbagai bidang.

Aplikasi ini nantinya diharapkan bisa membantu pengendara bermotor untuk menemukan rute terpendek menuju tambal ban dan bengkel di area kota Mojokerto. Dengan kemajuan teknologi yang semakin canggih tercetus ide untuk membuat "Aplikasi Sistem Informasi Geografis Tentang Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel Menggunakan Metode *Dijkstra* Berbasis *Webservice* (Studi Kasus Kota Mojokerto)".

## 1.2 Rumusan Masalah

Berdasarkan latar belakang di atas, maka permasalahan yang dapat penulis ambil dalam skripsi ini adalah :

1. Bagaimana membangun aplikasi penentu rute terpendek menuju tambal ban dan bengkel menggunakan metode *Dijkstra* berbasis *web service* ?
2. Bagaimana menampilkan rute terpendek menuju tambal ban dan bengkel pada aplikasi ?

## 1.3 Batasan Masalah

Batasan masalah dalam skripsi penulis yang berjudul Rancang Bangun Sistem Informasi Geografis Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel dengan Metode *Dijkstra* Berbasis *Webservice* (Studi Kasus Kota Mojokerto):

1. Lokasi tambal ban dan bengkel hanya berada di area sekitar jalan kota Mojokerto tidak termasuk jalan gang-gang.
2. Data penelitian ini terbatas pada kendaraan ringan (sepeda, sepeda motor).
3. Data spasial yang digunakan adalah lokasi tambal ban dan bengkel, serta data jalan.
4. *Input* data berupa titik pusat pemakai sistem sebagai titik nol (awal).
5. *Output* yang dihasilkan berupa peta, rute yang dilalui dan jarak tempuh yang digunakan antara tempat pemakai sistem menuju tempat tambal ban dan bengkel yang terdekat.

## 1.4 Tujuan dan Manfaat

### 1.4.1 Tujuan

Tujuan penulis dalam penelitian ini yaitu membuat aplikasi yang dapat membantu *user* khususnya pengguna kendaraan ringan (sepeda, sepeda motor) untuk menentukan rute terpendek menuju tambal ban dan bengkel menggunakan metode *Dijkstra*.

### 1.4.2 Manfaat

Manfaat dalam penelitian ini adalah sebagai berikut :

1. Dapat mempermudah masyarakat dalam mencari rute terpendek menuju Tambal Ban dan Bengkel yang terdapat di Kota Mojokerto.
2. Terciptanya aplikasi Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel menggunakan Metode *Dijkstra* untuk membantu memberikan informasi rute terpendek menuju tambal ban dan bengkel di dalam aplikasi dalam bentuk peta yang berbasis *webservice*.
3. Waktu yang digunakan oleh pengguna sistem (*user*) untuk mencari tambal ban dan bengkel menjadi lebih singkat dan rute menjadi lebih dekat.

## 1.5 Sistematika Penulisan Laporan

### 1.5.1 Bab I. Pendahuluan

Pada bab ini menjelaskan tentang latar belakang diadakannya penelitian ini

dan yang menjadi dasar permasalahan, yang meliputi latar belakang masalah, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, dan sistematika penulisan.

### **1.5.2 Bab II. Landasan Teori**

Bagian ini menjelaskan mengenai teori-teori diantaranya mengenai pengenalan dari obyek yaitu kalangan pengendara roda dua, pengenalan sistem informasi geografis, implementasi metode *Dijkstra*, dan *framework* yang akan digunakan untuk membangun sistem.

### **1.5.3 Bab III. Metodologi Penelitian**

Bab ini menjelaskan tentang metode yang di pilih, teknik, prosedur yang tepat, dan tools apa yang digunakan sehingga setiap tahap penelitian dapat dilakukan dengan tepat.

### **1.5.4 Bab IV. Analisa dan Perancangan Sistem**

Bab ini menjelaskan tentang perencanaan dan pembuatan sistem secara keseluruhan dan analisa terhadap hasil dari data yang sudah didapat.

### **1.5.5 Bab V. Implementasi Sistem**

Pada bab ini menjelaskan mengenai bagaimana sistem atau aplikasi yang dibuat dapat berjalan menurut perancangan dan analisa yang telah dilakukan sebelumnya. Sistem ini dapat menganalisa data berupa titik poin di mana *user* berada yang *diinputkan* oleh pengguna atau *user* untuk mendapatkan hasil rute terpendek menuju tambal ban dan bengkel menggunakan metode *Dijkstra*.

### **1.5.6 Bab VI. Pengujian dan Pembahasan**

Pada bab ini menjelaskan mengenai pengujian sistem yang dilakukan untuk megetahui performa sistem dalam penggunaanya, hal ini dapat berupa tampilan *form* dan *output* yang diimplementasikan dari proses analisa data yang berupa titik poin di mana user berada yang *diinputkan* oleh pengguna atau *user* menggunakan metode *Dijkstra* yang selanjutnya akan menghasilkan rute terpendek menuju tambal ban dan bengkel.

### **1.5.7 Bab VII. Kesimpulan**

Pada bab ini menjelaskan mengenai kesimpulan untuk menjawab permasalahan yang dihadapi dan saran yang berisikan solusi *alternatif* untuk permasalahan yang terjadi pada laporan akhir ini.

## BAB II. LANDASAN TEORI

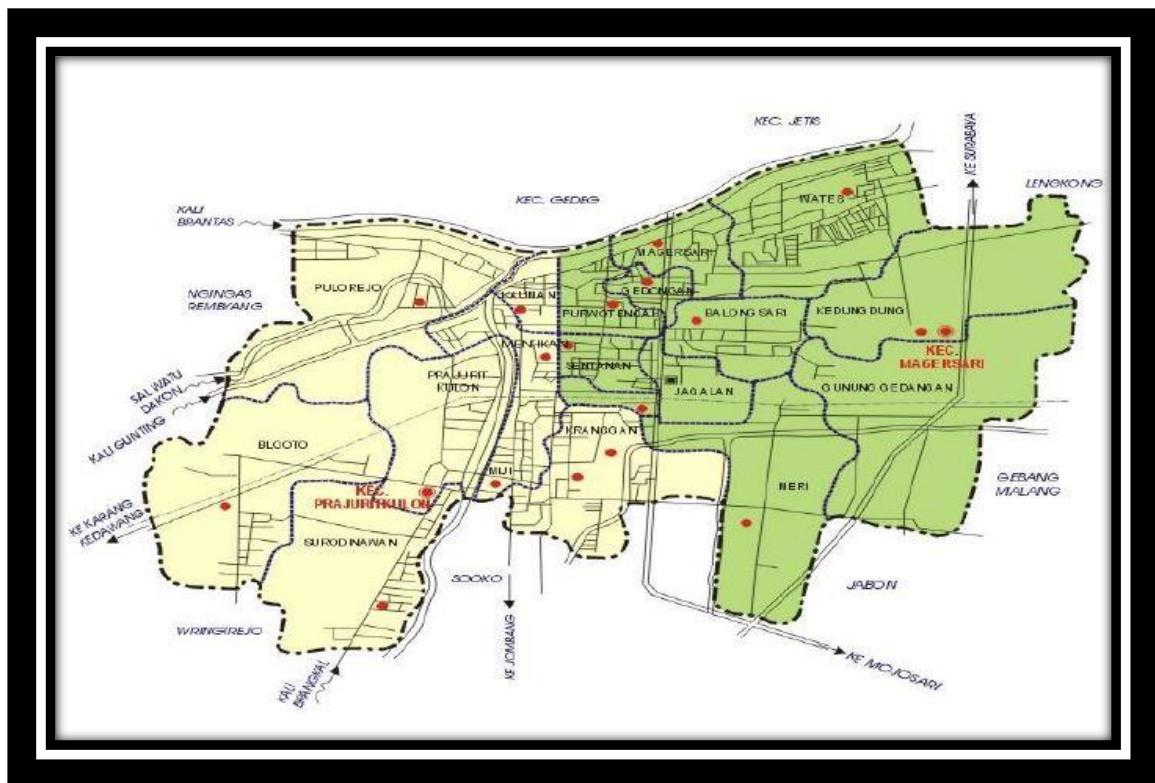
### 2.1 Kota Mojokerto

Luas keseluruhan wilayah kota Mojokerto adalah 16.465 km<sup>2</sup>. Kota Mojokerto hanya terbagi atas 3 Kecamatan yaitu Kecamatan Prajurit Kulon, Kecamatan Magersari, dan Kecamatan Kranggan yang terdiri dari 10 Kelurahan, 105 RW, dan 377 RT.

Secara geografis Wilayah Kota Mojokerto berada di antara 7°33' LS dan 122°28' BT dengan batas-batas wilayah sebagai berikut :

- Sebelah Utara : Sungai Brantas  
Sebelah Timur : Kecamatan Puri Kabupaten Mojokerto  
Sebelah Selatan : Kecamatan Sooko dan Puri Kabupaten Mojokerto  
Sebelah Barat : Kecamatan Sooko Kabupaten Mojokerto.<sup>[2]</sup>

Peta Kota Mojokerto di tunjukkan pada **gambar 2.1**.

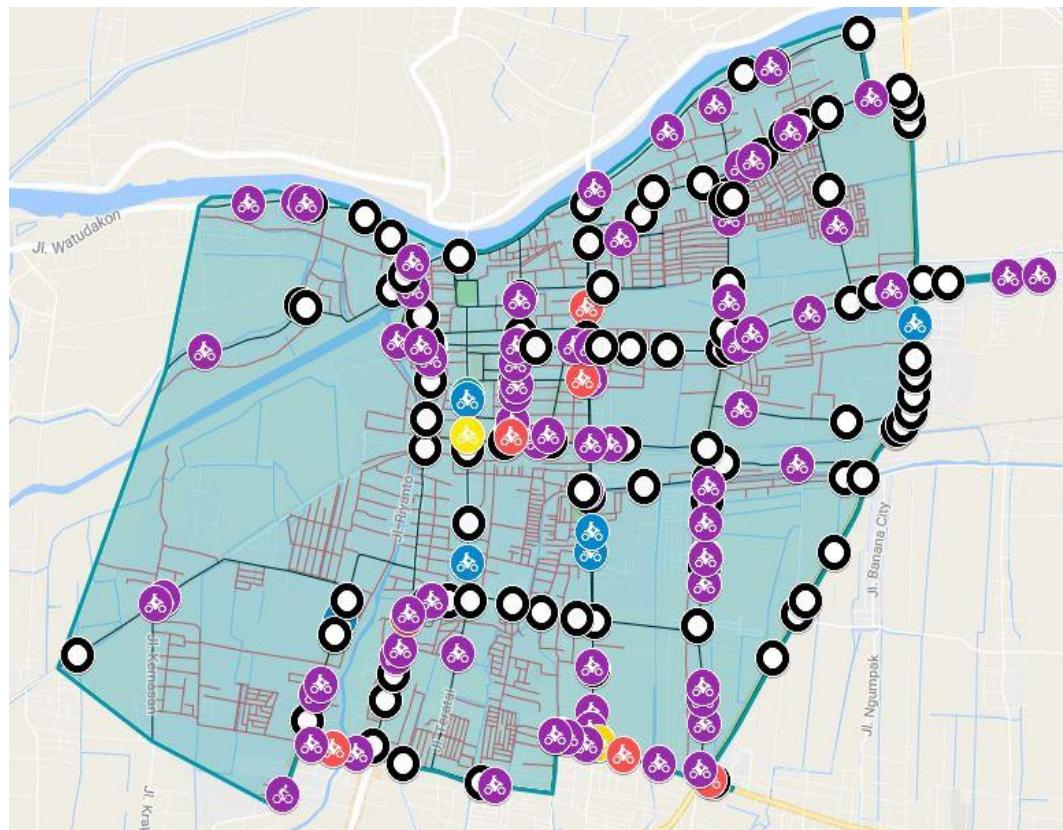


**Gambar 2. 1** Peta Kota Mojokerto.

## 2.2 Data Lokasi Tambal Ban dan Bengkel di Area Kota Mojokerto

Data Lokasi Tambal Ban dan Bengkel di Area Kota Mojokerto di dapat dari hasil observasi, penulis melakukan peninjauan atau pengamatan langsung terhadap sampel objek penelitian, yaitu mencari tambal ban dan bengkel yang berada di kota Mojokerto.

Pada data lokasi tambal ban dan bengkel di area Kota Mojokerto akan ditunjukkan pada **gambar 2.2**.

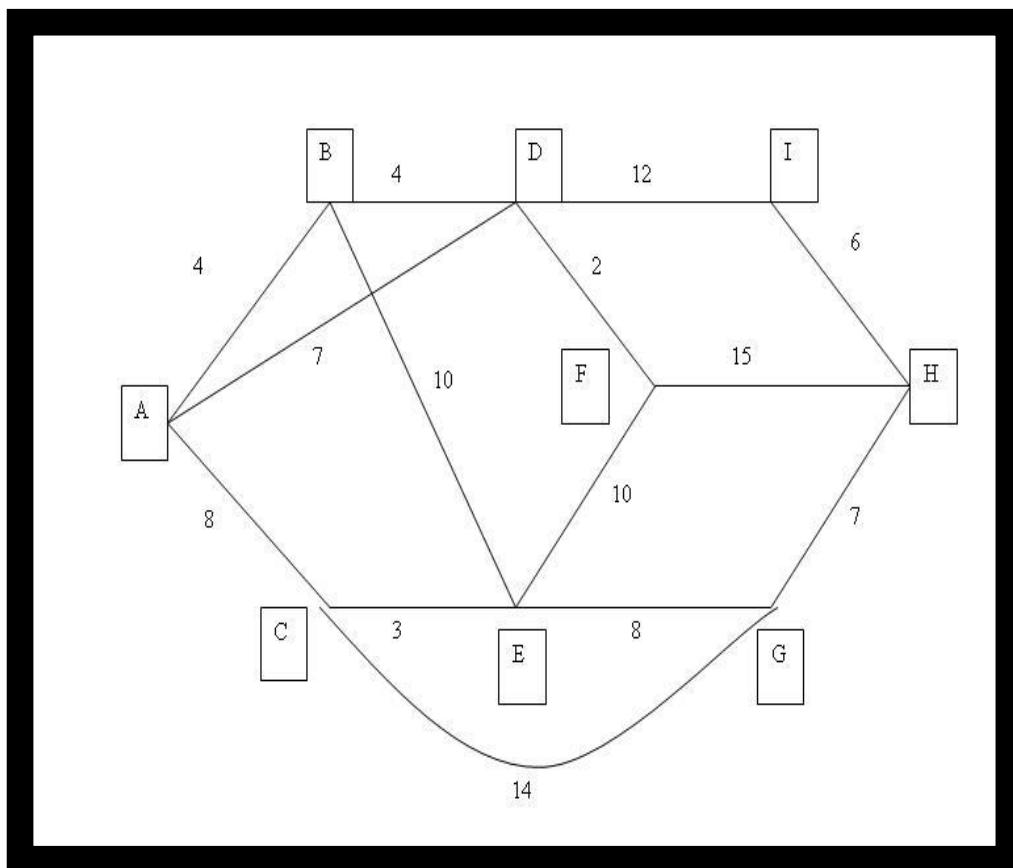


**Gambar 2. 2** Lokasi Tambal Ban dan Bengkel.

Tanda biru sebagai bengkel resmi Yamaha, tanda merah sebagai bengkel resmi Honda, tanda kuning sebagai bengkel resmi Suzuki, tanda hijau sebagai bengkel resmi Kawasaki, tanda ungu sebagai bengkel umum, tanda hitam sebagai tambal ban.

### 2.3 Metode Dijkstra

Algoritma yang ditemukan oleh *Dijkstra* untuk mencari path terpendek merupakan algoritma yang lebih efisien dibandingkan algoritma *Warshall*, meskipun implementasinya juga lebih sukar. Misalkan  $G$  adalah graf berarahberlabel dengan titik-titik  $V(G) = \{v_1, v_2, \dots, v_n\}$  dan path terpendek yang dicari adalah dari  $v_1$  ke  $v_n$ . Algoritma *Dijkstra* dimulai dari titik  $v_1$ . dalam iterasinya, algoritma akan mencari satu titik yang jumlah bobotnya dari titik 1 terkecil. Titik-titik yang terpiih dipisahkan dan titik-titik tersebut tidak diperhatikan lagi dalam iterasi berikutnya. Penjelasan metode *dijkstra* akan di tunjukkan pada **gambar 3.2.** [3]



**Gambar 2. 3** Penerapan metode *dijkstra*.

Mencari jalur teroptimal dari node A menuju node H

Iterasi I

Di mulai dari node A

$$d(B) = 4$$

$$d(C) = 8$$

$$d(D) = 7$$

$$d(E) = \infty$$

$$d(F) = \infty$$

$$d(G) = \infty$$

$$d(H) = \infty$$

$$d(I) = \infty$$

Dari node A menuju node B berbobot 4, menuju node C berbobot 8, menuju node D berbobot 7. Sedangkan rute dari node A menuju node E, F, G, H dan I berbobot tidak hingga karena dari node A tidak ada jalur untuk menuju node-node tersebut. Dari hasil bobot node diatas di ambil jalur ke node B karena bobotnya paling kecil dari semuanya. Jadi rute yang terbentuk adalah dari node A menuju node B, karena bobot rute tersebut paling kecil dari semuanya.

## Iterasi II

Di mulai dari node B

$$d(C) = \min(d(C), d(B)+d(C)) = \min(8, 4+\infty) = 8$$

$$d(D) = \min(d(D), d(B)+d(D)) = \min(7, 4+8) = 7$$

$$d(E) = \min(d(E), d(B)+d(E)) = \min(\infty, 4+10) = 14$$

$$d(F) = \min(d(F), d(B)+d(F)) = \min(\infty, 4+\infty) = \infty$$

$$d(G) = \min(d(G), d(B)+d(G)) = \min(\infty, 4+\infty) = \infty$$

$$d(H) = \min(d(H), d(B)+d(H)) = \min(\infty, 4+\infty) = \infty$$

$$d(I) = \min(d(I), d(B)+d(I)) = \min(\infty, 4+\infty) = \infty$$

Mengapa tak ada jalur dari node B menuju node A? Karena node A sudah terlewati atau dengan kata lain sudah terbentuk jalur dari node A ke node B, maka tidak perlu di catat lagi. Dari hasil bobot node diatas di ambil jalur

ke node D karena bobotnya paling kecil dari semuanya. Penjelasan rumus untuk mencari jarak minimum antar node sebagai berikut bobot jarak node yang dituju = minimum dari bobot jarak node sebelumnya dibandingkan dengan jumlah bobot dari node awal sekarang dengan jarak ke node yang akan dituju. Jadi rute yang terbentuk adalah dari node A menuju node D, karena bobot rute tersebut paling kecil dari semuanya. Mengapa rutanya berubah menjadi node A ke node D? Bukan ke node B ke node D? Karena pada perhitungan rumus di node D nilai bobot yang dihasilkan berasal dari perhitungan di iterasi I atau node A.  $d(D) = \min(d(D), d(B)+d(D)) = \min(7, 4+8) = 7$  (nilai 7 di ambil dari perhitungan di node sebelumnya yaitu node A atau iterasi I).

### Iterasi III

Di mulai dari node D

$$d(C) = \min(d(C), d(D)+d(C)) = \min(8, 7+\infty) = 8$$

$$d(E) = \min(d(E), d(D)+d(E)) = \min(14, 7+\infty) = 14$$

$$d(F) = \min(d(F), d(D)+d(F)) = \min(\infty, 7+2) = 9$$

$$d(G) = \min(d(G), d(D)+d(G)) = \min(\infty, 7+\infty) = \infty$$

$$d(H) = \min(d(H), d(D)+d(H)) = \min(\infty, 7+\infty) = \infty$$

$$d(I) = \min(d(I), d(D)+d(I)) = \min(\infty, 7+12) = 19$$

Dari hasil bobot node diatas di ambil jalur ke node C karena bobotnya paling kecil dari semuanya. Jadi rute yang terbentuk adalah dari node A menuju node C, karena bobot rute tersebut paling kecil dari semuanya. Sama halnya pada iterasi II nilai dari node C diambil dari node sebelumnya yaitu perhitungan di node A atau iterasi I.

### Iterasi IV

Di mulai dari node C

$$d(E) = \min(d(E), d(C)+d(E)) = \min(14, 8+3) = 11$$

$$d(F) = \min(d(F), d(C)+d(F)) = \min(9, 8+\infty) = 9$$

$$d(G) = \min(d(G), d(C)+d(G)) = \min(\infty, 8+14) = 22$$

$$d(H) = \min(d(H), d(C)+d(H)) = \min(\infty, 8+\infty) = \infty$$

$$d(I) = \min(d(I), d(C)+d(I)) = \min(19, 8+\infty) = 19$$

Dari hasil bobot node diatas di ambil jalur ke node F karena bobotnya paling kecil dari semuanya. Jadi rute yang terbentuk adalah dari node D menuju node F, karena bobot rute tersebut paling kecil dari semuanya dan bobot atau nilai rute ke node F di ambil dari perhitungan di node D atau iterasi III sehingga rute yang terbentuk dari node D ke node F.

#### Iterasi V

Di mulai dari node F

$$d(E) = \min(d(E), d(F)+d(E)) = \min(11, 9+8) = 11$$

$$d(G) = \min(d(G), d(F)+d(G)) = \min(22, 9+\infty) = 22$$

$$d(H) = \min(d(H), d(F)+d(H)) = \min(\infty, 9+15) = 24$$

$$d(I) = \min(d(I), d(F)+d(I)) = \min(19, 9+\infty) = 19$$

Dari hasil bobot node diatas di ambil jalur ke node E karena bobotnya paling kecil dari semuanya. Jadi rute yang terbentuk adalah dari node C menuju node E, karena bobot rute tersebut paling kecil dari semuanya dan bobot atau nilai rute ke node E di ambil dari perhitungan di node C atau iterasi IV sehingga rute yang terbentuk dari node C ke node E.

#### Iterasi VI

Di mulai dari node E

$$d(G) = \min(d(G), d(E)+d(G)) = \min(22, 11+8) = 19$$

$$d(H) = \min(d(H), d(E)+d(H)) = \min(\infty, 11+15) = 24$$

$$d(I) = \min(d(I), d(E)+d(I)) = \min(19, 11+\infty) = 19$$

Dari hasil bobot node diatas di ambil jalur ke node E karena bobotnya paling kecil dari semuanya. Jadi rute yang terbentuk adalah dari node E menuju node G, karena bobot rute tersebut paling kecil dari semuanya dan bobot atau nilai rute ke node G di ambil dari perhitungan di node E atau iterasi VI sehingga rute yang terbentuk dari node E ke node G.

### Iterasi VII

Di mulai dari node G

$$d(H) = \min(d(H), d(G)+d(H)) = \min(24, 19+7) = 24$$

$$d(I) = \min(d(I), d(G)+d(I)) = \min(19, 19+\infty) = 19$$

Dari hasil bobot node diatas di ambil jalur ke node I karena bobotnya paling kecil dari semuanya. Jadi rute yang terbentuk adalah dari node D menuju node I, karena bobot rute tersebut paling kecil dari semuanya dan bobot atau nilai rute ke node I di ambil dari perhitungan di node D atau iterasi III sehingga rute yang terbentuk dari node D ke node I.

### Iterasi VIII

Di mulai dari node I

$$d(H) = \min(d(H), d(I)+d(H)) = \min(24, 19+6) = 24$$

Jadi rute yang terbentuk adalah dari node F menuju node H, karena bobot rute tersebut paling kecil dari semuanya dan bobot atau nilai rute ke node H di ambil dari perhitungan di node F atau iterasi V sehingga rute yang terbentuk dari node F ke node H. Karena sudah mencapai node yang dituju yaitu node H maka mulailah proses pencarian jarak yang optimal dengan mengkalkulasi jarak-jarak yang sudah ditemukan. Jadi rute yang optimal dari node A ke node H adalah melewati node A menuju node D, node D menuju node F, node F menuju node H dengan nilai total atau bobot total adalah 24.

## 2.4 Sistem Informasi

Sistem informasi dapat didefinisikan sebagai suatu sistem di dalam suatu organisasi yang merupakan kombinasi dari orang-orang, fasilitas, teknologi, media, prosedur-prosedur dan pengendalian yang ditujukan untuk mendapatkan jalur komunikasi penting, memproses tipe transaksi rutin tertentu, memberi sinyal kepada manajemen dan yang lainnya terhadap kejadian-kejadian internal dan eksternal yang penting dan menyediakan suatu dasar informasi untuk pengambilan keputusan yang cerdik.[5]

## 2.5 Smartphone

*Smartphone* adalah perangkat telekomunikasi serbaguna. *Smartphone* atau yang disebut juga piranti pintar dewasa ini menjadi *trend* yang mewabah di Indonesia, digemarinya *smartphone* ini juga bukan tanpa alasan, tapi karena fitur yang ditawarkan sangat menarik dan mengubah hobi pengguna untuk *browsing*, *chatting* dan semacamnya yang awalnya dilakukan secara statis, sekarang dengan *smartphone* bisa dilakukan dengan *mobile* atau bergerak.

Seperti halnya pengguna atau user yang semakin cerdas memilih perangkat pintarnya, vendor pun bersaing menawarkan fitur dan *interface* yang menarik dan mudah untuk penggunanya, mereka menawarkan *OS (operating system)* dan aplikasi yang memanjakan penggunanya, *RIM* menawarkan *OS Blackberry*, Apple dengan Iphone nya, Samsung, Sony, Huawei, dan beberapa vendor lainnya mengadopsi *Android* sebagai *OS* nya, ada juga yang memakai jasa *Windows phone* untuk *OS*, seperti yang dilakukan oleh Nokia.[6]

## 2.6 Web Service

*Web service* adalah sistem perangkat lunak yang dirancang untuk mendukung interaksi yang bisa beroperasi *machine-to-machine* di atas jaringan. *Web service* mempunyai alat penghubung yang diuraikan di dalam format *machine - processable* (secara spesifik WSDL). Sistem lain saling berhubungan dengan *Web service* di dalam cara yang ditentukan oleh deskripsinya yang menggunakan pesan SOAP.

Keuntungan *web service* :

- *Web Service* mempunyai sifat *interoperability* sehingga bisa diakses oleh aplikasi yang berjalan pada *platform* yang berbeda-beda.
- *Web Service* menggunakan standar dan *protocol* terbuka pada Internet.
- Dengan menggunakan *HTTP* atau *SMTP*, *Web Service* bisa menembus pengamanan *firewall* suatu organisasi tanpa mengubah konfigurasi *firewall*.
- *Web Service* memungkinkan fungsi-fungsi pada banyak perangkat lunak di internet untuk dipadukan menjadi satu *Web Service* baru.
- *Web Service* memungkinkan penggunaan ulang layanan dan komponen. [6]

## 2.7 Sistem Informasi Geografis

Sistem Informasi Geografis (*SIG*) merupakan suatu sistem informasi berbasis komputer yang digunakan untuk menyajikan secara digital dan menganalisa penampakan geografis yang ada di permukaan bumi. Penyajian secara digital berarti mengubah keadaan menjadi bentuk digital. Setiap objek yang ada di permukaan bumi merupakan “*geo-referenced*”, yang merupakan kerangka hubungan *database* ke *SIG*. “*Geo-referenced*” menunjukkan lokasi suatu objek di ruang yang ditentukan oleh sistem koordinat, sedangkan *database* yaitu sekumpulan informasi tentang sesuatu dan hubungannya antar satu dengan lainnya.[4]

Dengan demikian sistem informasi geografis merupakan sebuah sistem yang dapat membantu tugas manusia untuk memecahkan masalah dan dapat dijadikan sebagai alat untuk menunjang aktifitas manusia. Maka, untuk membangun aplikasi diperlukan komponen-komponen dasar seperti:

1. Antar muka (*user Interface*)

Fasilitas yang menghubungkan antara sistem dengan pengguna.

2. *Location Based Service* (LBS)

Teknologi *Location Based service* (LBS) merupakan salah satu bagian dari implementasi *mobile GIS* yang lebih cenderung memberikan

fungsi terapan sehari-hari seperti menampilkan direktori kota, navigasi kendaraan, pencarian alamat serta jejaring sosial dibanding fungsionalitas pada teknologi *GIS* populer untuk *Field Based GIS*.

*Location Manager (API Maps)*: Menyediakan *tools/source* untuk *LBS*, *Aplication Programming Interface (API) Maps* menyediakan fasilitas untuk menampilkan, memanipulasi peta beserta feature lainnya seperti tampilan satelit, *street* (jalan), maupun gabungannya.

### 3. *JavaScript*

*JavaScript* adalah bahasa pemrograman *web* yang bersifat *Client Side Programming Language*. *Client Side Programming Language* adalah tipe bahasa pemrograman yang pemrosesannya dilakukan oleh *client*.

## **BAB III. METODOLOGI PENELITIAN**

### **3.1 Metodologi Penelitian**

Pada bab ini akan dijelaskan mengenai urutan dalam pembuatan Sistem Informasi Geografis penentuan rute terpendek menuju tambal ban dan bengkel menggunakan metode *Dijkstra* berdasarkan titik yang ditentukan oleh *user*. Sistem ini bekerja dengan menganalisa data berupa titik yang diinputkan oleh *user* yang selanjutnya akan diproses dan dianalisa menggunakan perhitungan dari metode *Dijkstra*. *Output* yang dihasilkan berupa rute terpendek menuju tambal ban dan bengkel. Metodologi yang akan digunakan dalam pengembangan sistem ini yaitu :

#### **3.1.1 Pengumpulan Data**

Pada tahapan ini penelitian dilakukan dengan cara mempelajari berbagai literatur melalui pengumpulan dokumen-dokumen, referensi-referensi, buku-buku, sumber dari internet, melakukan observasi maupun sumber lain yang mendukung dan diperlukan untuk merancang sistem yang berkaitan dengan penulisan skripsi yang dilakukan. observasi dapat dilakukan pada tahapan ini dengan tujuan mendapatkan informasi dan data secara *valid* sehingga data tersebut nantinya diolah dan digunakan dalam pengujian sistem.

##### **1. Studi Pustaka**

Pada tahapan pengumpulan data dengan studi pustaka, penulis membaca dan mempelajari referensi-referensi yang relevan berupa makalah, skripsi, dan buku. Fasilitas internet juga digunakan sebagai media untuk mencari data atau informasi yang dipublikasikan di dunia maya yang berkaitan dengan objek penelitian. Teori yang berhubungan dengan penelitian ini antara lain konsep aplikasi berbasis *Webservice*, *GPS*, *Google Maps*, *DBMS MySQL*, dan *PHP*.

## 2. Observasi

Pada tahapan pengumpulan data dengan observasi, penulis melakukan peninjauan atau pengamatan langsung terhadap sampel objek penelitian, yaitu mencari tambal ban dan bengkel yang berada di kota Mojokerto.

### 3.1.2 Analisa Kebutuhan

Analisa kebutuhan sistem yang akan dirancang disesuaikan dengan analisa kebutuhan *user*, meliputi :

a. Masukan (*Input*)

Data yang digunakan untuk *inputan* berupa titik yang di tentukan oleh *user*, dan data yang di perlukan berupa peta kota Mojokerto yang telah di beri tanda letak tambal ban dan bengkel, data ini didapat dari observasi penulis melakukan peninjauan atau pengamatan langsung terhadap sampel objek penelitian, yaitu mencari tambal ban dan bengkel yang berada di kota Mojokerto. *User* menginputkan titik yang di tentukan oleh *user* dan ditampilkan pada user *interface*.

b. Proses

Dalam memproses data inputan menjadi output, yang diperlukan yaitu data peta kota Mojokerto yang telah di beri tanda letak tambal ban dan bengkel lalu *user* memberikan titik poin di mana *user* berada. Selanjutnya data *inputan* tersebut akan diproses, dianalisa dan dihitung dengan menggunakan metode *Dijkstra* yang nantinya akan diperoleh rute terpendek menuju tambal ban dan bengkel.

c. Keluaran (*Output*)

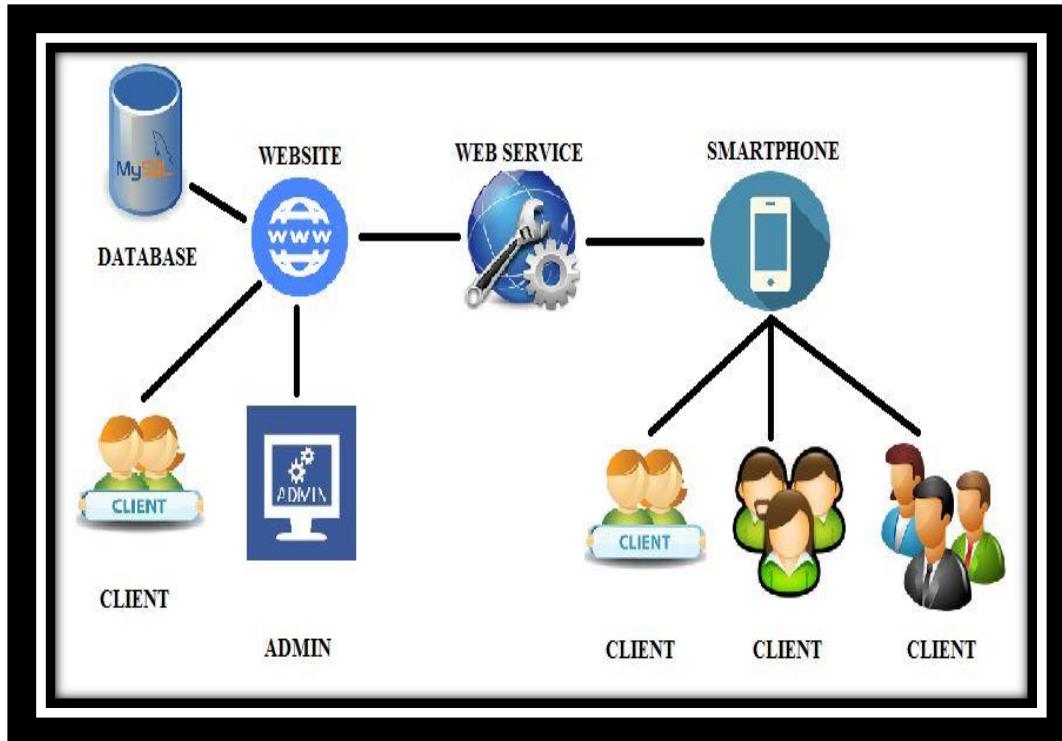
Hasil *output* dari sistem ini berupa rute terpendek menuju tambal ban dan bengkel.

### 3.1.3 Perancangan Sistem

Desain dan perancangan sistem merupakan bagian dari kebutuhan-kebutuhan fungsional dan sebagai persiapan untuk implementasi sistem. Langkah dari desain pembuatan sistem meliputi struktur data berupa *database* yang dipelukan dan

digunakan oleh sistem, desain *interface*, dan prosedur pengodean terhadap sistem yang akan dibuat.

Pada perancangan sistem akan di tunjukkan pada **gambar 3.1.**



**Gambar 3. 1** Perancangan Sistem.

### 3.1.4 Implementasi Sistem

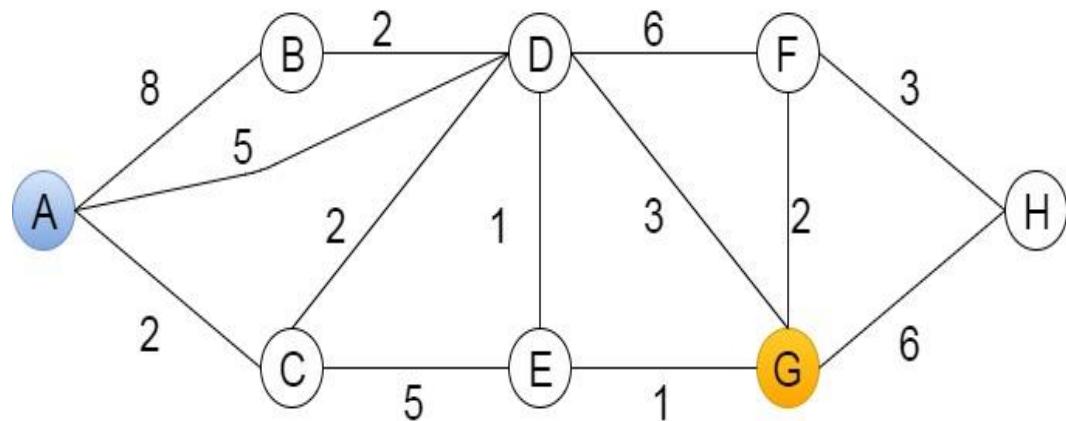
Implementasi sistem dilakukan berdasarkan perancangan aplikasi. Proses implementasi dilakukan dengan mengimplementasikan analisa kebutuhan perangkat sebagai penunjang diantaranya pemrograman *PHP*, *basisdata MySQL*, serta *tools* yang akan digunakan dalam pembuatan sistem, data peta kota Mojokerto yang telah di beri tanda letak tambal ban dan bengkel, Berdasarkan dari analisa kebutuhan tersebut, proses perancangan yang dapat dihasilkan berupa :

- Implementasi sistem berupa tampilan dengan mengimplementasikan *Location Manager*.
- Data peta kota Mojokerto yang telah di beri tanda letak tambal ban dan bengkel dimasukkan kedalam *database* dan diimplementasikan ke dalam aplikasi, sehingga data yang telah tersimpan dalam *database* tersebut dapat

dolah oleh sistem.

- c. Implementasi sistem berbasis *webservice* dalam menjalankan aplikasi akan terdapat *server* sebagai sumber informasi dan *client* akan mendapatkan informasi dari *server*.
- d. Implementasi metode *dijkstra* dalam melakukan proses analisa dan penghitungan data berdasarkan titik yang telah dimasukkan *user*, sehingga dari metode tadi diperoleh hasil rute terpendek menuju tambal ban dan bengkel.

Pada **gambar 3.2** di bawah ini adalah Jalur perhitungan *dijkstra*.



**Gambar 3. 2** Jalur perhitungan *dijkstra*.

Pada **gambar 3.3** di bawah ini adalah Jalur perhitungan *dijkstra*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1																											
2																											
3	Jumlah Kota	8																									
4	A,B,C,D,E,F,G,H																										
5	Start Kota	A																									
6	Stop Kota	G																									
7																											

**Algoritma Dijkstra untuk Mencari Jarak Jalur Terpendek**

untuk setiap langkah, (1) dari kota terpilih, jumlahkan angka terkecil sebelumnya dengan semua jarak dari kota terpilih ke kota terhubung yang kotaknya tidak berwarna biru,  
(2) bandingkan hasil penjumlahan pada langkah 1 dengan nilai pada kotak sebelumnya, tuliskan ke dalam kotak nilai terkecil dari kedua nilai tersebut  
(3) dari kotak yang masih berwarna putih, pilih angka terkecil maka pilihan kota berikutnya adalah kota dengan nilai terkecil tersebut

**Gambar 3. 3** Perhitungan *dijkstra*.

Pada gambar 3.4 di bawah ini adalah Jalur perhitungan *dijkstra*.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
7									0	1	2	3	4	5	6	7	8	9	
8	dari ↔ ke	jarak	cek						Langkah	Pilih	Terkecil	A	B	C	D	E	F	G	H
9	A	B	8																
10	A	C	2						1	A	2	0	8	2	5	~	~	~	
11	A	D	5							C	A	A	A	A	A	A	A	A	
12	B	D	2																
13	C	D	2																
14	C	E	5						2	C	4	0	8	2	4	7	~	~	
15	D	E	1							D	A	A	A	C	C	A	A	A	
16	D	F	6																
17	D	G	3																
18	E	G	1						3	D	5	0	6	2	4	5	10	7	~
19	F	G	2							E	A	D	A	C	D	D	D	A	
20	F	H	3																
21	G	H	6																
22									4	E	6	0	6	2	4	5	10	6	~
23										B	A	D	A	C	D	D	E	A	
24																			
25									5										
26																			
27																			
28																			

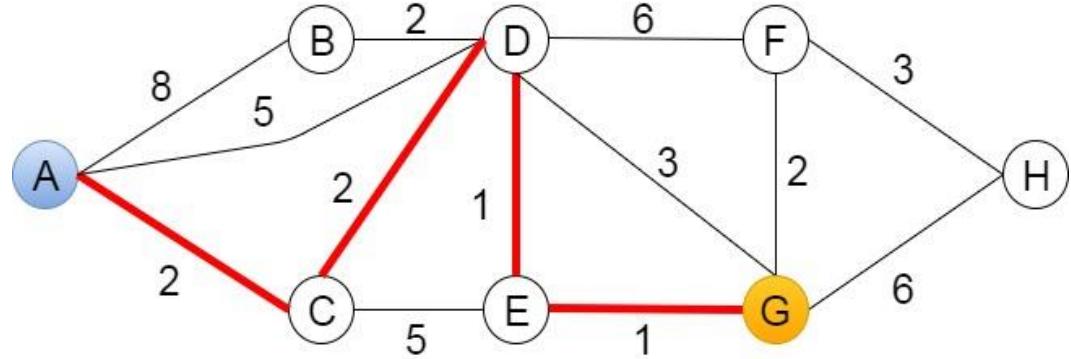
Gambar 3.4 Perhitungan *dijkstra*.

Pada gambar 3.5 di bawah ini adalah Jalur perhitungan *dijkstra*.

29																			
30									6	G	6	0	6	2	4	5	8	6	12
31										stop	A	D	A	C	D	G	E	G	
32																			
33									7		6	0	6	2	4	5	8	6	12
34											A	D	A	C	D	G	E	G	
35																			
36									8		6	0	6	2	4	5	8	6	12
37											A	D	A	C	D	G	E	G	
38																			
39																			
40																			

Gambar 3.5 Perhitungan *dijkstra*.

Pada **gambar 3.6** di bawah ini adalah jalur yang di lewati.



**Gambar 3.6** Jalur yang di lewati.

Berdasarkan hasil dari perhitungan yang telah di lakukan pada **gambar 3.2**, **gambar 3.3**, **gambar 3.4**, **gambar 3.5**, dan **gambar 3.6** maka di dapat jalur terpendek yaitu pada jalur A – C – D – E – G dengan nilai 6.

### 3.1.5 Pengujian Sistem

Pengujian sistem dilakukan dengan tujuan, sistem yang telah di rancang dapat berfungsi. Proses pengujian sistem dilakukan dengan:

1. Sistem akan menampilkan peta Kota Mojokerto sebagai sumber pengetahuan yang berisi data letak tambal ban dan bengkel.
2. Pengujian oleh *user*
  - a) Sistem dapat menampilkan peta Kota Mojokerto yang telah di beri tanda letak tambal ban dan bengkel yang akan dipilih oleh *user* pada halaman utama.
  - b) Sistem dapat menampilkan rute terpendek menuju tambal ban dan bengkel berdasarkan titik yang telah *diinputkan* oleh *user*.
3. Pengujian dilakukan oleh *admin* meliputi :
  - a) Sistem dapat melakukan proses *login*.
  - b) Sistem dapat melakukan penambahan data tambal ban.
  - c) Sistem dapat melakukan penambahan data bengkel.
  - d) Sistem dapat menemukan rute terpendek menuju tambal ban dan bengkel.

## BAB IV. ANALISIS DAN PERANCANGAN

Bab ini akan membahas analisis kebutuhan dan perancangan sistem Aplikasi Pencarian Jalur Terpendek Menuju Lokasi Bengkel dan Tambal Ban di Kota Mojokerto Berbasis *Webservice*.

### 4.1 Analisis Kebutuhan

Dibagian bawah ini adalah pembahasan untuk kebutuhan perangkat keras dan lunak yang dibutuhkan dalam pembahasan aplikasi dan perancangan desain sistem untuk pembuatan alur proses aplikasi.

#### 4.1.1 Deskripsi Umum

Sistem yang dibuat ini nantinya akan berguna pada *user* yang berada di area kota Mojokerto. Tujuannya adalah mempermudah *user* dalam mencari tambal ban dan bengkel terdekat serta dapat mempermudah *user* dalam mencari rute terdekat menuju tambal ban dan bengkel di area Kota Mojokerto.

Berdasarkan tugas dan fungsinya, pengguna sistem ini dibagi menjadi 2 yaitu:

##### a. Petugas Administrasi

- Mengelola data berupa data jalan, meliputi penambahan data, perubahan data dan juga menghapus data yang berada pada menu *admin*.
- Mengelola data berupa data lokasi tambal ban, meliputi penambahan data, perubahan data dan juga menghapus data yang berada pada menu *admin*.
- Mengelola data berupa data lokasi bengkel, meliputi penambahan data, perubahan data dan juga menghapus data yang berada pada menu *admin*.

##### b. *User*

- *User* dapat mengetahui rute terdekat menuju tambal ban dan bengkel di area kota mojokerto.
- *User* dapat mengetahui lokasi tambal ban dan bengkel di area kota mojokerto.
- *User* dapat mengetahui jarak menuju tambal ban dan bengkel di area kota.

#### 4.1.2 Kebutuhan Perangkat Keras (*Hardware*)

Kebutuhan perangkat keras (*Hardware*) yang mampu mendukung pengoperasian program, memenuhi spesifikasi minimal dari kebutuhan *hardware* dari sistem yang akan diterapkan. Seperti dibawah ini :

- *Processor Intel Core i3 Processor 2350M*
- Memori RAM 2 GB.
- *Graphics ATI Radeon HD 6370M 1G.*
- *Harddisk 500 GB.*
- *Mouse dan Keyboard standar.*
- *Smartphone android minimal 4.4 Kitkat.*

#### 4.1.3 Kebutuhan Perangkat Lunak (*Software*)

Untuk menjalankan sistem informasi ini dibutuhkan sistem perangkat lunak yang mampu mendukung pengoperasian program, perangkat lunak ini dibutuhkan agar sistem bisa berjalan dengan baik yang mampu mendukung pembuatan dan pengoperasian program Seperti dibawah ini:

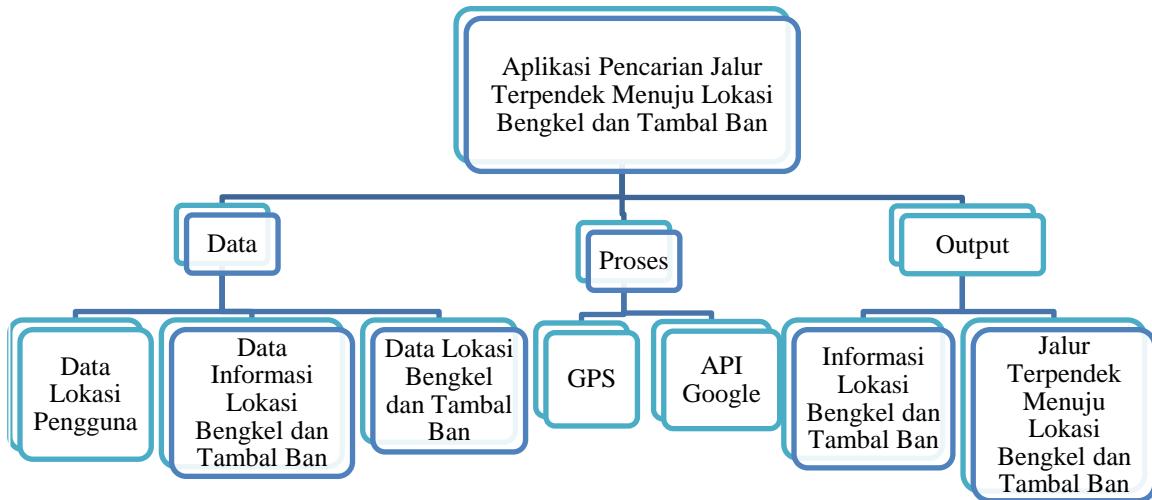
- Sistem Operasi Windows 7/10.
- *PHP sebagai bahasa pemrograman server side.*
- *Database Mysql.*
- *Web Server menggunakan Apache.*
- *Android Studio.*
- *Android SDK.*
- *Java Development Kit (JDK).*

### 4.2 Perancangan Sistem

Perancangan sistem dapat menggunakan berbagai model, model yang digunakan untuk menggambarkan alur proses Sistem Informasi Geografis Pencarian Jalur Terpendek Menuju Lokasi Bengkel dan Tambal Ban di Kota Mojokerto Berbasis *Webservice* sebagai berikut :

#### 4.2.1 Work Breakdown Structure (WBS)

Penjelasan pada **Gambar 4.1** dari WBS adalah sebagai berikut :



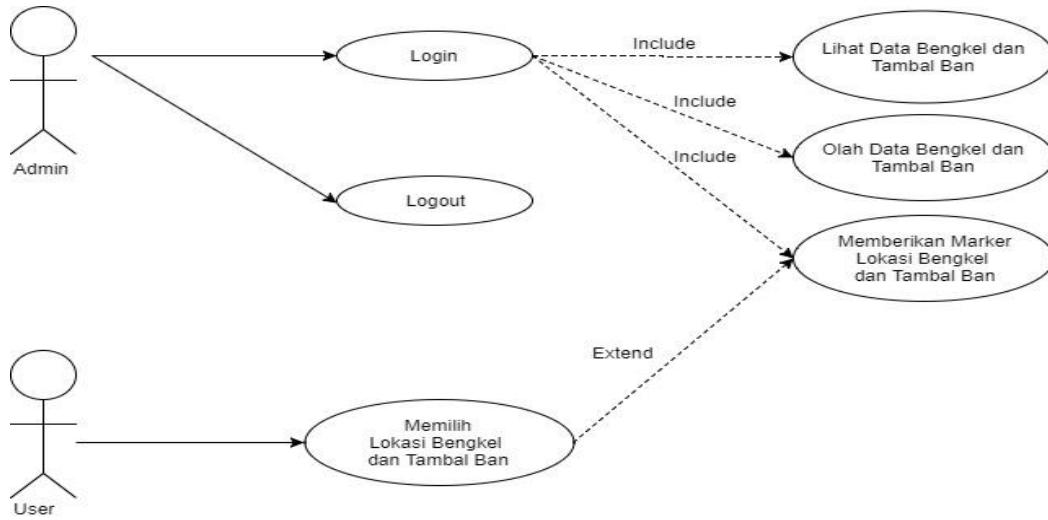
**Gambar 4. 1** Work Breakdown Structure (WBS).

Data yang digunakan pada sistem pencarian jalur terpendek untuk menemukan lokasi bengkel dan tambal ban di Kota Mojokerto adalah data lokasi pengguna, data informasi bengkel dan tambal, dan data lokasi bengkel dan tambal. Layanan yang dibutuhkan agar sistem mampu berjalan dengan baik yaitu *GPS* dan *API Google*. Kemudian sistem akan menghasilkan lokasi bengkel dan tambal, jalur terpendek dan informasi bengkel dan tambal.

#### 4.2.2 Use Case

*Use case* diagram adalah diagram yang menyajikan interaksi antara *use case* dan aktor. Dimana aktor dapat berupa orang, peralatan atau sistem lain yang berinteraksi dengan sistem yang sedang dibangun. *Use case* menggambarkan fungsionalitas sistem atau persyaratan – persyaratan yang harus dipenuhi sistem dari pandangan pemakai. *Use case* pada sistem ditampilkan pada **gambar 4.2**.

Penjelasan pada **Gambar 4.1** dari WBS adalah sebagai berikut :



**Gambar 4. 2** Diagram *Use case*.

Deskripsi pendefinisian aktor pada sistem pencarian jalur terpendek menuju lokasi Bengkel dan Tambal Ban di kota Mojokerto berbasis *Webservice* yang di sajikan pada **tabel 4.1**.

**Tabel 4. 1** Tabel fungsi aktor *use case*.

No	Aktor	Kebutuhan
1	<i>Admin</i>	Orang yang bertugas mengolah data Bengkel dan Tambal Ban.
2	<i>User</i>	Pengguna yang melakukan pencarian Bengkel dan Tambal Ban berdasarkan jarak yang diinginkan.

Selanjutnya, pendefinisian *use case* dijelaskan secara detail dan terperinci sebagaimana yang disajikan pada **tabel 4.2** pada tabel tersebut menjelaskan fitur – fitur dan menu yang ada pada aplikasi yang dibangun.

**Tabel 4. 2** Definisi *use case*.

No	Nama <i>Use Case</i>	Deskripsi	Pengguna
1.	<i>Login</i>	Berisi form yang digunakan untuk masuk kedalam sistem.	<i>Admin</i>

2.	<i>Logout</i>	Berisi form yang digunakan untuk keluar dari sistem.	<i>Admin</i>
3.	Lihat Data Bengkel dan Tambal Ban	Berisi data – data bengkel dan tambal ban yang sudah tersimpan sebelumnya di basis data.	<i>Admin</i>
4.	Olah Data Bengkel dan Tambal Ban	Berisi data – data bengkel dan tambal ban yaitu data id, nama, alamat, jenis.	<i>Admin</i>
5.	Memberikan <i>marker</i> lokasi Bengkel dan Tambal Ban	Berisi data bengkel dan tambal ban yang sudah selesai diolah dan siap untuk ditampilkan ke <i>User</i> .	<i>Admin</i>
6.	Memilih Bengkel dan Tambal Ban sesuai kriteria	Berisi data bengkel dan tambal ban, pengguna bisa memilih sesuai jarak yang diinginkan.	<i>User</i>

Selanjutnya, untuk pendeskripsian setiap *use case* akan disajikan pada tabel-tabel berikut ini yang ditampilkan pada **Tabel 4.3, Tabel 4.4, Tabel 4.5, Tabel 4.6, Tabel 4.7, dan Tabel 4.8.**

**Tabel 4. 3** Deskripsi *use case* "Login".

Use Case Name: <i>Login</i>	ID: UC.01 Siklus1	<i>Importance Level:</i> <i>High</i>		
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i>			
<i>Stakeholder and Interest:</i> Admin masuk kedalam sistem untuk mengolah data yang ada.				
<i>Brief Description:</i> Menjelaskan proses inisialisasi pengguna.				
<i>Trigger:</i> Admin masuk kedalam sistem untuk mengelola data bengkel dan tambal ban yang ada. <i>Type:</i> internal				

<i>Relationship:</i> <i>Association: Admin</i> <i>Include:</i>	
<i>Extend:</i> <i>Generalization:</i>	
<i>Normal flow of event:</i> 1. Admin memasukkan <i>username</i> dan <i>password</i> . 2. Admin menekan tombol <i>enter</i> untuk <i>login</i> . 3. Sistem memeriksa kebenaran data yang diberikan. 4. Jika data yang dimasukkan sesuai, maka <i>login admin</i> diterima. Sistem menampilkan halaman yang sesuai dengan hak akses yang diberikan.	
<i>Subflows:</i> 1a. Jika <i>username</i> dan <i>password</i> yang diinputkan kosong, maka akan muncul alert “ <i>Gagal Login</i> ”. 1b. Jika <i>username</i> dan <i>password</i> yang diinputkan salah, maka akan muncul alert “ <i>Gagal Login</i> ”.	

**Tabel 4. 4** Deskripsi *use case* "Logout".

<i>Use Case Name:</i> <i>Logout</i>	<i>ID: UC.02</i> Siklus2	<i>Importance Level:</i> <i>High</i>
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Admin keluar dari sistem setelah selesai melakukan pengolahan data atau aktivitas yang diperlukan sudah selesai.		
<i>Brief Description:</i> Menjelaskan proses inisialisasi pengguna.		
<i>Trigger:</i> Admin masuk kedalam sistem untuk mengelola data bengkel dan tambalan yang ada. <i>Type:</i> internal		

<i>Relationship:</i> <i>Association: Admin</i> <i>Include: Login</i>
<i>Extend:</i> <i>Generalization:</i>
<i>Normal flow of event:</i> 1. Admin menekan tombol <i>Logout</i> untuk melakukan <i>logout</i> . 2. Sistem akan menampilkan <i>form Login</i> .
<i>Subflows:</i> 2a. Setelah menekan tombol atau <i>button Logout</i> , maka akan muncul halaman <i>Login</i> .

**Tabel 4. 5** Deskripsi *use case* "Lihat data bengkel dan tambal ban".

<i>Use Case Name:</i> Lihat data Fasilitas Kesehatan	<i>ID:</i> UC.03 Siklus3	<i>Importance Level:</i> <i>High</i>		
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i>			
<i>Stakeholder and Interest:</i> Untuk melihat data yang sudah tersimpan di basis data.				
<i>Brief Description:</i> Menjelaskan proses <i>admin</i> melihat data yang ada.				
<i>Trigger:</i> Admin melihat data yang ada. <i>Type:</i> internal				
<i>Relationship:</i> <i>Association: Admin</i> <i>Include: Login</i>				
<i>Extend:</i> <i>Generalization:</i>				

<i>Normal flow of event:</i>
1. Jika <i>user</i> menekan tombol <i>login</i> , maka otomatis akan tampil <i>form</i> yang berisi data2 yang ada di dalam basis data.
<i>Subflows:</i>

**Tabel 4. 6** Deskripsi *use case* "Olah data bengkel dan tambal ban".

<i>Use Case Name:</i> Olah data Fasilitas Kesehatan	<i>ID:</i> UC.04 Siklus4	<i>Importance Level:</i> <i>High</i>
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Admin memasukkan data dan mengolah data yang diperlukan dalam basis data.		
<i>Brief Description:</i> Menjelaskan proses <i>admin</i> mengelola data agar siap ditampilkan dalam sistem.		
<i>Trigger:</i> Admin mengelola data yang ada. <i>Type:</i> internal		
<i>Relationship:</i> <i>Association:</i> Admin <i>Include:</i> Login		
<i>Extend:</i> <i>Generalization:</i>		
<i>Normal flow of event:</i>	1. Jika <i>Admin</i> menekan tombol <i>login</i> , maka otomatis akan tampil <i>form</i> yang berisi data2 yang ada di dalam basis data dan <i>admin</i> bisa melakukan pengolahan data seperti CRUD.	

***Subflows:***

- 1a. *Admin* menekan tombol atau *button submit*, apabila data berhasil tersimpan di *database* dan jika data tidak berhasil disimpan di *database* maka akan tampil pesan “ data tidak berhasil di simpan”.
- 4b. Jika *Admin* di *form* edit menekan *button submit* setelah melakukan beberapa data yang di ubah, apabila data tersimpan ke *database* sukses jika data tidak tersimpan ke *database* maka akan tampil pesan “data gagal diubah”.
- 4c. Jika *Admin* menekan *button delete* data yang di inginkannya berhasil dihapus dari *database* jika tidak berhasil dihapus maka akan tampil pesan “data gagal dihapus”.

**Tabel 4. 7** Deskripsi *use case* "Marker lokasi bengkel dan tambal ban".

<i>Use Case Name:</i> Memberikan <i>marker</i> Lokasi Fasilitas Kesehatan	<i>ID:</i> UC.05 Siklus5	<i>Importance Level:</i> <i>High</i>
<i>Primary Actor:</i> Admin	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Untuk memberikan penanda semua lokasi bengkel dan tambal ban di peta.		
<i>Brief Description:</i> Menjelaskan proses <i>Admin</i> memberikan penanda lokasi semua bengkel dan tambal ban pada peta secara otomatis ketika ada data yang masuk di basis data.		
<i>Trigger:</i> Admin menampilkan lokasi semua bengkel dan tambal ban. <i>Type:</i> internal		
<i>Relationship:</i> <i>Association:</i> Admin <i>Include:</i> Login		
<i>Extend:</i> <i>Generalization:</i>		
<i>Normal flow of event :</i> 1. <i>Admin</i> menekan tombol atau <i>button simpan</i> , data yang dimasukkan secara otomatis akan tampil di peta.		

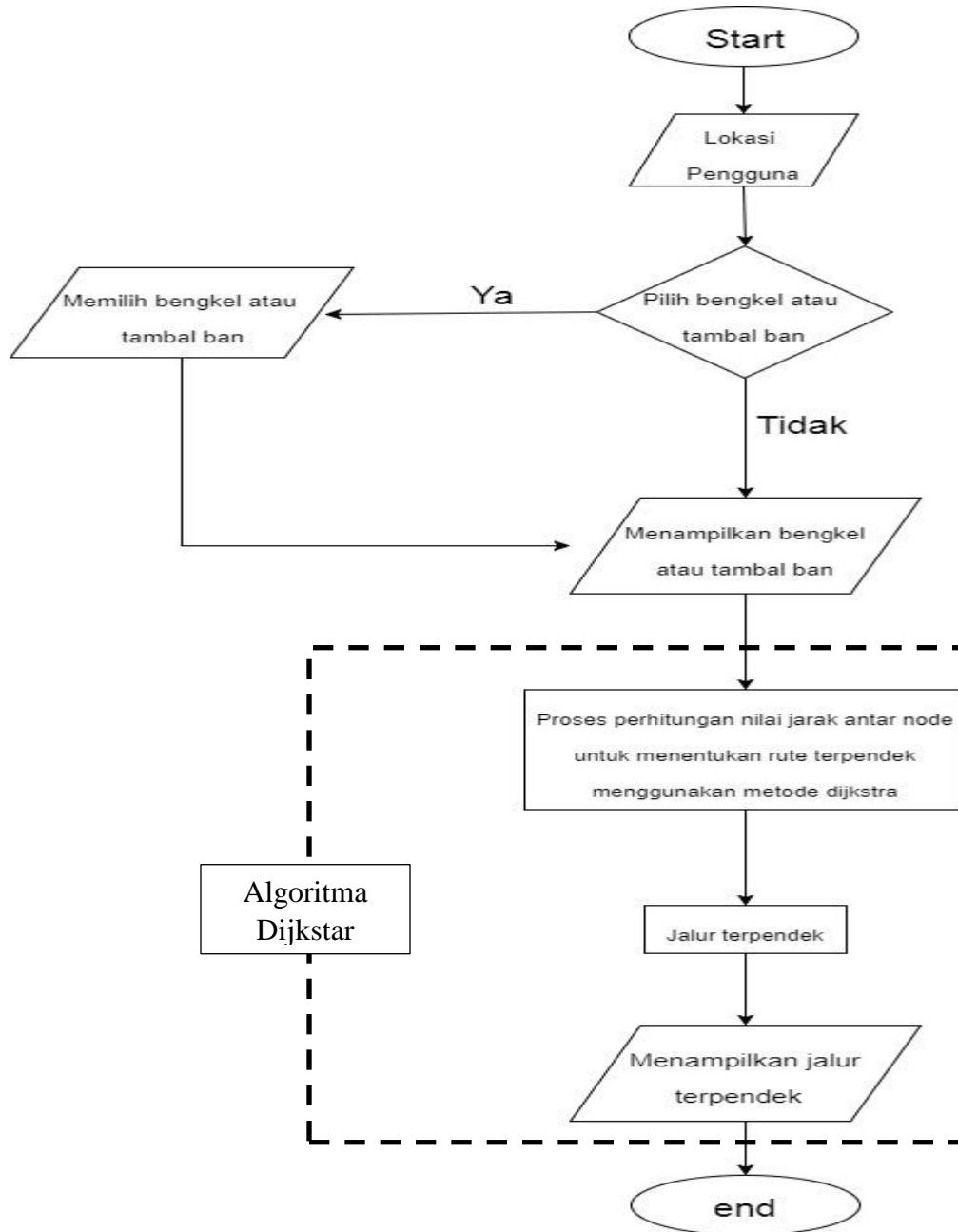
<i>Subflows:</i>
------------------

**Tabel 4. 8** Deskripsi *use case* memilih bengkel dan tambal ban sesuai jarak.

<i>Use Case Name:</i> Memilih Fasilitas Kesehatan sesuai kriteria	<i>ID:</i> UC.06 Siklus6	<i>Importance Level:</i> <i>High</i>
<i>Primary Actor:</i> Client	<i>Use Case Type:</i>	
<i>Stakeholder and Interest:</i> Pengguna memilih kriteria yang diinginkan dalam memilih lokasi bengkel dan tambal ban yang ingin ditampilkan.		
<i>Brief Description:</i> Menjelaskan proses pengguna memilih bengkel dan tambal ban yang diinginkan.		
<i>Trigger:</i> Pengguna memilih bengkel dan tambal ban. <i>Type:</i> internal		
<i>Extend:</i> <i>Generalization:</i>		
<i>Normal flow of event :</i> 1. Client menekan tombol Run untuk mengetahui rute dan jarak menuju lokasi bengkel dan tambal ban yang di inginkan.		
<i>Subflows:</i>		

#### 4.2.3 Flowchart

Flowchart pencarian lokasi bengkel dan tambal ban di jelaskan pada **gambar 4.3** berikut:

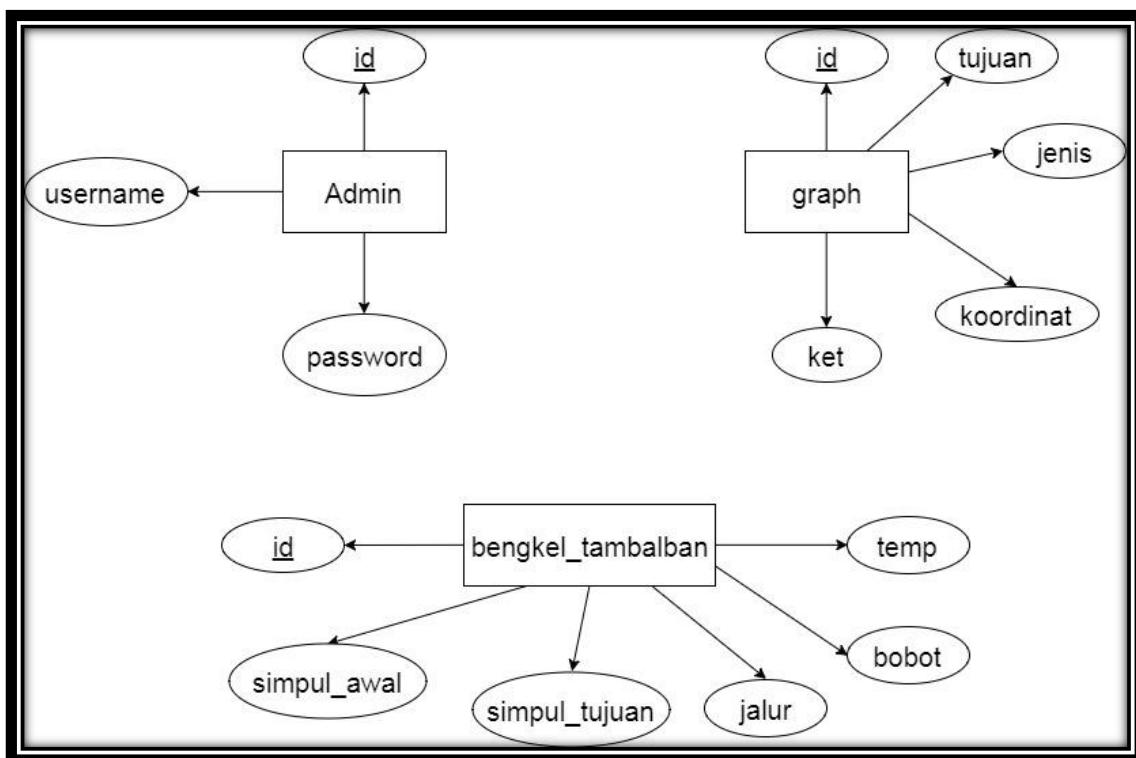


**Gambar 4. 3** Flowchart pencarian lokasi bengkel dan tambal ban.

Ketika membuka aplikasi, pertama kali sistem akan menampilkan peta kota mojokerto yang sudah di beri *marker* lokasi bengkel dan tambal ban. Selanjutnya pengguna harus memasukkan lokasi dimana dia berada, lalu pengguna harus memilih lokasi bengkel dan tambal ban terdekat. Jika pengguna tidak memilih lokasi bengkel dan tambal ban maka pilihan akan menjadi *default* yaitu lokasi bengkel dan tambal ban semua akan di tampilkan. Proses selanjutnya algoritma *dijkstra* berjalan pada sistem dan menampilkan jarak terpendek.

#### 4.2.4 Desain Database ERD (*Entity Relationship Diagram*)

Data disimpan dalam *database* dengan nama skripsi3. *database* yang digunakan terdapat tabel admin, tabel bengkel\_tambalban, dan tabel graph di dalam tabel admin terdapat *field* antara lain: id (*primary key*), username, dan password. Di tabel bengkel\_tambalban terdapat *field* antara lain: id (*primary key*), tujuan, jenis, koordinat dan ket. Di tabel graph terdapat *field* antara lain: id (*primary key*), simpul\_awal, simpul\_akhir, jalur, bobot, dan temp. tunjukkan pada **gambar 4.4**.



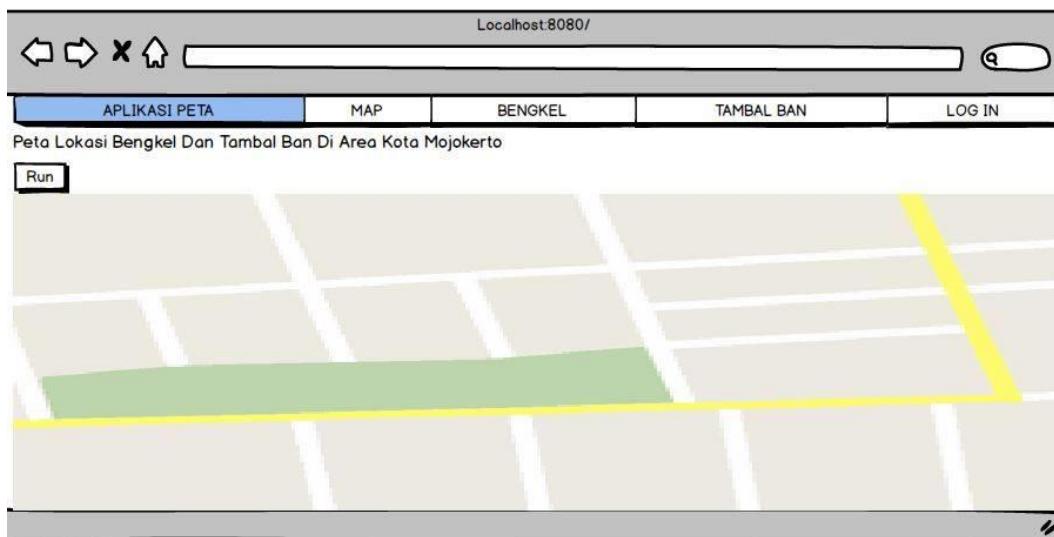
**Gambar 4. 4** Entity Relational Diagram.

#### 4.2.5 Desain Antarmuka (*Interface*)

Pada sistem informasi ini memiliki desain *interface* yang terdiri dari menu utama dan menu *login* untuk petugas administrasi (Halaman menu data bengkel dan tambal ban, menu menampilkan data *map*, menu menampilkan data lokasi *map*, menu tambah dan *edit* data, dan menu *logout*.

#### 4.2.6 *Interface* Halaman Menu Utama

*Form* utama digunakan sebagai pintu utama menuju sistem. Dimana *form* ini akan tampil pertama kali dalam menjalankan sistem ini. Desain *interface* halaman menu utama akan ditunjukkan pada **gambar 4.5** dan keterangan pada **tabel 4.9**.



**Gambar 4. 5** Desain *interface* halaman menu utama.

**Tabel 4. 9** Tabel keterangan menu utama.

	Keterangan
<i>Map</i>	Halaman ini adalah halaman utama yang dapat diakses oleh <i>user</i> dan <i>admin</i> .
<i>Login</i>	Halaman menuju <i>interface admin</i> selanjutnya.
<i>Button Bengkel</i>	<i>Button</i> ini berfungsi hanya untuk menampilkan <i>marker</i> lokasi bengkel.
<i>Button Tambal Ban</i>	<i>Button</i> ini berfungsi hanya untuk menampilkan <i>marker</i> lokasi tambal ban.

<i>Button RUN</i>	<i>Button ini berfungsi untuk menjalankan metode lalu menampilkan rute menuju lokasi bengkel dan tambal ban.</i>
-------------------	------------------------------------------------------------------------------------------------------------------

#### 4.2.7 *Interface* Halaman *login* Petugas Administrasi

Halaman login pada petugas administrasi terdapat *button* diantaranya *button sign in* dan *cancel* desain *interface* halaman *login* akan ditunjukkan pada **gambar 4.6** dan keterangan pada **tabel 4.10**.

The screenshot shows a web browser window with the URL "localhost:8080/" in the address bar. The main content is a "LOG IN" form. It includes two input fields for "Username" and "Password", a "Remember me" checkbox, a "Sign me in" button, and a link at the bottom labeled "Aplikasi Pemetaan Lokasi Bengkel Dan Tambal Ban". The browser interface shows standard controls like back, forward, and search.

**Gambar 4. 6** Desain *interface* halaman *login* petugas administrasi.

**Tabel 4. 10** Tabel keterangan menu *login*.

	Keterangan
<i>Sign In</i>	<i>Button ini akan menuju ke halaman utama admin jika memasukkan username dan password admin.</i>
Aplikasi Pemetaan Lokasi Bengkel Dan Tambal Ban	<i>Button ini akan menuju ke halaman utama jika memasukkan username dan password admin.</i>

#### 4.2.8 Interface Halaman Petugas Administrasi

Desain *interface* halaman utama akan ditunjukkan pada **gambar 4.7**, dan keterangan pada **tabel 4.11**.

Tabel Data Bengkel Dan Tambal Ban					
Id	Tujuan	Jenis	Koordinat	Edit	Delete

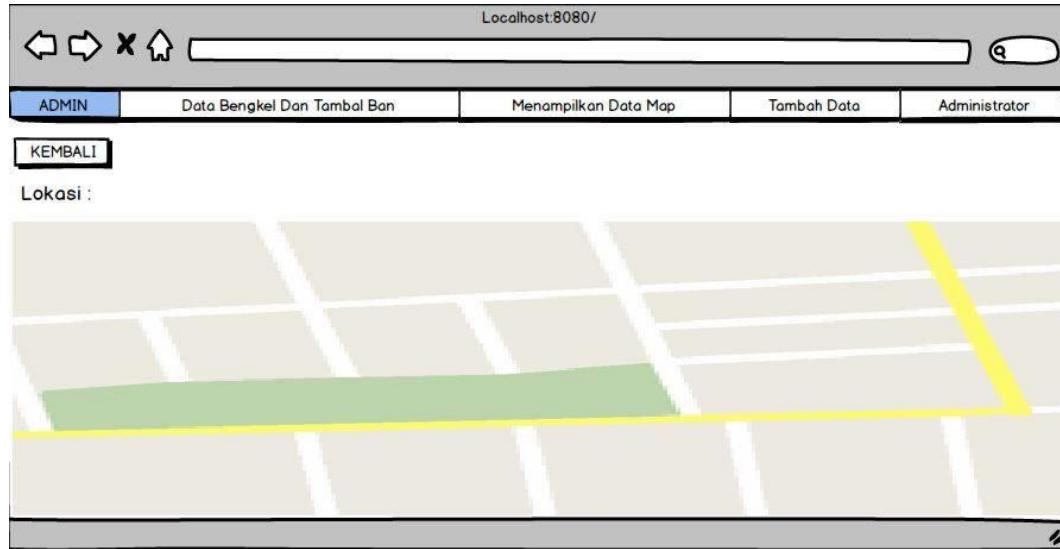
**Gambar 4. 7** Desain *interface* halaman utama petugas administrasi menu data bengkel dan tambal ban.

Desain *interface* halaman utama akan ditunjukkan pada **gambar 4.8**, dan keterangan pada **tabel 4.11**.

Tabel Data Map			
Id	Tujuan	Jenis	Lihat Map

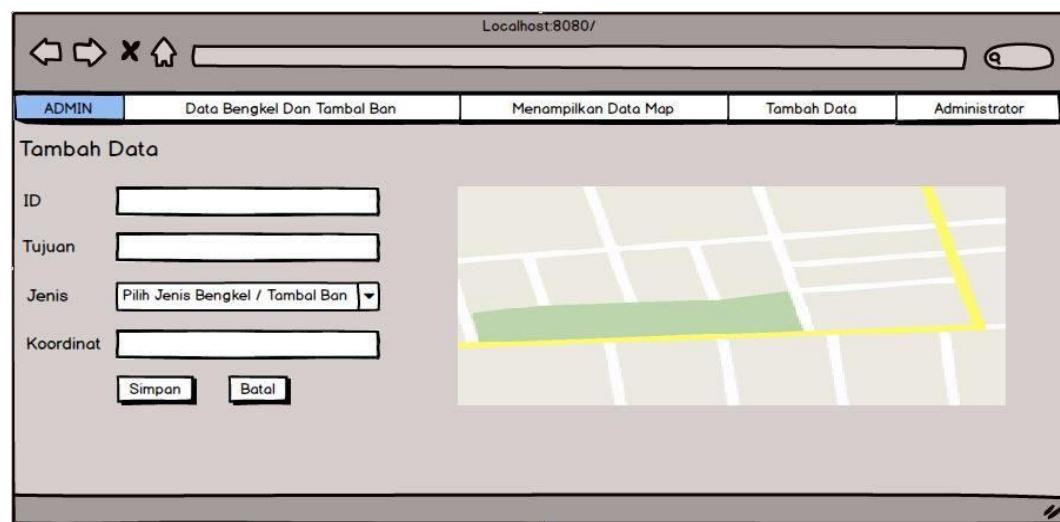
**Gambar 4. 8** Desain *interface* halaman utama petugas administrasi menu menampilkan data map.

Desain *interface* halaman utama akan ditunjukkan pada **gambar 4.9**, dan keterangan pada **tabel 4.11**.



**Gambar 4. 9** Desain *interface* halaman utama petugas administrasi menu menampilkan lokasi *map*.

Desain *interface* halaman utama akan ditunjukkan pada **gambar 4.10**, dan keterangan pada **tabel 4.11**.



**Gambar 4. 10** Desain *interface* halaman utama petugas administrasi menu tambah dan *edit* data.

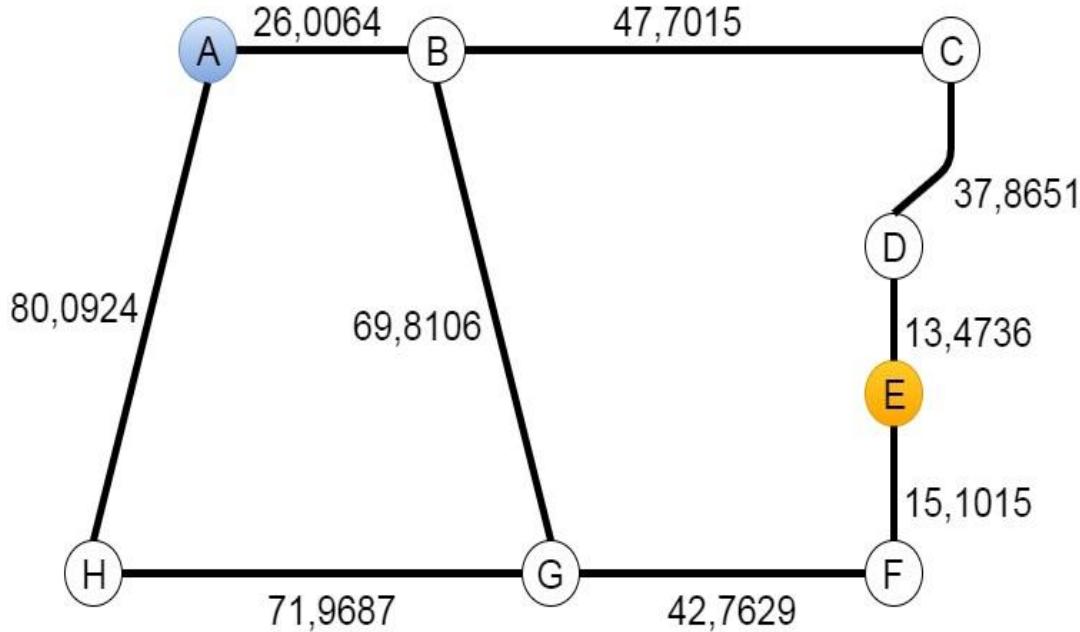
**Tabel 4. 11** Tabel keterangan halaman menu petugas administrasi.

Fitur	Keterangan
Data bengkel dan tambal ban	Di dalam menu ini terdapat data id, tujuan, jenis, dan koordinat dari bengkel dan tambal ban lalu terdapat juga <i>button edit</i> dan <i>delete</i> untuk mengolah data yang dilakukan oleh <i>admin</i> .
Menampilkan data <i>map</i>	Di dalam menu ini terdapat data id, tujuan, jenis dan ada fitur lihat <i>map</i> yang berfungsi untuk melihat map lebih detail.
Menampilkan lokasi <i>map</i>	Di dalam menu ini terdapat nama lokasi dan peta lokasi bengkel dan tambal ban.
Tambah data	Di dalam menu ini <i>admin</i> dapat menambah data bengkel dan tambal ban dengan memasukkan id, tujuan, jenis bengkel atau tambal ban, koordinat (berisi <i>latitude</i> dan <i>longitude</i> ) dan <i>admin</i> juga dapat mengedit data bengkel dan tambal ban dengan menekan <i>button edit</i> yang berada pada menu data bengkel dan tambal ban lalu di <i>edit</i> di dalam menu ini.
<i>Button Simpan</i>	<i>Button</i> ini berfungsi untuk menyimpan data ke <i>database</i> .
<i>Button Batal</i>	<i>Button</i> ini berfungsi untuk menghapus data yang berada pada <i>form</i> yang berada pada menu tambah dan <i>edit</i> data.

#### 4.2.9 Perancangan Metode Dijkstra

Pada perancangan metode dijkstra di bawah ini ditunjukan start awal atau titik awalnya berada pada titik A (Pertigaan Jl. Prapanca atau Pertigaan Jl. Cakarayam Gg. 3) yang berada pada koordinat -7.464152, 112.428114 dan untuk titik tujuannya berada pada titik E (Tambal ban Brawijaya V) yang berada pada koordinat -7.46459, 112.42916, gambar perancangan metode dijkstra di tunjukan pada **gambar 4.11**. Di antara setiap titik - titik memiliki jarak untuk detail setiap jarak antar titik di tunjukan pada **table 4.12**. Pada **gambar 4.12** dan **gambar 4.13** adalah perhitungan metode dijkstra yang dilakukan di *microsoft excel*. Dan untuk perbandingan antar rute di tunjukan pada **table 4.13**.

Di bawah ini adalah perancangan metode dijkstra akan ditunjukan pada **gambar 4.11**.



**Gambar 4. 11** Perancangan metode *dijkstra*.

Di bawah ini adalah jarak antar titik akan ditunjukan pada **tabel 4.12**.

**Tabel 4. 12** Jarak antar titik.

No	Dari	Ke	Jarak (Meter)
1	A	B	26,0064
2	A	H	80,0924
3	B	C	47,7015
4	B	G	69,8106
5	C	D	37,8651
6	D	E	13,4736
7	H	G	71,9687
8	G	F	42,7629
9	F	E	15,1015

Data pada **tabel 4.12** di atas di dapat dari node yang sudah secara manual lalu si simpan pada *database* skripsi3 tabel graph *field* simpul\_awal, simpul\_akhir, jalur dan untuk data jarak di dapat dari bantuan aplikasi google maps lalu di simpan pada *database* skripsi3 tabel graph *field* bobot.

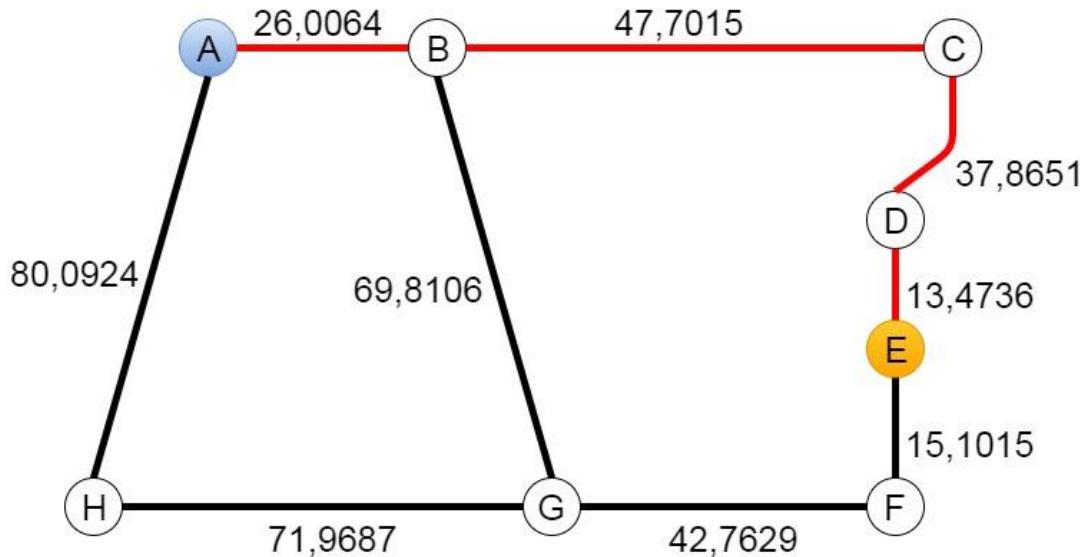
Di bawah ini adalah perhitungan metode dijkstra akan ditunjukan pada **gambar 4.12.**

## Algoritma Dijkstra untuk Mencari Jarak Jalur Terpendek

Jumlah Kota	8	Jarak Terpendek dari A ke E								125.0466								
A,B,C,D,E,F,G,H		Rute Terpendek di tempuh A - E								A - B - C - D - E								
untuk setiap langkah, (1) dari kota terpilih, jumlahkan angka terkecil sebelumnya dengan semua jarak dari kota terpilih ke kota terhubung yang kotaknya tidak berwarna biru, (2) bandingkan hasil penjumlahan pada langkah 1 dengan nilai pada kotak sebelumnya, tuliskan ke dalam kotak nilai terkecil dari kedua nilai tersebut (3) dari kotak yang masih berwarna putih, pilih angka terkecil maka pilihan kota berikutnya adalah kota dengan nilai terkecil tersebut																		
dari ↔ ke	jarak	cek	Langkah	Pilih	Terkecil	A	B	C	D	E	F	G	H					
A      B	26.0064					0+26.0064							0+80.0924					
A      H	80.0924		1	A	26.0064	0	26.0064	~	~	~	~	~	80.0924					
B      C	47.7015			B	A	A	A	A	A	A	A	A						
B      G	69.8106					26.0064												
C      D	37.8651						26.0064+47.7015											
D      E	13.4736		2	B	73.7079	0	26.0064	73.7079	~	~	~	95.817	80.0924					
H      G	71.9687			C	A	A	B	A	A	A	B	A						
G      F	42.7629					73.7079												
F      E	15.1015						73.7079+37.8651											
			3	C	80.0924	0	26.0064	73.7079	111.573	~	~	95.817	80.0924					
				H	A	A	B	C	A	A	B	A						
						80.0924												
			4	H	95.817	0	26.0064	73.7079	111.573	~	~	95.817	80.0924					
				G	A	A	B	C	A	A	B	A						
					95.817													
			5	G	111.573	0	26.0064	73.7079	111.573	~	138.5799	95.817	80.0924					
				D	A	A	B	C	A	G	B	A						
					111.573													
			6	D	125.0467	0	26.0064	73.7079	111.573	125.0466	138.5799	95.817	80.0924					
				E	A	A	B	C	D	G	B	A						
					125.0466													
			7	E	125.047	0	26.0064	73.7079	111.573	125.0466	138.5799	95.817	80.0924					
				stop	A	A	B	C	D	G	B	A						
					125.0466+15.1015													

**Gambar 4. 12 Perhitungan metode dijkstra.**

Di bawah ini adalah hasil perhitungan metode dijkstra akan ditunjukan pada **gambar 4.13.**



**Gambar 4. 13** Hasil perhitungan metode *dijkstra*.

Berdasarkan data pada **gambar 4.11** dan perhitungan yang telah di lakukan pada **gambar 4.12** di dapatkan hasil perhitungan metode *dijkstra* pada **gambar 4.13** maka di dapat jalur terpendek yaitu pada jalur A – B – C – D – E dengan nilai 125,04 meter. Meter panjang jalan di dapat dari *database* skripsi3 tabel *graph field* bobot. Untuk perbandingan antar rute di tunjukan pada **table 4.13**.

**Tabel 4. 13** Daftar Rute.

No	Rute	Jarak (Meter)
1.	A – B – C – D – E.	125,04
2.	A – B – G – F – E.	153,68
3.	A – H – G – F – E.	209,92
4.	A – H – G – B – C – D – E.	320,91

## BAB V. IMPLEMENTASI

Setelah melakukan tahapan perancangan aplikasi, maka pada bab ini akan dibahas tentang proses implementasi, yaitu realisasi perancangan menjadi nyata. Bagian pada bab ini meliputi implementasi basis data dan implementasi sistem. Implementasi basis data menggunakan *MySQL* dan implementasi sistem dibangun menggunakan bahasa pemrograman *JAVA* sebagai bahasa pemrograman berbasis *mobile* untuk pengguna dan menggunakan bahasa pemrograman *PHP* sebagai bahasa pemrograman berbasis *website* untuk *admin*.

### 5.1 Implementasi Basis Data

Implementasi basis data merupakan hasil implementasi dari perancangan basis data sesuai perancangan yang telah dilakukan sebelumnya pada bab sebelumnya. **gambar 5.1** merupakan hasil implementasi basis data yang dibangun menggunakan *MySQL*.

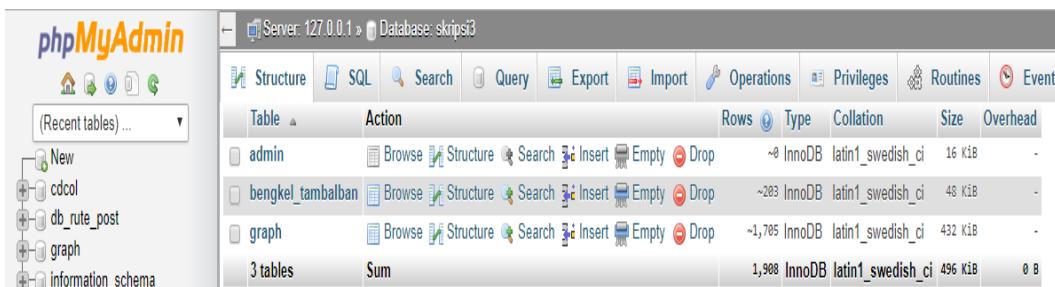


Table	Action	Rows	Type	Collation	Size	Overhead
admin	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	~8	InnoDB	latin1_swedish_ci	16 Kib	-
bengkel_tambalan	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	~283	InnoDB	latin1_swedish_ci	48 Kib	-
graph	<a href="#">Browse</a> <a href="#">Structure</a> <a href="#">Search</a> <a href="#">Insert</a> <a href="#">Empty</a> <a href="#">Drop</a>	~1,785	InnoDB	latin1_swedish_ci	432 Kib	-
3 tables		Sum			1,908 InnoDB latin1_swedish_ci	496 Kib

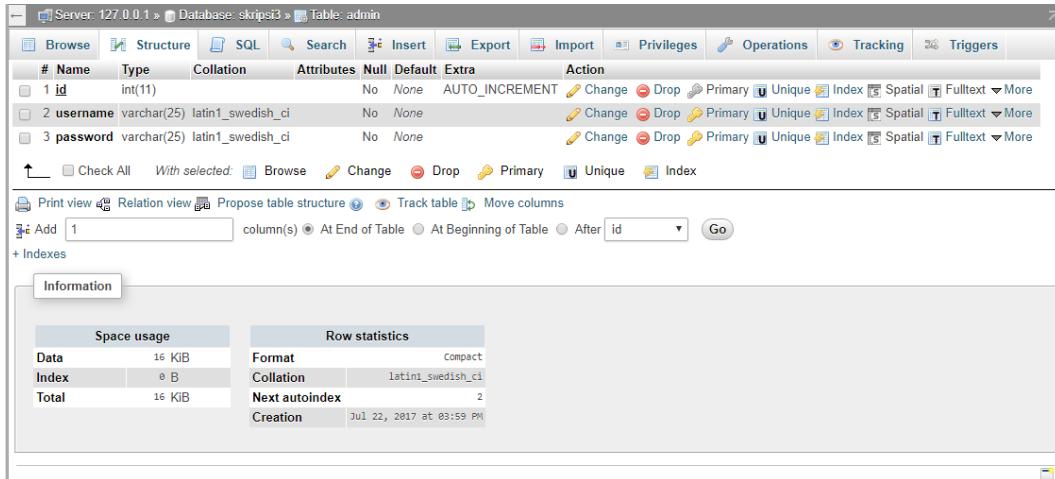
**Gambar 5.1 Database.**

Basis data skripsi3 mempunyai 3 tabel, yaitu tabel *admin*, *graph*, dan *bengkel\_tambalan*. Implementasi tabel-tabel tersebut adalah sebagai berikut:

#### 5.1.1 Gambar Tabel Admin

Pada **gambar 5.2** adalah implementasi dari rancangan tabel *admin* yang berfungsi menyimpan data *admin*. Tabel ini memiliki *field* antara lain: *id* (*primary key*), *username*, dan *password*.

Tabel *Admin* akan ditunjukan pada **gambar 5.2** seperti di bawah ini.

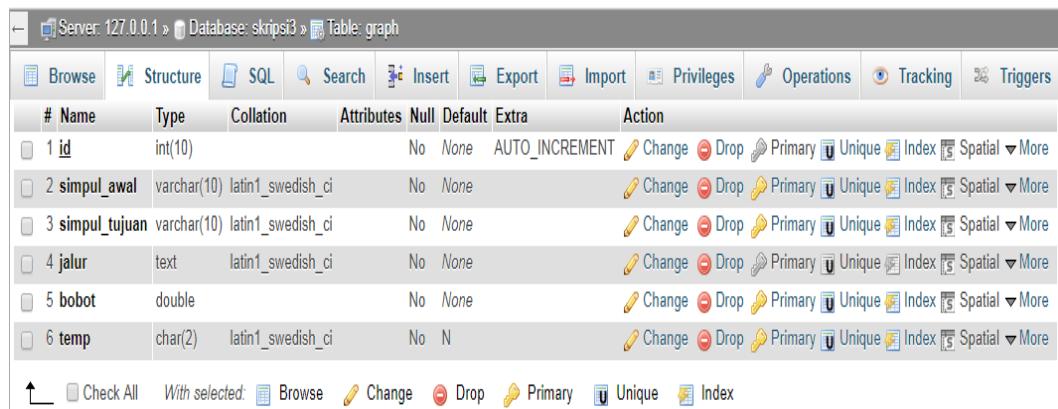


The screenshot shows the MySQL Workbench interface for the 'admin' table in the 'skripsi3' database. The table has three columns: 'id' (int(11)), 'username' (varchar(25)), and 'password' (varchar(25)). The 'id' column is set as the primary key and auto-increments. The 'username' and 'password' columns have collation 'latin1\_swedish\_ci'. The table structure includes tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers. Below the table, there is an 'Information' panel displaying 'Space usage' (Data: 16 KIB, Index: 0 B, Total: 16 KIB) and 'Row statistics' (Format: Compact, Collation: latin1\_swedish\_ci, Next autoindex: 2, Creation: Jul 22, 2017 at 03:59 PM).

**Gambar 5. 2 Tabel Admin.**

### 5.1.2 Gambar Tabel Graph

Pada **gambar 5.3** adalah implementasi dari rancangan tabel *graph* yang berfungsi menyimpan data bengkel dan tambal ban. Tabel ini memiliki *field* antara lain: id (*primary key*), simpul\_awal, simpul\_tujuan, jalur, bobot, dan temp.



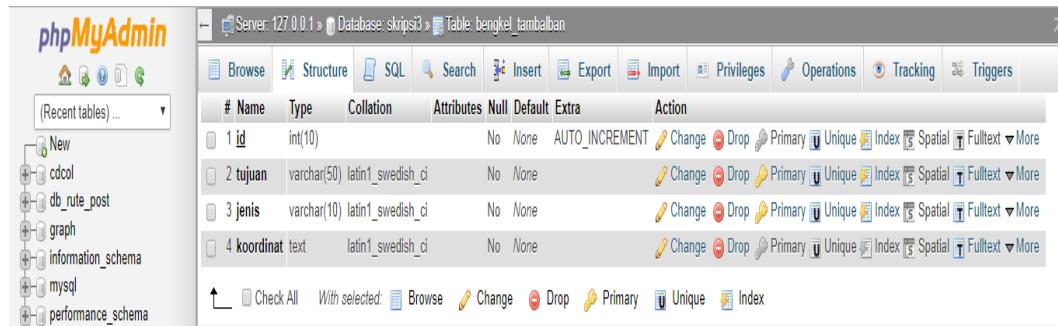
The screenshot shows the MySQL Workbench interface for the 'graph' table in the 'skripsi3' database. The table has six columns: 'id' (int(10)), 'simpul\_awal' (varchar(10)), 'simpul\_tujuan' (varchar(10)), 'jalur' (text), 'bobot' (double), and 'temp' (char(2)). The 'id' column is set as the primary key and auto-increments. The 'simpul\_awal' and 'simpul\_tujuan' columns have collation 'latin1\_swedish\_ci'. The table structure includes tabs for Browse, Structure, SQL, Search, Insert, Export, Import, Privileges, Operations, Tracking, and Triggers.

**Gambar 5. 3 Tabel Graph.**

### 5.1.3 Gambar Tabel Bengkel\_tambalban

Pada **gambar 5.4** adalah implementasi dari rancangan tabel bengkel\_tambalban yang berfungsi menyimpan data lokasi bengkel tambal ban. Tabel ini memiliki *field* antara lain: id (*primary key*), tujuan, jenis, dan koordinat.

Tabel bengkel\_tambalban akan ditunjukkan pada **gambar 5.4** seperti di bawah ini.



The screenshot shows the phpMyAdmin interface with the following details:

- Server:** 127.0.0.1
  - Database:** skripsi3
  - Table:** bengkel\_tambalban
- Structure Tab:** Active tab showing the table structure.
- Table Data:**

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	id	int(10)	latin1_swedish_ci		No	AUTO_INCREMENT		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">Fulltext</a> <a href="#">More</a>
2	tujuan	varchar(50)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">Fulltext</a> <a href="#">More</a>
3	jenis	varchar(10)	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">Fulltext</a> <a href="#">More</a>
4	koordinat	text	latin1_swedish_ci		No	None		<a href="#">Change</a> <a href="#">Drop</a> <a href="#">Primary</a> <a href="#">Unique</a> <a href="#">Index</a> <a href="#">Spatial</a> <a href="#">Fulltext</a> <a href="#">More</a>
- Buttons at the bottom:**
  - Up
  - Check All
  - With selected: [Browse](#), [Change](#), [Drop](#), [Primary](#), [Unique](#), [Index](#)

**Gambar 5. 4** Tabel bengkel\_tambalban.

## 5.2 Implementasi Antar Muka

Implementasi antar muka merupakan proses pengubahan sistem yang telah dirancang kemudian diterapkan dalam program. Berikut ini akan dijelaskan implementasi dari antarmuka dari Rancang Bangun Sistem Informasi Geografis Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel dengan Metode *Dijkstra* Berbasis Webservice (Studi Kasus Kota Mojokerto).

### 5.2.1 Halaman Utama

Halaman utama adalah *website* yang dapat digunakan oleh *User* dan *Admin* yang ditunjukkan pada **gambar 5.5**. *website* digunakan oleh *User* dan *Admin* untuk mencarian jalur terpendek menuju lokasi bengkel dan tambal ban di Kota Mojokerto. Untuk desain yang digunakan *website* ini memilih desain yang *simple* agar mudah digunakan. Pemilihan desain menggunakan *template bootstrap*. Pada halaman utama sendiri terdapat menu yang dapat dipilih seperti user atau admin padat memilih.

Halaman utama akan ditunjukkan pada **gambar 5.5** seperti di bawah ini.



**Gambar 5. 5** Halaman utama.

Pada **gambar 5.6** di bawah ini adalah *script* untuk menampilkan *marker*.

```
<?php
include "Main.php";

// koneksi
$koneksi = new Main();
$koneksi = $koneksi->koneksi;

$query = mysqli_query($koneksi, "select * from bengkel_tambalban");
while ($data = mysqli_fetch_array($query, MYSQLI_ASSOC))
{
    $dataMarker[] = array(
        'id' => $data['id'],
        'tujuan' => $data['tujuan'],
        'jenis' => $data['jenis'],
        'lat' => (float)trim(explode(',', $data['koordinat']))[0]),
        'lng' => (float)trim(explode(',', $data['koordinat']))[1]));
}

// query
$sql      = "SELECT * FROM bengkel_tambalban";
$query   = mysqli_query($koneksi, $sql);

// select option
// echo 'TUJUAN : <select id="select_tujuan" onchange="choose_destination(this.value)">';
// echo '<option value="pilih">-- PILIH --</option>';

// while($fetch = mysqli_fetch_array($query, MYSQLI_ASSOC))
// {
//     $idTujuan      = $fetch['id'];
//     $koordinat      = $fetch['koordinat'];
//     $exp_koordinat = explode(',', $koordinat);
//     $json_koordinat = '{"lat": '.$exp_koordinat[0].', "lng": '.$exp_koordinat[1].'}';

//     echo "<option value='".$json_koordinat' data-id='".$idTujuan.$fetch[tujuan].'">$fetch[tujuan]</option>";
// }
// echo '</select>';
?>
```

**Gambar 5. 6** *script* untuk menampilkan *marker*.

Pada **gambar 5.7** di bawah ini adalah *script* untuk menampilkan peta, *user*.

```
function initialize() {
    /* setup map */
    var mapOptions = {
        zoom: 14,
        center: new google.maps.LatLng(-7.470467, 112.440347)
    };
    map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

    // GPS POSISI SAYA
    navigator.geolocation.getCurrentPosition(function(position) {
        posisi_lat = position.coords.latitude;
        posisi_long = position.coords.longitude;

        markeruser = new google.maps.Marker({
            position: {lat: posisi_lat, lng: posisi_long},
            map: map,
            icon: 'https://fuadalidawami.000webhostapp.com/saya2.png',
            draggable: true,
            title: 'test drag',
        });
        map.setCenter(new google.maps.LatLng(posisi_lat, posisi_long));
    });
}
```

**Gambar 5. 7** *script* untuk menampilkan peta dan *user*.

Pada **gambar 5.7** di bawah ini adalah *script* untuk membedakan warna *marker* sesuai jenis.

```
var dataMarker = JSON.parse('<?>=json_encode($dataMarker)?>');
var icons = 'https://fuadalidawami.000webhostapp.com/iconsMarker40.png';
var infowindow = new google.maps.InfoWindow();

for (i=0 ; i<dataMarker.length; i++){
    if(dataMarker[i].jenis == 'Bengkel'){
        var gambar_marker = 'img/bengkel.png';
    } else {
        var gambar_marker = 'img/tambal.png';
    }

    markerdestination = new google.maps.Marker({
        position: {lat: dataMarker[i].lat, lng: dataMarker[i].lng},
        map: map,
        icon: gambar_marker,
        draggable: false,
        title: dataMarker[i].tujuan,
        idTujuan: dataMarker[i].id
    });

    var infowindow = new google.maps.InfoWindow()
    var content = dataMarker[i].tujuan;

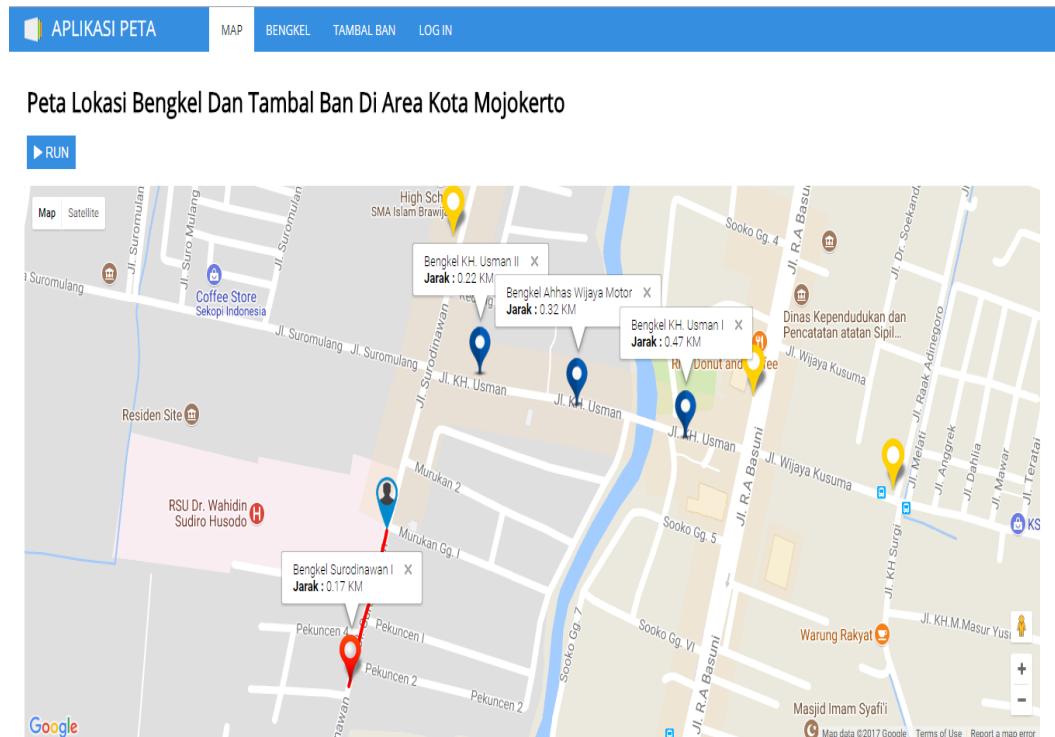
    google.maps.event.addListener(markerdestination,'click', (function(markerdestination,content,infowindow){
        return function() {
            infowindow.setContent(content);
            infowindow.open(map,markerdestination);
            $('select option[data-id="'+ this.idTujuan + '"]').attr("selected","selected");
            $('select option[data-id="'+ this.idTujuan + '"]').change()
        };
    })(markerdestination,content,infowindow));
}
```

**Gambar 5. 8** *script* untuk membedakan warna *marker* sesuai jenis.

Pada halaman utama sendiri terdapat beberapa fitur yang dapat di pilih oleh *user* atau *admin*, seperti fitur Tujuan dan *RUN*, fitur Bengkel, fitur Tambal Ban, dan fitur *Login*.

### 5.2.2 Fitur Tujuan dan *RUN*

Fitur Tujuan ini berfungsi untuk memilih bengkel atau tambal ban yang akan di jadikan sebagai tujuan atau juga bisa dengan langsung mengklik salah satu *marker* yang terdapat pada *map* sebagai tujuan, lalu button *RUN* ini berfungsi untuk menjalankan metode *dijkstra* agar sistem dapat menampilkan rute terpendek menuju tujuan dan sistem dapat menampilkan jarak tempuh menuju tujuan. Fitur ini di tunjukan seperti pada **gambar 5.9**.



**Gambar 5.9** Fitur *RUN*.

Pada **gambar 5.10** di bawah ini adalah *script* untuk menampilkan rute terdekat.

```
function send_dijkstra(){
    if(markeruser == ''){
        alert('Isi dulu koordinat user & tujuan');
        return false;
    }
    // $('#txtJarak').text(' ');
    console.log(markeruser.position.lat());
    console.log(markeruser.position.lng());
    now_koord_user = '{"lat": ' + markeruser.position.lat() + ', "lng": ' + markeruser.position.lng() + '}';
    // OTOMATIS MENCARI TITIK TERDEKAT
    var dataMarker = JSON.parse('<?=json_encode($dataMarker)?>');
    var acuan = 99999999;
    var latitude_terdekat = 0;
    var longitude_terdekat = 0;
    var tujuan_terdekat = "";
    var id_tujuan = "";

    for (i=0 ; i<dataMarker.length; i++){
        var lat = dataMarker[i].lat;
        var lng = dataMarker[i].lng;

        var posisi_saya = new google.maps.LatLng(markeruser.position.lat(), markeruser.position.lng());
        var latLng = new google.maps.LatLng(lat, lng);
        var distance_real = google.maps.geometry.spherical.computeDistanceBetween(posisi_saya, latLng);
        var distance = parseFloat(distance_real) / 1000; // (MENJADIKAN KILOMETER)
        distance = distance.toFixed(2);
        var jarak = parseFloat(distance_real);

        if(jarak < acuan){
            acuan = jarak;
            tujuan_terdekat = dataMarker[i].tujuan;
            latitude_terdekat = lat;
            longitude_terdekat = lng;
            id_tujuan = i;
        }
    }
}
```

**Gambar 5. 10** *script* untuk menampilkan rute terdekat.

Pada **gambar 5.11** di bawah ini adalah *script* untuk menampilkan rute terdekat.

```
}
now_koord_destination = '{"lat": ' + latitude_terdekat + ', "lng": ' + longitude_terdekat + '}';

var mapOptions = {
    zoom: 14,
    center: new google.maps.LatLng(latitude_terdekat, longitude_terdekat)
};
map = new google.maps.Map(document.getElementById('map-canvas'), mapOptions);

for (i=0 ; i<dataMarker.length; i++){
    var lat = dataMarker[i].lat;
    var lng = dataMarker[i].lng;

    if(i != id_tujuan){

        if(dataMarker[i].jenis == 'Bengkel'){
            var gambar_marker = 'img/bengkel.png';
        } else {
            var gambar_marker = 'img/tambal.png';
        }

        markerdestination = new google.maps.Marker({
            position: {lat: dataMarker[i].lat, lng: dataMarker[i].lng},
            map: map,
            icon: gambar_marker,
            draggable: false,
            title: dataMarker[i].tujuan,
            idTujuan: dataMarker[i].id
        });

        var posisi_saya = new google.maps.LatLng(markeruser.position.lat(), markeruser.position.lng());
        var latLng = new google.maps.LatLng(lat, lng);
        var distance_real = google.maps.geometry.spherical.computeDistanceBetween(posisi_saya, latLng);
        var distance = parseFloat(distance_real) / 1000; // (MENJADIKAN KILOMETER)
        distance = distance.toFixed(2);
    }
}
```

**Gambar 5. 11** *script* untuk menampilkan rute terdekat.

Pada **gambar 5.12** di bawah ini adalah *script* untuk menampilkan rute terdekat.

```

var infowindow = new google.maps.InfoWindow()
var content = dataMarker[i].tujuan + '  
 <b>Jarak :</b> '+distance+' KM';

google.maps.event.addListener(markerdestination,'click', (function(markerdestination,content,infowindow){
    return function() {
        infowindow.setContent(content);
        infowindow.open(map,markerdestination);
        $('select option[data-id="'+ this.idTujuan +'"]').attr("selected","selected");
        $('select option[data-id="'+ this.idTujuan +'"]').change();
    };
})(markerdestination,content,infowindow));
}

markeruser = new google.maps.Marker({
    position: {lat: markeruser.position.lat(), lng: markeruser.position.lng()},
    map: map,
    icon: 'https://fudalidawami.000webhostapp.com/saya2.png',
    draggable: true,
    title: 'Me',
});
markerterdekat = new google.maps.Marker({
    position: {lat: latitude_terdekat, lng: longitude_terdekat},
    map: map,
    icon: 'img/terdekat.png',
    draggable: false
});
markerterdekat.addListener('click', function() {
    infowindow.setContent(tujuan_terdekat + '  
 <b>Jarak :</b> '+parseFloat(acuan/1000).toFixed(2) +' KM');
    infowindow.open(map, markerterdekat);
});

```

**Gambar 5. 12** *script* untuk menampilkan rute terdekat.

Pada **gambar 5.13** di bawah ini adalah *script* untuk menampilkan rute terdekat.

```

// RUTE MENGIKUTI JALAN
var directionsService = new google.maps.DirectionsService;
var directionsDisplay = new google.maps.DirectionsRenderer({
    map: map,
    preserveViewport: true,
    suppressMarkers: true,
    polylineOptions: { strokeColor: "red", strokeWeight: 4 }
});

var start = new google.maps.LatLng(markeruser.position.lat(), markeruser.position.lng());
var end = new google.maps.LatLng(latitude_terdekat, longitude_terdekat);

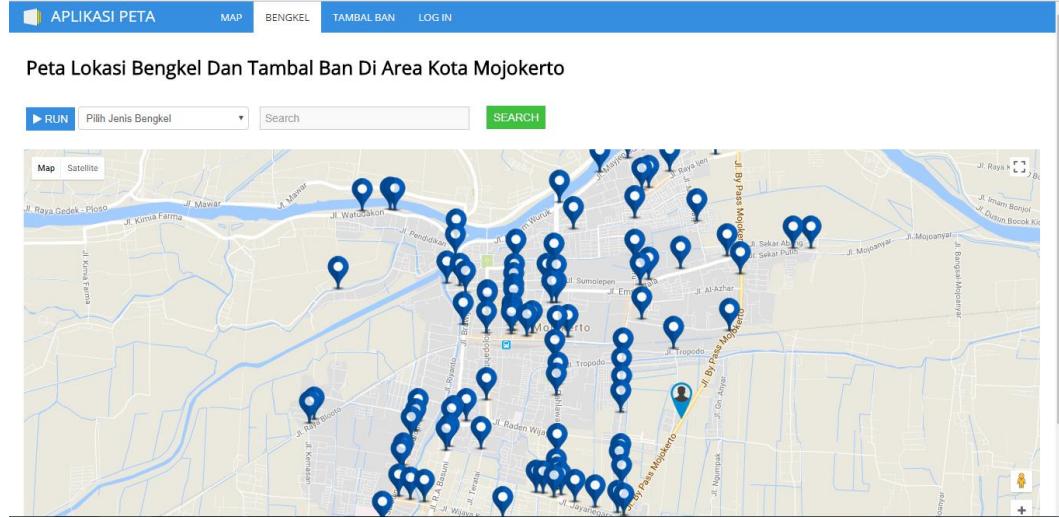
var bounds = new google.maps.LatLngBounds();
bounds.extend(start);
bounds.extend(end);
map.fitBounds(bounds);
var request = {
    origin: start,
    destination: end,
    travelMode: google.maps.TravelMode.WALKING
};
directionsService.route(request, function (response, status) {
    if (status == google.maps.DirectionsStatus.OK) {
        directionsDisplay.setDirections(response);
        directionsDisplay.setMap(map);
    } else {
        alert("Geocode was not successful for the following reason: " + status);
    }
});
// AKHIR RUTE MENGIKUTI JALAN

```

**Gambar 5. 13** *script* untuk menampilkan rute terdekat.

### 5.2.3 Fitur Menu Bengkel

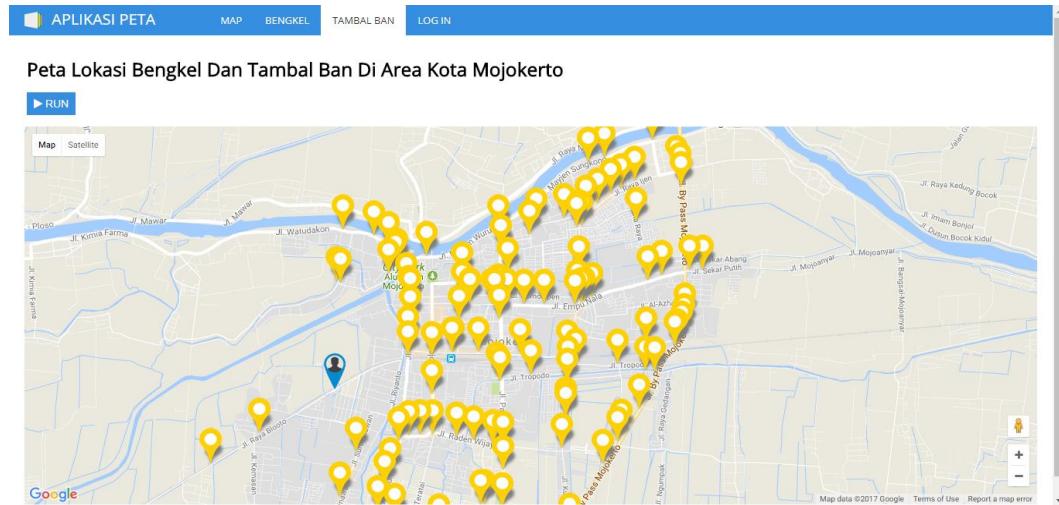
Fitur Bengkel ini berfungsi untuk memfilter marker yang berada pada map agar yang tampil hanya bengkel sebagai tujuan. Fitur ini di tunjukan seperti pada **gambar 5.14.**



**Gambar 5. 14** Fitur Bengkel.

### 5.2.4 Fitur Menu Tambal Ban

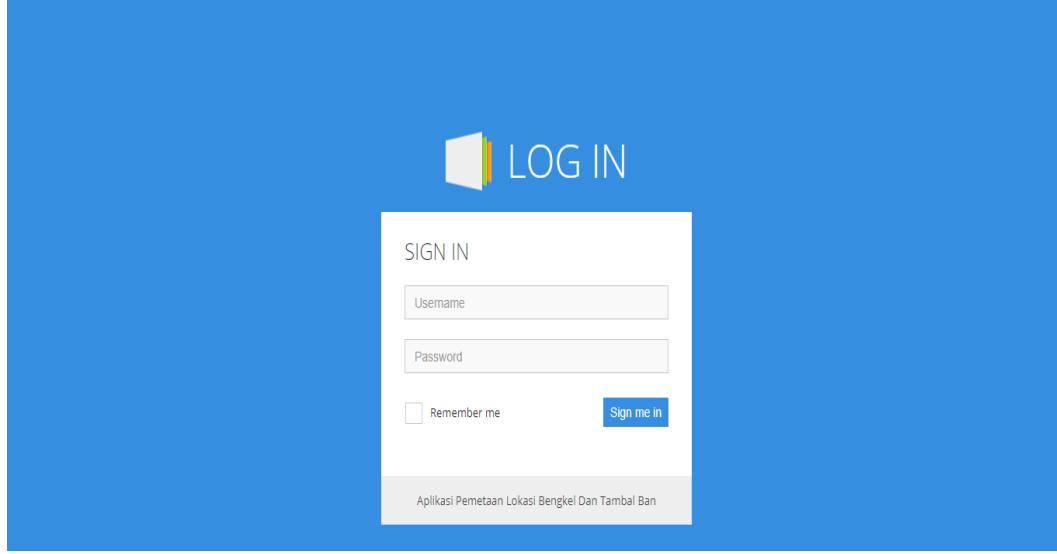
Fitur Tambal Ban ini berfungsi untuk memfilter marker yang berada pada map agar yang tampil hanya tambal ban sebagai tujuan. Fitur ini di tunjukan seperti pada **gambar 5.15.**



**Gambar 5. 15** Fitur Bengkel.

### 5.2.5 Halaman *Login*

Berikut ini adalah halaman *login* untuk *admin*. *Admin* harus memasukkan *username* dan *password* untuk bisa mendapatkan hak akses disini juga terdapat *button cancel* yang berfungsi untuk kembali ke halaman utama. Halaman *login* ditunjukkan pada **gambar 5.16**.



**Gambar 5. 16** Halaman *Login*.

Pada **gambar 5.17** di bawah ini adalah *script* untuk halaman *Login*.

```
<?php
$button = $_POST['login'];
if(isset($button)){
    $username = $_POST['form-username'];
    $password = $_POST['form-password'];
    $query = $db->prepare("SELECT * from admin where (username = '" . $_POST['username'] . "') and (password = '" . $_POST['password'] . "')", $db);
    $query->bindParam(':username', $username);
    $query->bindParam(':password', $password);
    $query->execute();

    if($query->rowCount() > 0){
        session_start();
        $_SESSION['u'] = $username;
        $_SESSION['p'] = $password;
        session_regenerate_id(true);
        header('location:ceklogin.php');
    }
}
?>
```

**Gambar 5. 17** *script* halaman *login*.

Pada **gambar 5.18** di bawah ini adalah *script* untuk cek *login*.

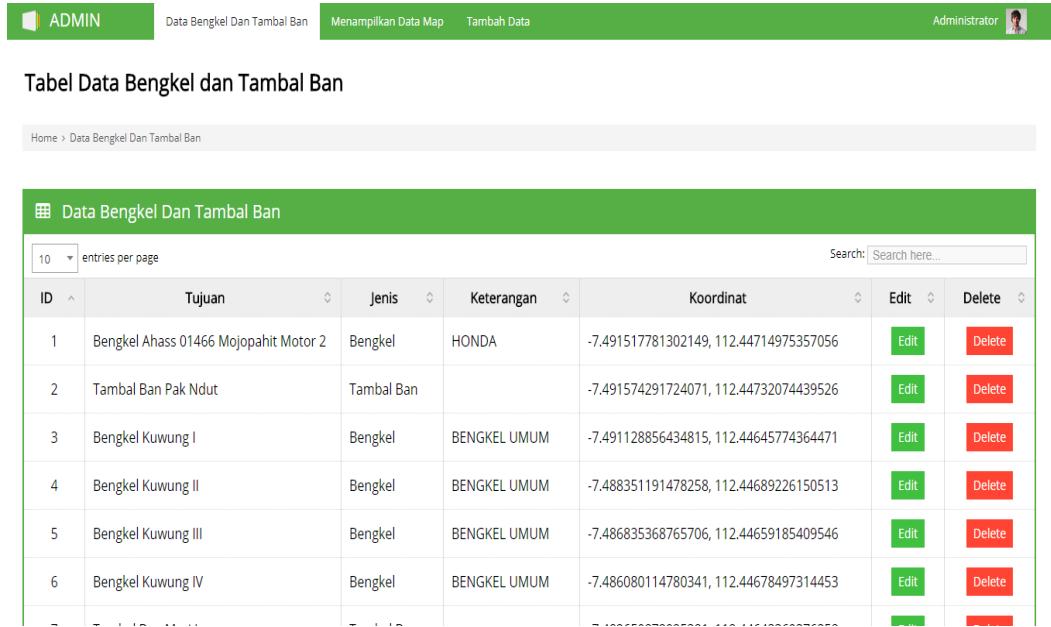
```
<?php
session_start();
$server = "localhost"; //ganti sesuai server Anda
$username = "root"; //ganti sesuai username Anda
$password = ""; //ganti sesuai password Anda
$db_name = "skripsi1"; //ganti sesuatu nama database Anda
$db = mysql_connect($server,$username,$password) or DIE("koneksi ke database gagal !!");
mysql_select_db($db_name) or DIE("nama database tersebut tidak ada !!");
$login = mysql_query("select * from admin where (username = '" . $_POST['username'] . "'") and (password = '" . $_POST['password'] . "'");
$rowcount = mysql_num_rows($login);
if ($rowcount == 1) {
$_SESSION['username'] = $_POST['username'];
header("Location: admin.php");
}
else
{
?>
<script>
alert('Gagal Login.');
window.location.href='login.php';
</script>
<?php
?>
```

**Gambar 5. 18** *script* cek *login*.

### 5.2.6 Halaman Utama Admin

Halaman utama *admin* adalah *website server* dari aplikasi yang digunakan oleh *admin*. Pada *website server* akan mengolah semua data yang akan digunakan untuk aplikasi pencarian jalur terpendek menuju lokasi bengkel tambal ban di Kota Mojokerto. *Website server* ini juga hanya bisa diakses oleh administrator dari aplikasi. Sebelum masuk halaman utama *admin*, *admin* terlebih dahulu harus melakukan *login*. Halaman tersebut memuat fitur-fitur yang digunakan untuk mengelola isi dan *content* dari aplikasi pencarian jalur terpendek menuju lokasi bengkel tambal ban. Pada halaman ini terdapat fitur Data Bengkel dan Tambal ban, Menampilkan Data *Map*, Menampilkan Data Lokasi *Map*, Tambah dan *Edit* Data dan *logout*. Pada saat halaman utama di buka *admin* akan diarahkan langsung menuju Fitur Data Bengkel dan Tambal ban dimana fitur ini berfungsi untuk menampilkan semua data yang sudah tersimpan di *database*. Data tersebut antara lain id, tujuan, jenis, dan koordinat pada halaman ini *admin* mempunyai hak akses untuk mengolah data bengkel dan tambal ban pada halaman ini terdapat 2 tombol yaitu tombol *edit* dan *delete*. Tombol *edit* berfungsi untuk mengedit data, sedangkan tombol *delete* untuk menghapus data.

Halaman utama admin ditunjukkan pada **gambar 5.19** seperti di bawah ini.



The screenshot shows a web-based administration interface. At the top, there's a green header bar with the title 'ADMIN' and three menu items: 'Data Bengkel Dan Tambal Ban', 'Menampilkan Data Map', and 'Tambah Data'. On the right side of the header, it says 'Administrator' and shows a user profile icon. Below the header is a breadcrumb navigation bar: 'Home > Data Bengkel Dan Tambal Ban'. The main content area has a green header 'Data Bengkel Dan Tambal Ban'. It contains a table with the following data:

ID	Tujuan	Jenis	Keterangan	Koordinat	Edit	Delete
1	Bengkel Ahass 01466 Mojopahit Motor 2	Bengkel	HONDA	-7.491517781302149, 112.44714975357056	<button>Edit</button>	<button>Delete</button>
2	Tambal Ban Pak Ndut	Tambal Ban		-7.491574291724071, 112.44732074439526	<button>Edit</button>	<button>Delete</button>
3	Bengkel Kuwung I	Bengkel	BENGKEL UMUM	-7.491128856434815, 112.44645774364471	<button>Edit</button>	<button>Delete</button>
4	Bengkel Kuwung II	Bengkel	BENGKEL UMUM	-7.488351191478258, 112.44689226150513	<button>Edit</button>	<button>Delete</button>
5	Bengkel Kuwung III	Bengkel	BENGKEL UMUM	-7.486835368765706, 112.44659185409546	<button>Edit</button>	<button>Delete</button>
6	Bengkel Kuwung IV	Bengkel	BENGKEL UMUM	-7.486080114780341, 112.44678497314453	<button>Edit</button>	<button>Delete</button>

**Gambar 5. 19** Halaman Utama Admin.

Pada **gambar 5.20** di bawah ini adalah *script* untuk halaman *admin*.

```
<thead>
<tr>
    <th>Id</th>
    <th>Tujuan</th>
    <th>Jenis</th>
    <th>Koordinat</th>
    <th></th>
    <th></th>
</tr>
</thead>
<tbody>
<?php
$temp = 0;
$query = $db->prepare('SELECT * from bengkel_tambalban');
$query->execute();
$qq = $query->fetchAll(PDO::FETCH_ASSOC);
foreach ($qq as $key => $value) {
    echo "<tr>";
    echo "<td>". $value['id'] . "</td>";
    echo "<td>". $value['tujuan'] . "</td>";
    echo "<td>". $value['jenis'] . "</td>";
    echo "<td>". $value['koordinat'] . "</td>";
?
    <td class="text-center">
        <a href="editbaru.php?id=<?php echo $value['id'] ?>"><button type="submit" name="Submit" class="btn btn-success"> Edit</button></a>
    </td>
    <td class="text-center">
        <a onclick="return confirm('Apakah anda yakin?')" href="delete.php?id=<?php echo $value['id'] ?>"><button type="submit" name="Submit" class="btn btn-danger"> Delete</button></a>
    </td>
}
?
echo "</tr>";
$temp++;
if ($temp == 500) {
    break;
}
}
?>
```

**Gambar 5. 20** script halaman admin.

Pada **gambar 5.20** di bawah ini adalah *script* untuk halaman *button edit*.

```
<?php
include 'co.php';

if (isset($_POST)) {
    // Tidak aman ini script Mudah kena sql inject
    /*
    $sql = "UPDATE tabel biodata SET nama = '$_POST[nama]',
        jenis_kelamin = '$_POST[jenis_kelamin]',
        alamat = '$_POST[alamat]',
        no_hp = '$_POST[no_hp]',
        status = '$_POST[status]'.
    WHERE id = '$_POST[id]'";
    $dbh->exec($sql);
    */

    // Lebih amannya
    // by NIN4N
    $sql = "UPDATE bengkel_tambalan SET tujuan=?,
        jenis=?,
        koordinat=?,
        WHERE id=?";
    $stmt = $dbh->prepare($sql);
    $stmt->bindParam(1,$_POST['tujuan']);
    $stmt->bindParam(2,$_POST['jenis']);
    $stmt->bindParam(3,$_POST['koordinat']);
    $stmt->bindParam(4,$_POST['id']);
    $stmt->execute();
    if ($stmt->rowCount() == 0){
        exit("Maaf gagal melakukan Update. Ini terjadi biasanya data belum ada yang di rubah <br /><a href='admin.php'>Kembali</a>");
    }
}

header("location:admin.php");
?>
```

**Gambar 5. 21** *script button edit.*

Pada **gambar 5.20** di bawah ini adalah *script* untuk halaman *button delete*.

```
<?php

if (isset($_GET['id'])){
require_once 'koneksi1.php';

$database = new koneksi();
$db=$database->getConnection();

$id = $_GET['id'];
$query = $db->prepare("DELETE FROM bengkel_tambalan WHERE id = :id");
$query->bindParam(':id', $id);
$query->execute();
}
header('location:admin.php');
?>
```

**Gambar 5. 22** *script button delete.*

Jika tombol *edit* diklik, maka informasi yang akan dirubah akan tampil di halaman *input* data seperti ditunjukkan **gambar 5.23** dan *script* untuk menampilkan halaman *edit* di tunjukkan pada **gambar 5.24**. Namun jika keadaan *form* sudah terisi,

*Admin* cukup merubah data yang diinginkan saja tanpa perlu memasukkan semua informasi dari awal.

Halaman *edit* ditunjukkan pada **gambar 5.23** seperti di bawah ini.

The screenshot shows a web application interface titled 'Edit Data'. At the top, there is a navigation bar with links for 'Data Bengkel Dan Tambal Ban', 'Menampilkan Data Map', and 'Tambah Data'. On the right, it shows the user 'Administrator' with a profile picture. The main content area has a title 'Edit Data' and a breadcrumb trail: 'Home > Data Bengkel Dan Tambal Ban > Edit Data'. Below this is a form with the following fields:

- Tujuan: Bengkel Ahass 01466 Mojopahit Motor 2
- Jenis Bengkel atau Tambal Ban: Pilih Jenis Bengkel / Tambal Ban
- Jenis Bengkel: Pilih Jenis Bengkel / Tambal Ban
- Koordinat: -7.491517781302149, 112.44714975357056

At the bottom of the form are two buttons: 'Simpan' (Save) and 'Batal' (Cancel). To the right of the form is a Google Map showing the location of the entered coordinates. The map displays several streets and landmarks in Mojokerto, including Jl. Pasar Mojokerto, Jl. Gempol-Mojokerto, and Stikes Bina Sehat F.

**Gambar 5. 23** Halaman *edit*.

Pada **gambar 5.24** di bawah ini adalah *script* untuk menampilkan halaman *edit*.

```
<form name="postform" action="update.php" role="form" method="post">
<input type="hidden" name="id" value=<?php echo $data['id']; ?> />

<!!--<br/>
<h4><label>Id</label></h4>
<div class="controls" class="control-label" for="nama">
<textarea style="width:100px; height:50px;" id="nama" placeholder="Id" name="id"> <?php echo $data['id']; ?></textarea>
</div>-->

<br/>
<h4><label>Tujuan</label></h4>
<div class="controls" class="control-label" for="tujuan">
<textarea style="width:500px; height:75px;" id="tujuan" placeholder="tujuan" name="tujuan"><?php echo $data['tujuan']; ?></textarea>
</div>

<br/>
<div class="form-controls">
<h4><label>Jenis Bengkel atau Tambal Ban</label></h4>
<div class="control-label" for="jenis">
<select style="width:500px" style="height:50px" class="form-control" id="jenis" name="jenis">
<option disabled selected>Pilih Jenis Bengkel / Tambal Ban</option>
<option>Bengkel</option>
<option>Tambalan</option>
</select>
</div>
</div>

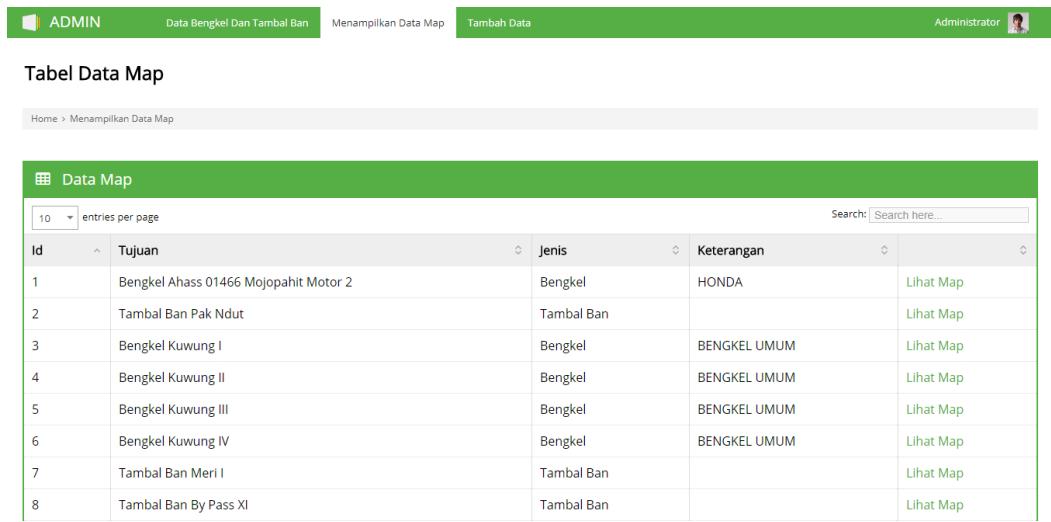
<br/>
<h4><label>Koordinat</label></h4>
<div class="controls" class="control-label" for="koordinat">
<textarea style="width:500px; height:75px;" id="koordinat" placeholder="koordinat" name="koordinat"><?php echo $data['koordinat']; ?></textarea>
</div>

<br/>
<div class="controls">
<!--<button type="Submit" class="btn btn-success" name="Submit">Submit</button>-->
<button type="Submit" name="Submit" class="btn btn-success" ><i class="icon-edit"></i> Simpan</button>
<button type="reset" class="btn btn-danger" name=":reset">Reset</button>
<!--<button type="button" class="btn btn-primary" onclick="cari_alamat()">Cari Alamat</button></td>-->
</div>
</form>
```

**Gambar 5. 24** *script* untuk menampilkan halaman *edit*.

### 5.2.7 Halaman Menampilkan Data Map

Halaman ini berisi data id, tujuan, dan jenis dari data bengkel dan tambal ban. Di sini sebagai tambahan jika *admin* mengklik *text Lihat Map*, *admin* akan diarahkan langsung menuju ke halaman Menampilkan Lokasi *Map*. Halaman Menampilkan Data *Map* dapat di lihat seperti pada **gambar 5.25** dan *script* untuk menampilkan halaman Menampilkan Data *Map* dapat di lihat seperti pada **gambar 5.26**.



The screenshot shows a web application interface with a green header bar. The header contains the text 'ADMIN', 'Data Bengkel Dan Tambal Ban', 'Menampilkan Data Map', 'Tambah Data', and 'Administrator'. Below the header is a table titled 'Data Map' with 8 rows of data. The columns are labeled 'Id', 'Tujuan', 'Jenis', 'Keterangan', and 'Lihat Map'. The data includes various repair shop names and their details.

Id	Tujuan	Jenis	Keterangan	Lihat Map
1	Bengkel Ahass 01466 Mojopahit Motor 2	Bengkel	HONDA	<a href="#">Lihat Map</a>
2	Tambal Ban Pak Ndut	Tambal Ban		<a href="#">Lihat Map</a>
3	Bengkel Kuwung I	Bengkel	BENGKEL UMUM	<a href="#">Lihat Map</a>
4	Bengkel Kuwung II	Bengkel	BENGKEL UMUM	<a href="#">Lihat Map</a>
5	Bengkel Kuwung III	Bengkel	BENGKEL UMUM	<a href="#">Lihat Map</a>
6	Bengkel Kuwung IV	Bengkel	BENGKEL UMUM	<a href="#">Lihat Map</a>
7	Tambal Ban Meri I	Tambal Ban		<a href="#">Lihat Map</a>
8	Tambal Ban By Pass XI	Tambal Ban		<a href="#">Lihat Map</a>

Gambar 5. 25 Halaman Menampilkan Data *Map*.

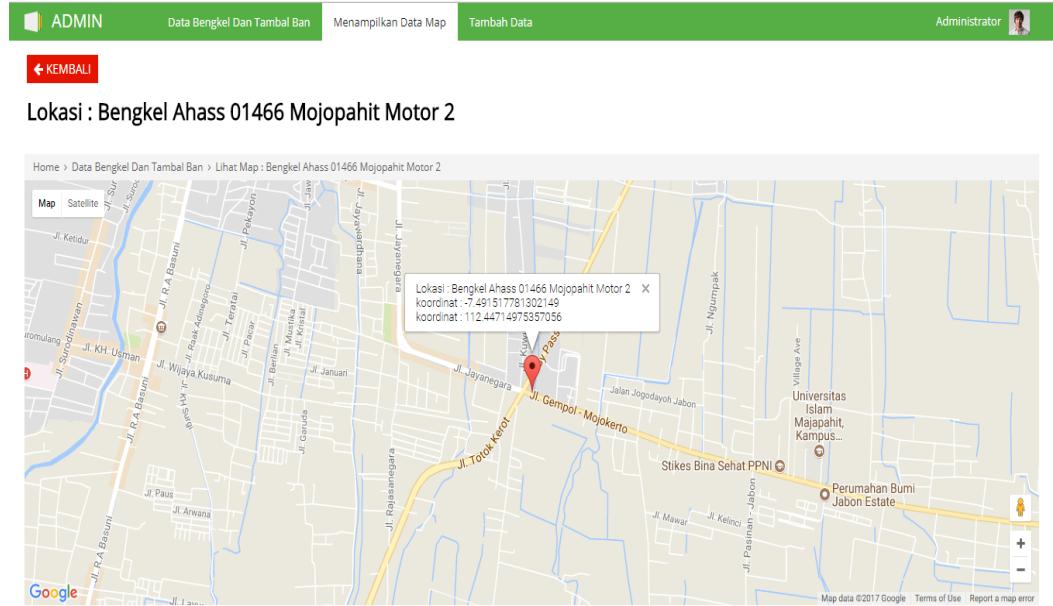
```
<thead>
<tr>
<th>Id</th>
<th>Tujuan</th>
<th>Jenis</th>
<th></th>
</tr>
</thead>
<tbody>
<?php
$cari = mysql_query("select * from bengkel_tambalban");

while($dapat = mysql_fetch_array($cari))
{
echo "
<tr>
<td>$dapat[id]</td>
<td>$dapat[tujuan]</td>
<td>$dapat[jenis]</td>
<td><a href='tampil_map1.php?id=$dapat[id] '>Lihat Map</a></td>
</tr>";}
?>
```

Gambar 5. 26 *script* untuk menampilkan halaman "Menampilkan Data *Map*".

### 5.2.8 Halaman Menampilkan Lokasi Map

Halaman ini berisi nama bengkel atau tambal ban dan letak lokasi bengkel atau tambal ban. Halaman Menampilkan Data *Map* dapat di lihat pada **gambar 5.27** dan *script* untuk menampilkan halaman Menampilkan Lokasi *Map* dapat di lihat seperti pada **gambar 5.28** dan **gambar 5.29**.



**Gambar 5. 27** Halaman Menampilkan Lokasi *Map*.

```
<script type="text/javascript">
function initialize() {
    var mapOptions = {
        zoom: 15,
        center: new google.maps.LatLng(<?php echo "$dcari[koordinat]"; ?>)
    }
    var map = new google.maps.Map(document.getElementById('map-canvas'),
        mapOptions);

    setMarkers(map, beaches);
}

var beaches = [
    [<?php echo "$dcari[tujuan]"; ?>, <?php echo "$dcari[koordinat]"; ?>, <?php echo "$dcari[koordinat]"; ?>],
];
function setMarkers(map, locations) {
    var shape = {
        coords: [1, 1, 1, 20, 18, 20, 18, 1],
        type: 'poly'
    };
    var infoWindow = new google.maps.InfoWindow;
    for (var i = 0; i < locations.length; i++) {
        var beach = locations[i];
        var myLatLng = new google.maps.LatLng(beach[1], beach[2]);
        var marker = new google.maps.Marker({
            position: myLatLng,
            map: map,
            icon: beach[4],
            shape: shape,
            title: beach[0],
            zIndex: beach[3]
        });
        var html = 'Lokasi : '+beach[0]+'<br/>koordinat : '+beach[1]+'<br/>koordinat : '+beach[2]+'';
        bindInfoWindow(marker, map, infoWindow, html);
    }
}
```

**Gambar 5. 28** *script* untuk menampilkan halaman "Menampilkan Lokasi *Map*".

Pada **gambar 5.29** di bawah ini adalah *script* untuk menampilkan halaman "Menampilkan Lokasi Map".

```
function bindInfoWindow(marker, map, infoWindow, html) {
  google.maps.event.addListener(marker, 'click', function() {
    infoWindow.setContent(html);
    infoWindow.open(map, marker);
  });
}

google.maps.event.addDomListener(window, 'load', initialize);
</script>
```

**Gambar 5.29** script untuk menampilkan halaman "Menampilkan Lokasi Map".

### **5.2.9 Halaman Tambah Data**

Halaman ini merupakan halaman yang berfungsi untuk mengelola semua data yang digunakan oleh aplikasi pencarian jalur terpendek menuju bengkel dan tambal ban. Pada fitur Tambah Data, *admin* bisa menambahkan data bengkel dan tambal ban yang akan dimasukkan ke *database* ditunjukkan pada **gambar 5.30** dan *script* untuk menampilkan halaman Tambah Data dapat di lihat seperti pada **gambar 5.31** dan **gambar 5.32**.

ADMIN Data Bengkel Dan Tambal Ban Menampilkan Data Map Tambah Data Administrator

## Tambah Data

Home > Tambah Data

### Tambah Data

ID	<input type="text"/>
Tujuan	<input type="text"/>
Jenis Bengkel atau Tambal Ban	<input type="button" value="Pilih Jenis Bengkel / Tambal Ban"/>
Jenis Bengkel	<input type="button" value="Pilih Jenis Bengkel / Tambal Ban"/>
Koordinat	<input type="text"/>

**Simpan** **Batal**

Map data ©2017 Google Terms of Use Report a map error

**Gambar 5. 30 Halaman Tambah Data.**

Pada **gambar 5.31** di bawah ini adalah *script* untuk menampilkan halaman "Tambah Data".

```
<div class="control-group">
    <label for="id" class="control-label">ID</label>
    <div class="controls">
        <input required type="text" name="id" id="id" class="input-xlarge" style="width: 100%; height: 30px;">
    </div>
</div>

<div class="control-group">
    <label for="tujuan" class="control-label">Tujuan</label>
    <div class="controls">
        <input required type="text" name="tujuan" id="tujuan" class="input-xlarge" style="width: 100%; height: 30px;">
    </div>
</div>

<div class="control-group">
    <label for="jenis" class="control-label">Jenis Bengkel atau Tambal Ban</label>
    <div class="controls">
        <select style="width: 100%;" class="form-control" id="jenis" name="jenis">
            <option disabled selected>Pilih Jenis Bengkel / Tambal Ban</option>
            <option value="Bengkel">Bengkel</option>
            <option value="Tambal Ban">Tambal Ban</option>
        </select>
    </div>
</div>

<div class="control-group">
    <label for="koordinat" class="control-label">Koordinat</label>
    <div class="controls">
        <input required type="text" name="koordinat" id="koordinat" class="input-xlarge" style="width: 100%; height: 30px;">
    </div>
</div>

<div class="form-actions">
    <input type="submit" class="btn btn-primary" value="Simpan" name="aksi">
    <input type="reset" class="btn btn-danger" value="Batal">
</div>
```

**Gambar 5. 31** *script* untuk menampilkan halaman "Tambah Data".

Pada **gambar 5.32** di bawah ini adalah *script* untuk menampilkan halaman "Tambah Data".

```
<script type="text/javascript">
var markers = [];
var map;

function initialize() {
    var mapOptions = {
        zoom: 15,
        center: new google.maps.LatLng(-7.470467, 112.440347)
    }
    map = new google.maps.Map(document.getElementById('map-canvas'),
        mapOptions);

    // GPS POSISI SAYA
    navigator.geolocation.getCurrentPosition(function(position) {
        posisi_lat = position.coords.latitude;
        posisi_long = position.coords.longitude;
        map.setCenter(new google.maps.LatLng(posisi_lat, posisi_long));
    });

    google.maps.event.addListener(map, 'click', function(event) {
        deleteMarkers();
        addMarker(event.latLng);
        setMapOnAll(map);
        $("#koordinat").val(event.latLng.lat(), "+event.latLng.lng());
    });
}

function bindInfoWindow(marker, map, infoWindow, html) {
    google.maps.event.addListener(marker, 'click', function() {
        infoWindow.setContent(html);
        infoWindow.open(map, marker);
    });
}

function addMarker(location) {
    var marker = new google.maps.Marker({
        position: location,
        map: map
    });
    markers.push(marker);
}
```

**Gambar 5. 32** *script* untuk menampilkan halaman "Tambah Data".

Pada **gambar 5.33** di bawah ini adalah *script* untuk menampilkan halaman "Tambah Data".

```
// Sets the map on all markers in the array.  
function setMapOnAll(map) {  
    for (var i = 0; i < markers.length; i++) {  
        markers[i].setMap(map);  
    }  
  
    // Removes the markers from the map, but keeps them in the array.  
    function clearMarkers() {  
        setMapOnAll(null);  
    }  
  
    // Shows any markers currently in the array.  
    function showMarkers() {  
        setMapOnAll(map);  
    }  
  
    // Deletes all markers in the array by removing references to them.  
    function deleteMarkers() {  
        clearMarkers();  
        markers = [];  
    }  
  
    google.maps.event.addListener(window, 'load', initialize);  
</script>
```

**Gambar 5. 33** *script* untuk menampilkan halaman "Tambah Data".

Pada **gambar 5.30.** memiliki 2 tombol. yaitu tombol simpan, dan batal. Tombol simpan digunakan untuk menyimpan data yang sudah diisikan pada *form* yang ada. Tombol batal digunakan untuk mengosongkan *form* yang ada jika admin membatalkan untuk mengisi data.

## BAB VI. PENGUJIAN DAN PEMBAHASAN

Bab ini berisi hasil-hasil pengujian yang dilakukan terhadap Sistem Informasi Geografis Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel Menggunakan Metode *Dijkstra* Berbasis *Webservice* (Studi Kasus Kota Mojokerto).

### 6.1 Uji Coba

Proses uji coba dilakukan dengan sub - sub bab uji coba fungsional dan sub-sub bab uji coba akurasi. Uji coba fungsional dilakukan untuk mengetahui apakah sistem yang dibangun sudah menyediakan fungsi - fungsi yang sesuai dengan yang dibutuhkan. Pengujian aplikasi dilakukan dengan cara memverifikasi data yang ditampilkan aplikasi dengan kodisi riil dilapangan.

#### 6.1.1 Pengujian Spesifikasi Perangkat Uji Coba

Proses uji coba sistem ini dilakukan pada perangkat *laptop* yang ditunjukkan pada **Tabel 6.1** dan **Tabel 6.3** dan perangkat *smartphone* dengan spesifikasi yang diuraikan pada **Tabel 6.2** dan **Tabel 6.4** sebagai berikut:

**Tabel 6. 1** Spesifikasi perangkat keras *PC / Laptop*.

Perangkat Keras	Keterangan
<i>Processor</i>	Intel Core i3 Processor 2350M
<i>RAM</i>	2 GB
<i>Harddisk</i>	500 GB
<i>VGA</i>	<i>Graphics ATI Radeon HD 6370M 1G.</i>

**Tabel 6. 2** Spesifikasi perangkat keras *smartphone*.

Perangkat Keras	Keterangan
<i>CPU</i>	Intel® Atom™ Z2560 ( 1.6 GHz )
<i>Internal</i>	16 GB
<i>RAM</i>	2 GB
<i>Network</i>	3G, HSPA, EDGE, 4G LTE, 4G LTE Cat 4

**Tabel 6. 3** Spesifikasi perangkat lunak *PC / Laptop*.

Perangkat Lunak	Keterangan
<i>Windows 7</i>	Sistem Operasi yang digunakan untuk menjalankan program.
<i>Android Studio</i>	<i>Software</i> untuk menulis program.
<i>Java dan PHP</i>	Bahasa pemrograman.
<i>MySQL</i>	<i>Software</i> untuk pembuatan <i>database</i> sistem.

**Tabel 6. 4** Spesifikasi perangkat lunak *smartphone*.

Perangkat Lunak	Keterangan
OS Android™ v5.0 (Lollipop)	Sistem Operasi yang digunakan untuk menjalankan program.

## 6.2 Pengujian Performa Fungsionalitas Sistem

Pengujian performa fungsionalitas sistem ini dilakukan dengan cara menjalankan setiap fitur dalam aplikasi dan melihat apakah hasilnya sudah berjalan sesuai dengan yang diharapkan. Hasil pengujian dapat dilihat dalam **Tabel 6.5**.

**Tabel 6. 5** Fungsionalitas Sistem.

No	Pola Pengujian	Hasil Pengujian	Status Pengujian
<i>Website Admin</i>			
1.	Proses <i>login</i> .	<i>Login</i> berhasil.	Sesuai.
2.	Proses <i>logout</i> .	<i>Logout</i> berhasil.	Sesuai.
3.	Proses perhitungan setelah memilih lokasi bengkel atau tambal ban.	Proses perhitungan setelah memilih lokasi bengkel tambal ban berhasil.	Sesuai
4.	Kelola data lokasi bengkel tambal ban		
	a. Menampilkan data lokasi bengkel tambal ban.	Data bengkel tambal ban ditampil.	Sesuai
	b. Mengubah data lokasi bengkel tambal ban.	Data bengkel tambal ban diubah.	Sesuai
	c. Menghapus data lokasi bengkel tambal ban.	Data bengkel tambal ban dihapus.	Sesuai
	d. Menampilkan data	Data bengkel tambal ban	Sesuai

	lokasi bengkel tambal ban setelah di <i>edit</i> .	ditampilkan.	
<b>Aplikasi Pengguna</b>			
5.	a. Masukkan lokasi bengkel atau tambal ban.	Tampil <i>marker</i> , rute dan jarak tempuh sesuai bengkel atau tambal ban yang di pilih.	Sesuai
	b. Masukkan lokasi sesuai gps.	Tampil lokasi berdasarkan gps.	Sesuai
	c. Mengklik <i>button</i> Bengkel.	Hanya Menampilkan lokasi bengkel pada <i>map</i> .	Sesuai
	d. Mengklik <i>button</i> Tambal Ban.	Hanya Menampilkan lokasi tambal ban pada <i>map</i> .	Sesuai
6.	<b>Hasil Pencarian</b>		
	a. Menampilkan informasi bengkel dan tambal ban.	Tampil informasi bengkel dan tambal ban.	Sesuai
	b. Menampilkan jalur terpendek.	Tampil jalur terpendek.	Sesuai
	c. Menampilkan jarak tempuh.	Tampil jarak tempuh.	Sesuai

### 6.3 Analisis Hasil Uji Coba Sistem

Analisis hasil uji coba yang didapat dari setelah melakukan uji coba sistem yaitu:

- a. Pada sistem, *Admin* mampu melakukan proses *login* sesuai dengan rancangan yang telah dibuat.
- b. Pada sistem, *admin* mampu melakukan kelola data bengkel dan tambal ban.
- c. Pada sistem, Pengguna dapat melakukan pencarian berdasarkan kriteria yang ada. Lalu setelah menekan tombol *RUN*, proses perhitungan Algoritma *Dijkstra* berjalan kemudian akan menghasilkan jalur terpendek menuju bengkel atau tambal ban yang dipilih sebelumnya.

#### 6.4 Hasil Uji Coba dengan Responden

Untuk menguji sistem dari segi tampilan, *friendly user*, kehandalan atau keakurasiannya serta manfaat dari aplikasi yang dikembangkan, telah dilakukan penyebaran kuesioner kepada 20 responden. Responden yang dimaksud adalah para warga Mojokerto dan juga pendatang. Pendapat dari responden tersebut ditampilkan pada **Tabel 6.6**.

**Tabel 6.6** Data hasil kuesioner.

No	Pernyataan	S	R	TS	Jumlah
1	Aplikasi memiliki tampilan menarik.	15	5	0	20
2	Aplikasi mudah digunakan.	9	11	0	20
3	Proses yang cepat saat mencari rute terpendek.	14	6	0	20
4	Aplikasi membantu pengguna menemukan bengkel & tambalan.	16	4	0	20
5	Aplikasi bermanfaat bagi pendatang dan warga Mojokerto.	16	4	0	20

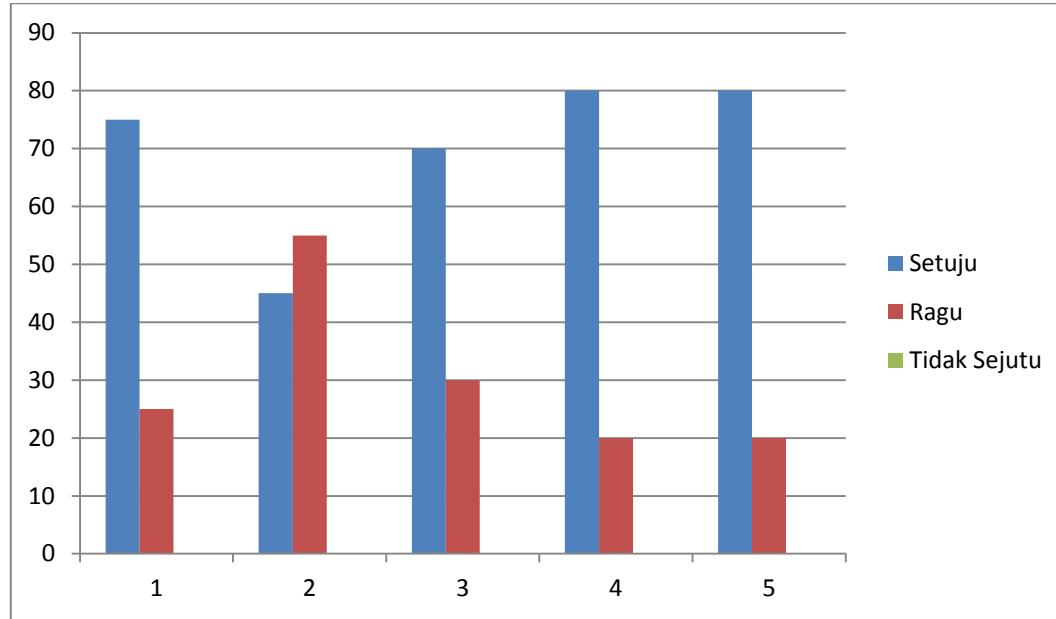
Keterangan :

S = Setuju

R = Ragu-Ragu

TS = Tidak Setuju

Dari hasil kuisisioner menunjukkan bahwa 15 responden berpendapat bahwa aplikasi memiliki tampilan menarik. Sedangkan untuk tingkat kemudahan atau *friendly user* 9 responden setuju. Untuk manfaat pencarian jalur terpendek menuju lokasi fasilitas kesehatan dengan cepat responden 14 mengatakan setuju. Begitu juga dengan pendapat bahwa 16 responden setuju aplikasi ini membantu mereka menemukan bengkel dan tambalan yang dimaksud dan 16 responden menganggap aplikasi ini bermanfaat bagi mereka.

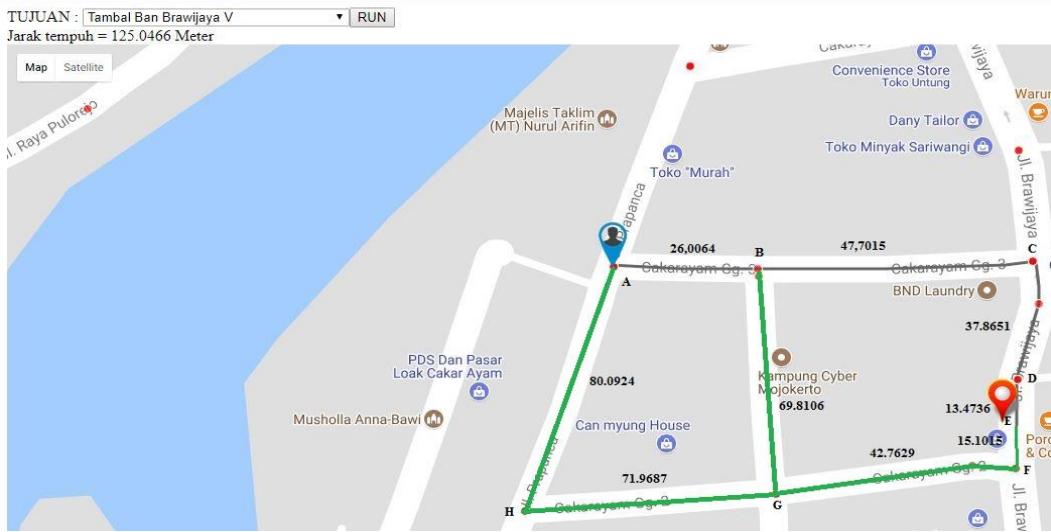
**Tabel 6. 7** Grafik presentase kusioner.

**Grafik 6.7** menunjukkan bahwa 75% responden berpendapat bahwa aplikasi memiliki tampilan menarik. Sedangkan untuk tingkat kemudahan atau *friendly user* 45% responden setuju. Untuk manfaat pencarian jalur terpendek menuju lokasi bengkel dan tambal ban dengan cepat responden 70% mengatakan setuju. Begitu juga dengan pendapat bahwa 80% aplikasi ini membantu mereka menemukan bengkel dan tambal ban yang dimaksud dan 80% responden menganggap aplikasi ini bermanfaat bagi mereka.

## 6.5 Pengujian Algoritma Dijkstra

Untuk menguji apakah algoritma *dijkstra* ini berjalan sesuai dengan hasil yang diharapkan, pengujian algoritma *dijkstra* dilakukan dengan melakukan pengecekan secara manual dengan menggunakan perhitungan di *microsoft excel* untuk menentukan apakah jalur yang diambil sudah optimal atau tidak. Pengujian algoritma *dijkstra* berdasarkan data seperti pada **tabel 4.12**.

Pada **gambar 6.1** di bawah ini adalah perhitungan pada program.



**Gambar 6. 1** Contoh perhitungan pada program.

Pada perhitungan pada program di atas start awal atau titik awalnya berada pada titik A (Pertigaan Jl. Prapanca atau Pertigaan Jl. Cakarayam Gg. 3) yang berada pada koordinat -7.464152, 112.428114 dan untuk titik tujuannya berada pada titik E (Tambal ban Brawijaya V) yang berada pada koordinat -7.46459, 112.42916.

Di bawah ini adalah perhitungan metode dijkstra akan ditunjukkan pada **gambar 6.2**.

## Algoritma Dijkstra untuk Mencari Jarak Jalur Terpendek

Di bawah ini adalah perhitungan metode dijkstra akan ditunjukkan pada **gambar 6.2**.

A	4	H	95.817	0	26.0064	73.7079	111.573	~	~	95.817	80.0924
		G		A	A	B	C	A	A	B	A
<b>95.817</b>											
A	5	G	111.573	0	26.0064	73.7079	111.573	~	138.5799	95.817	80.0924
		D		A	A	B	C	A	G	B	A
<b>111.573</b>											
<b>111.573 + 13.473</b>											
A	6	D	125.047	0	26.0064	73.7079	111.573	125.0466	138.5799	95.817	80.0924
		E		A	A	B	C	D	G	B	A
<b>125.0466</b>											
<b>125.0466 + 15.1015</b>											
A	7	E	125.047	0	26.0064	73.7079	111.573	125.0466	138.5799	95.817	80.0924
		stop		A	A	B	C	D	G	B	A

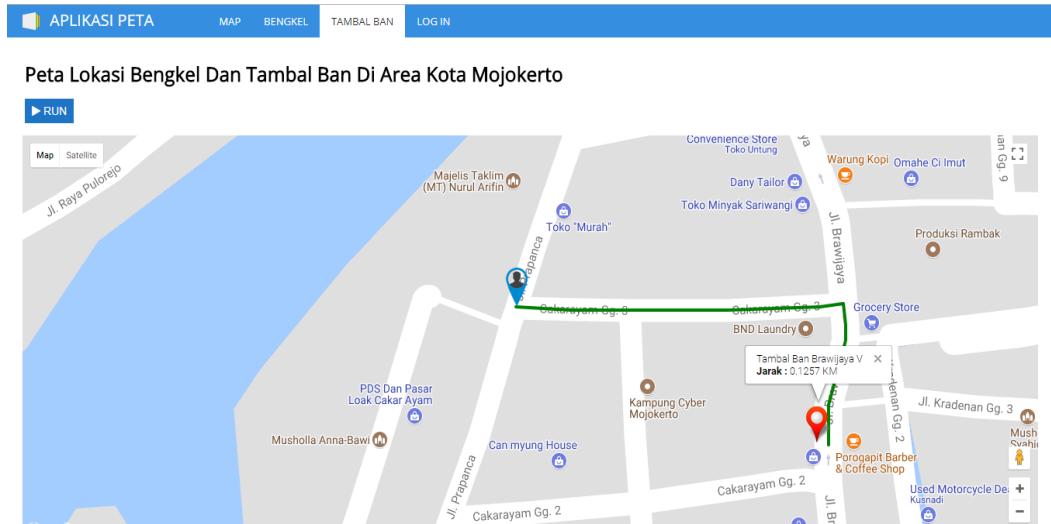
**Gambar 6.2** Perhitungan metode *dijkstra*.

Dari **gambar 6.1** terdapat 4 rute yang dapat di lalui dan berdasarkan hasil dari perhitungan yang telah di lakukan pada **gambar 6.2** maka di dapat jalur terpendek yaitu pada jalur A – B – C – D – E dengan nilai jarak tempuh 125,04 Meter panjang jalan di dapat dari *database* skripsi3 tabel graph *field* bobot. Untuk daftar rute dapat dilihat pada **tabel 6.8**.

**Tabel 6.8** Daftar rute.

No	Rute	Jarak (Meter)
1.	A – B – C – D – E.	125,04
2.	A – B – G – F – E.	153,68
3.	A – H – G – F – E.	209,92
4.	A – H – G – B – C – D – E.	320,91

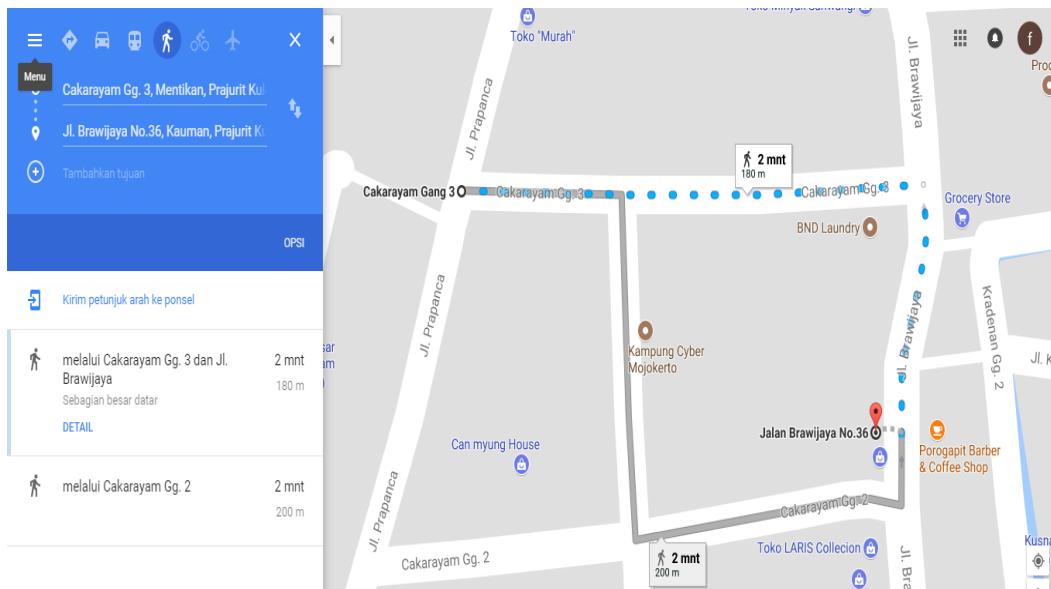
Pada **gambar 6.3** di bawah ini adalah perbandingan dengan *google maps*.



**Gambar 6. 3** Peta pada Aplikasi.

Berdasarkan **gambar 6.3** diketahui jarak tempuh 125 Meter.

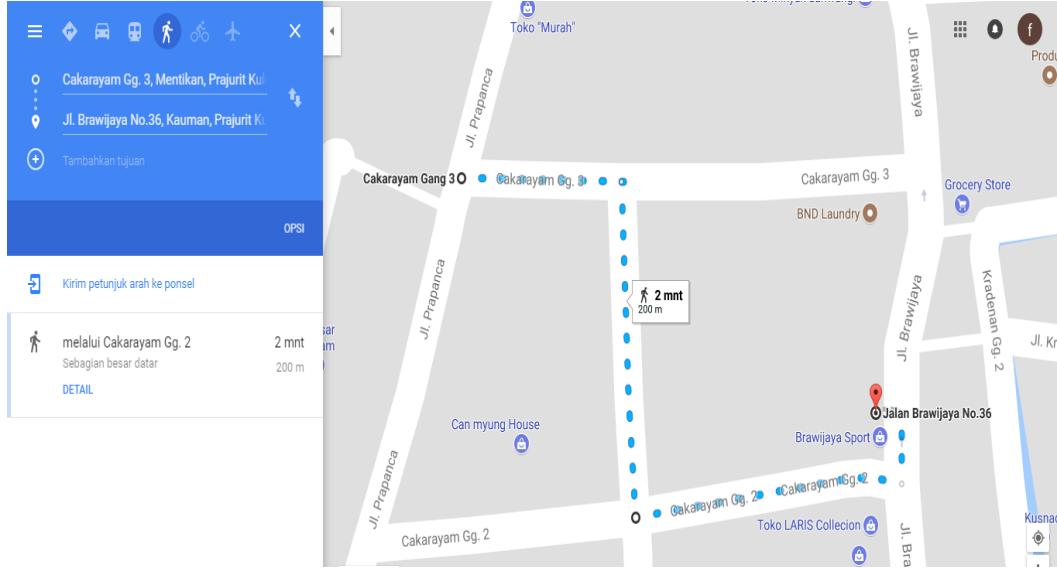
Pada **gambar 6.4** di bawah ini adalah perbandingan dengan *google maps*.



**Gambar 6. 4** Peta pada *google maps*.

Berdasarkan **gambar 6.4** diketahui jarak tempuh 180 Meter.

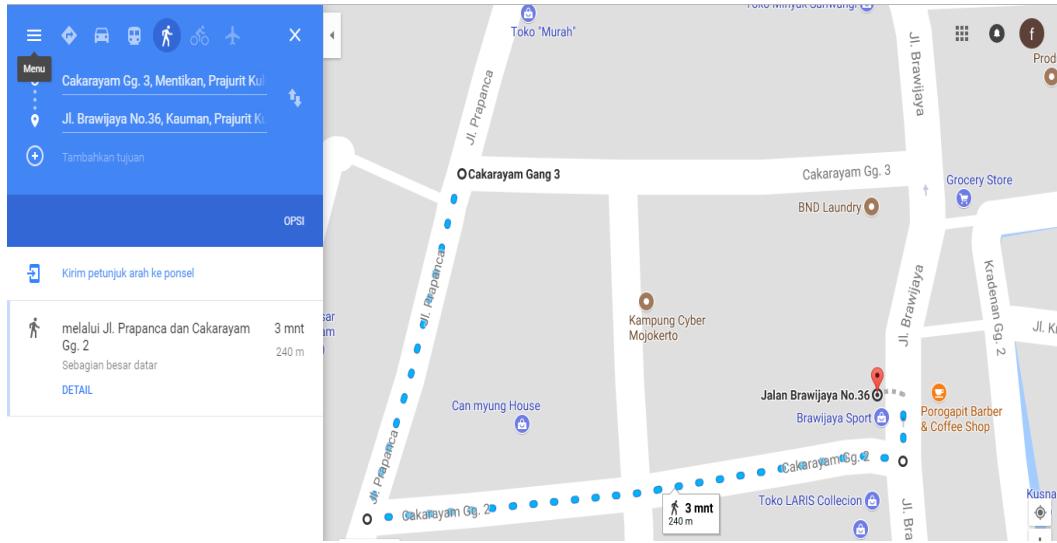
Pada **gambar 6.5** di bawah ini adalah perbandingan dengan *google maps*.



**Gambar 6. 5** Peta pada *google maps*.

Berdasarkan **gambar 6.5** diketahui jarak tempuh 200 Meter.

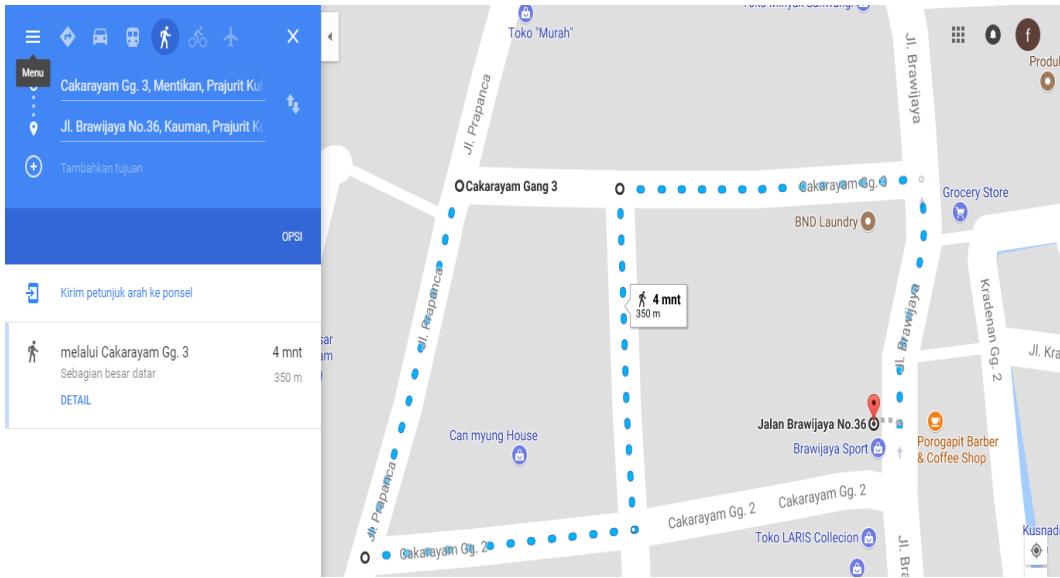
Pada **gambar 6.6** di bawah ini adalah perbandingan dengan *google maps*.



**Gambar 6. 6** Peta pada *google maps*.

Berdasarkan **gambar 6.6** diketahui jarak tempuh 240 Meter.

Pada **gambar 6.7** di bawah ini adalah perbandingan dengan *google maps*.



**Gambar 6.7** Peta pada *google maps*.

Berdasarkan **gambar 6.7** diketahui jarak tempuh 350 Meter.

## **BAB VII. KESIMPULAN**

### **7.1 Kesimpulan**

Berdasarkan hasil dari perancangan Aplikasi Sistem Informasi Geografis Penentuan Rute Terpendek Menuju Tambal Ban dan Bengkel dengan Metode Dijkstra Berbasis *Webservice* (Studi Kasus Kota Mojokerto), dapat di ambil beberapa kesimpulan, yaitu: dengan menggunakan data yang dihasilkan dari uji coba dan dari hasil kuesioner yang disebarluaskan kepada 20 responden, aplikasi mampu berjalan dengan baik secara fungsional untuk menampilkan jalur terdekat tambal ban dan bengkel di Kota Mojokerto menggunakan algoritma *dijkstra*, dapat membantu pendatang dan warga Mojokerto untuk menuju lokasi tambal ban dan bengkel di area Kota Mojokerto.

### **7.2 Saran**

Dalam menguji aplikasi ini dapat digunakan metode lain seperti metode genetika dan metode *A-Star* untuk dijadikan perbandingan keakuriasan dalam penelitian lebih lanjut dan untuk fitur dapat ditambah tidak hanya untuk pencarian tambal ban dan bengkel tetapi juga untuk fitur pencarian lokasi lain nya seperti restoran, rumah sakit, apotek dan lain-lain. Untuk penelitian lebih lanjut area dapat di perluas tidak hanya pada wilayah Kota Mojokerto saja tetapi wilayah – wilayah lain.

## **DAFTAR PUSTAKA**

- [1] Ardi Dwi Saputra I.S dan Yulmaini. 2012. "Perancangan Sistem Informasi Geografis (SIG) Pariwisata di Provinsi Lampung" Jurnal Informatika, Vol. 12, No. 2, Informatics & Business Institute Darmajaya.
- [2] Badan Pusat Statistik, 2016, Kota Mojokerto Dalam Angka 2016, Badan Pusat Statistik Kota Mojokerto.
- [3] Fitria dan Apri Triansyah. 2013."Implementasi Algoritma Dijkstra Dalam Aplikasi Untuk Menentukan Lintasan Terpendek Jalan Darat Antar Kota Di Sumatera Bagian Selatan". Jurnal Sistem Informasi, VOL. 5, NO. 2, The Informatics and Business Institute Darmajaya Bandar Lampung.
- [4] Hendry Ari Wibowo dan M.Said Hasibuan,M.Kom. 2016. "Sistem informasi Geografis Lokasi Bengkel dan Tambal Ban Kota Bandar Lampung Berbasis Android" Jurnal Teknik Informatika, Informatics And Business Institute Darmajaya, Bandar Lampung.
- [5] Muhammad Nizar Al Ayubi dan Ika Ratna I. A, S.Kom, MT. 2015. "Sistem Informasi Geografis Lokasi Pabrik di Kecamatan Ngoro Berbasis Android" Jurnal Teknik Informatika Universitas Muhammadiyah Sidoarjo.
- [6] Pradana, Bayu Endira. 2014. "Aplikasi Pencarian Bengkel Tambal Ban Berbasis Android Dengan Layanan *Cloud Computing*" Jurnal Informatika Mulawarman, Universitas Mulawarman.
- [7] UPT Dinas Pendapatan Provinsi Jawa Timur-Mojokerto. Data Jumlah Pembelian Kendaraan Bermotor Roda Dua di Kota Mojokerto.
- [8] Wahyu Dwi Mustofa dan Melwin Syafrizal. 2015. "Perancangan Aplikasi Pencarian Bengkel di Kabupaten Gunungkidul Menggunakan Global Positioning Sistem (GPS) Berbasis Android" Jurnal Teknik Informatika, Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM Yogyakarta.
- [9] Wiwik Anisiyah, Fahrul Agus dan Hamdani. 2011. "Penentuan Rute Terpendek Menuju Pusat Kesehatan Menggunakan Metode Dijkstra Berbasis Webgis" Jurnal Informatika Mulawarman Vol. 6 No. 3, Universitas Mulawarman.

## DAFTAR LAMPIRAN

### Lampiran 1 Lembar Bimbingan Skripsi I



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
POLITEKNIK NEGERI MALANG  
JURUSAN TEKNOLOGI INFORMASI  
PROGRAM STUDI TEKNIK INFORMATIKA  
JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



NO SKRIPSI: 67

### LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : Rancang Bangun Aplikasi Sistem Informasi Geografis Penentuan Rute Terpendek  
Menuju Tambal Ban dan Bengkel Menggunakan Metode Dijkstra Berbasis Webservice

Nama : Fuad Ali Dawami

NIM : 1341180126

1-178

No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	24 maret 17	konsepsi data	Sufi	Ridwan
2.	12 april 17	metode pemambilan data	Sufi	Ridwan
3.	21 april 17	bab 1-3 revisi	Sufi	Ridwan
4.		bab 3 revisi	Sufi	Ridwan
5.		bab 4 revisi	Sufi	Ridwan
6.		bab 5 revisi	Sufi	Ridwan
7.		bab 6 revisi	Sufi	Ridwan
8.		bab 6 revisi	Sufi	Ridwan
9.		revisi program	Sufi	Ridwan
10.		revisi program	Sufi	Ridwan
11.		revisi program	Sufi	Ridwan
12.		revisi program	Sufi	Ridwan
13.		revisi program	Sufi	Ridwan
14.				
15.				
16.				
17.				
18.				
19.				

Malang, .....  
Dosen Pembimbing,

Budi Harijanto, ST., M.MKom  
NIP. 196201051990031002

## Lampiran 2 Lembar Bimbingan Skripsi II



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
 POLITEKNIK NEGERI MALANG  
 JURUSAN TEKNOLOGI INFORMASI  
 PROGRAM STUDI TEKNIK INFORMATIKA  
 JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



NO SKRIPSI: 67

### LEMBAR BIMBINGAN SKRIPSI 2016/2017

JUDUL : Rancang Bangun Aplikasi Sistem Informasi Geografis Penentuan Rute Terpendek  
 Menuju Tambal Ban dan Bengkel Menggunakan Metode Dijkstra Berbasis Webservice

Nama : Fuad Ali Dawami

NIM : 1341180126

No.	Tanggal	Materi Bimbingan	Tanda Tangan	
			Mahasiswa	Dosen
1.	9 maret 17	konsultasi: fentang Data	Sud	
2.	24 maret 17	laporan Data jalan	Sud	
3.	12 April 17	Pembuatan admin	Sud	
4.	20 April 17	pembuatan fitur sistem	Sud	
5.	27 April 17	konsultasi metode	Sud	
6.	4 Juni 17	konsultasi metode	Sud	
7.		konsultasi program	Sud	
8.	"	program	Sud	
9.	"	program	Sud	
10.	"	bab 1 - 3	Sud	
11.	"	bab 4 - 5	Sud	
12.	"	bab 6	Sud	
13.				
14.				
15.				
16.				
17.				
18.				
19.				

Malang, .....  
 Dosen Pembimbing,

Usman Nurhasan, S.Kom., MT  
 NIP. 198609232015041001

### Lampiran 3 Lembar Revisi Skripsi I



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
 POLITEKNIK NEGERI MALANG  
 JURUSAN TEKNOLOGI INFORMASI  
 PROGRAM STUDI TEKNIK INFORMATIKA  
 JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



#### FORM REVISI SKRIPSI

No. Skripsi : 67

Nama Mahasiswa : Fuad Ali Dawami NIM : 1341180126  
 Tanggal Ujian : 30 Agustus 2017  
 Judul : Sistem Informasi Geografis Penentuan Rute Terpendek  
 Menuju Tambal Ban dan Bengkel Menggunakan Metode  
 Dijkstra Berbasis Webservice (Studi Kasus Kota Mojokerto)

NO	SARAN PERBAIKAN	PARAF
-	<ul style="list-style-type: none"> <li>- Studi kasus dibutuhkan contoh rute.</li> <li>- ada filter pencarian yg filter</li> <li>- Penalaran menggunakan dlm arah</li> <li>- Diketahui teori maneh di bagian pertama.</li> </ul>	{ } b-

Malang, 5 Sept 2017  
 Dosen Penguji,

(Ely Setyo Astuti)

#### FORM VERIFIKASI:

Laporan Akhir telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji	Ely Setyo Astuti	Mr.	5 Sept 2017
Pembimbing 1	Budi Harjanto	Harjanto	7 - Sep - 2017
Pembimbing 2	Usman Muhsinin, S.Kom	U	7 - Sep - 2017

## Lampiran 4 Lembar Revisi Skripsi II



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
 POLITEKNIK NEGERI MALANG  
 JURUSAN TEKNOLOGI INFORMASI  
 PROGRAM STUDI TEKNIK INFORMATIKA  
 JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



### FORM REVISI SKRIPSI

No. Skripsi : 67

Nama Mahasiswa : Fuad Ali Dawami NIM : 1341180126  
 Tanggal Ujian : 30 Agustus 2017  
 Judul : Sistem Informasi Geografis Penentuan Rute Terpendek  
 Menuju Tambal Ban dan Bengkel Menggunakan Metode  
 Dijkstra Berbasis Webservice (Studi Kasus Kota Mojokerto)

NO	SARAN PERBAIKAN	PARAF
	<p>- <i>E.R.D</i> <i>Pelos</i>  <i>Referensi / literatur perbaikan</i></p>	<p>) M</p>

Malang, .....

Dosen Penguji,

(*Flynn*, .....

### FORM VERIFIKASI:

Laporan Akhir telah diperbaiki sesuai dengan saran perbaikan dari dosen penguji.

PENGUJI/PEMBIMBING	NAMA	TTD	TANGGAL
Penguji	<i>Rhyma</i>	<i>Rhyma</i>	7-Sep-2017
Pembimbing 1	<i>Budi Hanjanto</i>	<i>Budi Hanjanto</i>	7-Sep-2017
Pembimbing 2	<i>Usman Murhsan S.Kom</i>	<i>Usman Murhsan S.Kom</i>	7-Sep-2017

## Lampiran 5 Form Verifikasi Abstrak Bahasa Inggris dan Tata Tulis Buku Skripsi



KEMENTERIAN RISET, TEKNOLOGI DAN PENDIDIKAN TINGGI  
**POLITEKNIK NEGERI MALANG**  
 JURUSAN TEKNOLOGI INFORMASI  
 PROGRAM STUDI TEKNIK INFORMATIKA  
 JL. Soekarno Hatta PO Box 04 Malang Telp. (0341) 404424 pes. 1122



No. Skripsi : 67

### FORM VERIFIKASI

#### ABSTRAK BAHASA INGGRIS DAN TATA TULIS BUKU SKRIPSI

Nama Mahasiswa 1 : Fuad Ali Dawami NIM : 1341180126

Tanggal Ujian : ..... 30 Agustus 2017 .....

Judul : Rancang Bangun Sistem Informasi Geografis Pencarian Rute Terpendek  
 Menuju Tambal Ban dan Bengkel Menggunakan Metode *Dijkstra* Berbasis  
*Webservice* (Studi Kasus Kota Mojokerto).

NO	BAGIAN YANG DIVERIFIKASI	NAMA VERIFIKATOR	TANGGAL VERIFIKASI	TTD
1	Abstrak Berbahasa Inggris	Atigah N. Asri	28 -09 -2017	
2	Tata Tulis Buku Skripsi	Budi Harjanto	3 -10 -2017	

## BIODATA PENULIS



Nama	: Fuad Ali Dawami
Tempat, Tanggal Lahir	: Bojonegoro, 28 Desember 1995
Alamat	: Dsn. Ngumpak, RT 01, RW 03, Ds. Jabon, Kec Mojoanyar, Mojokerto, Jawa Timur.
No HP	: +62856-5536-6360
Email	: <a href="mailto:fuadalidawami50@gmail.com">fuadalidawami50@gmail.com</a>

### RIWAYAT PENDIDIKAN

- 2002 – 2007 SD Negeri Magersari 1 Kota Mojokerto
- 2007 – 2010 SMP Negeri 5 Kota Mojokerto
- 2010 – 2013 SMAK Santo Thomas Aquino Kota Mojokerto
- 2013 – 2017 Politeknik Negeri Malang