

TUGAS AKHIR

RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK DENGAN MENGGUNAKAN FUNGSI *WAYPOINT* DAN METODE *HEURISTIK* *GREEDY* DI KOTA MATARAM BERBASIS *WEB*

Tugas Akhir
Untuk memenuhi sebagai persyaratan
mencapai derajat Sarjana S-1 Program Studi Teknik Informatika



OLEH :

ANANG NUGROHO

F1D013007

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MATARAM

2020

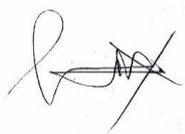
TUGAS AKHIR

RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS Pencarian Lokasi Penjualan Oleh – Oleh Khas Lombok Dengan Menggunakan Fungsi Waypoint dan Metode *Heuristik Greedy* di Kota Mataram Berbasis Web

Oleh :
Anang Nugroho
F1D 013 007

Telah diperiksa dan disetujui oleh Tim Pembimbing :

1. Pembimbing Utama



Tanggal :6/Juni/2020

Dr.Eng. I Gde Putu Wirarama Wedashwara Wirawan, ST., MT.
NIP. 19840919 201803 1001

2. Pembimbing Pendamping



Tanggal : 6/Juni/2020

Ariyan Zubaidi, S.Kom., M.T.
NIP. 198609132015041001

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Teknik
Universitas Mataram



Prof. Dr. Eng. I Gede Pasek Suta Wijaya ST.,MT.
NIP.197311302000031001

TUGAS AKHIR

RANCANG BANGUN SISTEM INFORMASI GEOGRAFIS PENCARIAN LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK DENGAN MENGUNAKAN FUNGSI WAYPOINT DAN METODE *HEURISTIK GREEDY* DI KOTA MATARAM BERBASIS *WEB*

Oleh :
Anang Nugroho
F1D 013 007

Telah Diujikan di depan Penguji
Pada tanggal 03 Juni 2020
dan dinyatakan telah memenuhi syarat mencapai serajat Sarjana S-1
Program Studi Teknik Informatika

Susunan Tim Penguji

1. Penguji 1



Andy Hidayat Jatmika, S.T., M.Kom.
NIP. 19831209 201212 1 001

Tanggal : 6/Juni/2020

2. Penguji 2



Ahmad Zafrullah Mardiansyah, ST., M.Eng.

Tanggal : 6/Juni/2020

3. Penguji 3



Fitri Bimantoro, ST., M.Kom.
NIP : 19860622 201504 1 002

Tanggal : 6/Juni/2020

Mataram, 6 Juni 2020
Dekan Fakultas Teknik
Universitas Mataram



Akmaluddin, S.T., M.Sc., (Eng)., Ph.D.
NIP. 19861231 199412 1 001

PERYATAAN KEASLIAN TUGAS AKHIR

Saya yang bertanda tangan di bawah ini menyatakan bahwa dalam Skripsi ini tidak terdapat karya yang pernah di ajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.

Mataram, Juni
2020

Anang Nugroho

PRAKATA

Puji syukur penulis panjatkan ke hadirat Tuhan Yang Maha Pengasih dan Penyayang atas segala berkat, bimbingan dan karunia-Nya, sehingga penulis dapat menyelesaikan penyusunan tugas akhir dengan judul “Rancang Bangun Sistem Informasi Geografis Pencarian Lokasi Penjualan Oleh – Oleh Khas Lombok Dengan Menggunakan Fungsi *Waypoint* Dan Metode *Heuristik Greedy* Di Kota Mataram Berbasis *Web*”.

Tugas Akhir ini dilaksanakan di Laboratorium Komputer, Program Studi Teknik Informatika Universitas Mataram. Tujuan dari penelitian ini adalah membuat aplikasi untuk pencarian lokasi dari toko oleh – oleh khas Lombok dikota Mataram.

Dalam penulisan skripsi ini tentunya tidak lepas dari kekurangan, baik aspek kualitas maupun aspek kuantitas dari materi penelitian yang disajikan. Semua ini didasarkan dari keterbatasan yang dimiliki penulis. Penulis menyadari bahwa skripsi ini jauh dari sempurna sehingga penulis membutuhkan kritik dan saran yang bersifat membangun untuk kemajuan teknologi di masa yang akan datang. Akhir kata semoga tidaklah terlampau berlebihan, bila penulis berharap agar karya ini dapat bermanfaat bagi pembaca.

Mataram Juni 2020

penulis

UCAPAN TERIMA KASIH

Tugas Akhir ini dapat diselesaikan berkat bimbingan dan dukungan ilmiah maupun materiil dari berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih yang setulus-tulusnya kepada :

Bapak Dr.Eng. I Gde Putu Wirarama Wedashwara Wirawan, ST., MT.. selaku dosen pembimbing pertama yang telah memberikan bimbingan dan arahan kepada penulis selama penyusunan Tugas Akhir ini, sehingga dapat terselesaikan dengan baik.

1. Bapak Ariyan Zubaidi, S.Kom., M.T.. selaku dosen pembimbing pendamping yang telah memberikan bimbingan dan arahan selama menyusun Tugas Akhir ini.
2. Bapak Andy Hidayat Jatmika, S.T., M.Kom, Bapak Ahmad Zafrullah Mardiansyah, S.T.,M.Eng, Bapak Fitri Bimantoro, ST., M.Kom. selaku dosen penguji yang memberikan kritik dan saran dalam sehingga Tugas Akhir ini dapat dibuat dengan baik.
3. Keluarga terutama ibu yang selalu memberikan nasihat dan juga dukungan secara penuh baik berupa materi maupun secara moril kepada penulis.
4. Mahasiswa Teknik Informatika, Boy Arista, Yoga Permana Adi P. dan mahasiswa lain yang telah membantu penulis dalam penyusunan tugas akhir.
5. Semua pihak yang tidak dapat penulis sebutkan satu persatu, yang telah memberikan bimbingan kepada penulis dalam menyelesaikan Tugas Akhir ini. Semoga Tuhan Yang Maha Esa memberikan imbalan yang setimpal atas bantuan yang diberikan kepada penulis.

DAFTAR ISI

HALAMAN JUDUL	
LEMBAR PENGESAHAN	i
PERNYATAAN KEASLIAN TUGAS AKHIR	iii
PRAKATA	iv
UCAPAN TERIMA KASIH	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	xi
ABSTRAK	xii
ABSTRACT	xiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	3
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian.....	4
1.6 Sistematika Penulisan.....	4
BAB II TINJAUAN PUSTAKA DAN LANDASAN TEORI	5
2.1 Tinjauan Pustaka	5
2.2 Landasan Teori	7
2.2.1 Algoritma <i>Heuristic Greedy</i>	7
2.2.2 Sistem Informasi	12
2.2.3 Definisi Sistem Informasi Geografis.....	12
2.2.4 Google Maps API.....	13
2.2.5 <i>Geolocation</i>	13
2.2.6 <i>Geocoding</i>	14
2.2.7 <i>Waypoint</i>	14
2.2.8 <i>Entity Relationship Diagram (ERD)</i>	15
2.2.9 <i>Unified Modelling Language (UML)</i>	16
2.2.10 <i>CodeIgniter</i>	20
2.2.11 XAMPP	21

2.2.12 Metode Pengembangan <i>Waterfall</i>	21
BAB III METODE PERANCANGAN	23
3.1 Alat dan Bahan	23
3.2 Metode Pengembangan	24
3.2.1 Pendefinisian Kebutuhan Sistem	24
3.2.1.1 Pengumpulan Data	25
3.2.1.2 Penjabaran Kebutuhan Sistem	25
3.2.1.3 Penjabaran Kebutuhan <i>End User</i>	25
3.2.2 Perancangan Sistem	26
3.2.2.1 Perancangan <i>Behavior</i>	26
3.2.2.2 Perancangan <i>Class</i>	27
3.2.2.3 Perancangan Aktifitas Aktor	29
3.2.2.4 Perancangan <i>Deployment</i>	37
3.2.2.5 Perancangan <i>Database</i>	37
3.2.2.6 Perancangan <i>Layout Website</i>	40
3.2.3 <i>Coding</i> atau Implementasi	49
3.2.4 Pengujian	49
3.2.5 Evaluasi	50
3.2.5 Dokumentasi	50
BAB IV HASIL DAN PEMBAHASAN	51
4.1 Alat dan Bahan	51
4.1.1 Pendefinisian Kebutuhan Sistem	51
4.1.2 Implementasi <i>Class</i> Sistem	53
4.1.3 Implementasi <i>Interface</i> Sistem	55
4.2 Pengujian Sistem	68
4.2.1 Hasil Pengujian <i>Black Box</i>	68
4.2.2 Hasil Pengujian <i>White Box</i>	75
4.2.3 Hasil Pengujian Kuesioner dengan Metode (MOS)	77
BAB V PENUTUP	81
5.1 Kesimpulan	81
5.2 Saran	81
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 2.1 Graph Berarah awal	8
Gambar 2.2 Graph Berarah langkah pertama	9
Gambar 2.3 Graph Berarah langkah kedua.....	9
Gambar 2.4 Graph Berarah langkah ketiga	10
Gambar 2.5 Graph Berarah penerapan awal mula.....	10
Gambar 2.6 Graph Berarah penerapan pertama.....	11
Gambar 2.7 Graph Berarah penerapan kedua.....	11
Gambar 2.8 Graph Berarah penerapan ketiga.....	12
Gambar 2.9 Simbol Entitas.....	15
Gambar 2.10 Simbol Relasi.....	15
Gambar 2.11 Simbol Atribut	15
Gambar 2.12 Model Pengembangan <i>Waterfall</i>	22
Gambar 3.1 <i>Flowchart</i> Pengembangan	24
Gambar 3.2 <i>Usecase</i> Diagram Pencarian lokasi oleh – oleh terdekat	27
Gambar 3.3 <i>Class</i> Diagram sistem	28
Gambar 3.4 <i>Sequence</i> Diagram <i>Login</i>	29
Gambar 3.5 <i>Sequence</i> Diagram Mengelola data <i>admin</i>	30
Gambar 3.6 <i>Sequence</i> Diagram Mengelola data lokasi	31
Gambar 3.7 <i>Sequence</i> Diagram Mengelola data barang.....	32
Gambar 3.8 <i>Sequence</i> Diagram Mengelola berita	33
Gambar 3.9 <i>Sequence</i> Diagram Melihat Berita	34
Gambar 3.10 <i>Sequence</i> Diagram cari toko	35
Gambar 3.11 <i>Sequence</i> Diagram cari rute	35
Gambar 3.12 <i>Sequence</i> Diagram daftar toko	36
Gambar 3.13 <i>Sequence</i> Diagram <i>Logout</i>	37
Gambar 3.14 <i>Deployment</i> Diagram	37
Gambar 3.15 <i>Entity Relationship</i> Diagram.....	38
Gambar 3.16 Halaman <i>Home Admin</i>	41
Gambar 3.17 Halaman Tambah Admin.....	41
Gambar 3.18 Halaman Daftar Admin.....	42
Gambar 3.19 Halaman Daftar Toko.	42

Gambar 3.20 Halaman Tambah Toko.....	43
Gambar 3.21 Halaman Daftar Barang	43
Gambar 3.22 Halaman Tambah Barang	44
Gambar 3.23 Halaman Kelola Berita.....	45
Gambar 3.24 Halaman Cari Rute.....	45
Gambar 3.25 Halaman Cari Toko.....	46
Gambar 3.26 Halaman Daftar Toko	47
Gambar 3.27 Halaman Berita	47
Gambar 3.28 Halaman Tambah Berita	48
Gambar 3.29 Halaman Publikasi Berita	49
Gambar 4.1 Struktur <i>Database</i> “oleh_oleh”	51
Gambar 4.2. Struktur Tabel barang	52
Gambar 4.3. Struktur Tabel berita	52
Gambar 4.4. Struktur Kategori	53
Gambar 4.5.Struktur Tabel Toko	53
Gambar 4.6.Struktur Tabel User.....	53
Gambar 4.7 Class model.....	54
Gambar 4.8 Class view	55
Gambar 4.9 Class controller	55
Gambar 4.10 Halaman Home	56
Gambar 4.11. Halaman Login.....	57
Gambar 4.12 Halaman Kelola User.....	57
Gambar 4.13. Halaman Import data user.....	58
Gambar 4.14. Halaman Edit User.....	58
Gambar 4.15 Halaman Kelola Kategori	59
Gambar 4.16. Halaman Import data user.....	59
Gambar 4.17. Halaman Edit Kategori	59
Gambar 4.18. Halaman Kelola Toko	60
Gambar 4.19 Halaman Tambah Toko.....	60
Gambar 4.20 Halaman Edit data Toko	61
Gambar 4.21 Halaman profile	62
Gambar 4.22. Halaman Kelola Barang.....	62
Gambar 4.23 Halaman Tambah Barang	63

Gambar 4.24 Halaman Edit data Barang	63
Gambar 4.25. Halaman Kelola Berita.....	64
Gambar 4.26 Halaman Tambah berita	64
Gambar 4.27 Halaman Edit data Berita	65
Gambar 4.28. Halaman Kelola Barang	65
Gambar 4.30 Halaman Hasil Cari Rute	66
Gambar 4.31. Halaman Cari Toko	66
Gambar 4.32. Halaman Hasil Cari Toko.	67
Gambar 4.33 Halaman Berita	67
Gambar 4.34. Gambar Hasil fungsi pencarian rute	75
Gambar 4.35. Grafik persentase jawaban kuisisioner untuk setiap pertanyaan	78
Gambar 4.36. Grafik persentase jawaban kuisisioner untuk setiap pertanyaan	78

DAFTAR TABEL

Tabel 2.1 Tabel <i>Geolocation</i>	13
Tabel 2.2 Tabel Simbol <i>Class</i> Diagram.....	16
Tabel 2.3 Tabel Simbol <i>Usecase</i> Diagram.....	17
Tabel 2.4 Tabel Simbol <i>Activity</i> Diagram	18
Tabel 2.5 Tabel Simbol <i>Sequence</i> Diagram.....	19
Tabel 3.1 Tabel Admin	38
Tabel 3.2 Tabel Berita	39
Tabel 3.3 Tabel Barang.....	39
Tabel 3.4 Tabel Kategori	39
Tabel 3.5 Tabel Toko.....	39
Tabel 4.1 Pengujian black box pada fungsi login admin.....	68
Tabel 4.2 Pengujian black box Kelola User	69
Tabel 4.3 Pengujian black box pada fungsi Kelola Toko	70
Tabel 4.4 Pengujian black box pada fungsi Kelola Kategori	70
Tabel 4.5 Pengujian black box pada fungsi Edit Profile.....	71
Tabel 4.6 Pengujian black box Kelola Barang	72
Tabel 4.7 Pengujian black box pada fungsi Kelola Berita.....	72
Tabel 4.8 Pengujian black box pada fungsi Kelola Kategori	73
Tabel 4.9 Pengujian black box pada fungsi Cari Rute.....	74
Tabel 4.10 Pengujian black box pada fungsi Cari Toko.....	74
Table 4.11 Tabel Ketetangaan Graf Masalah	75
Table 4.12 Hasil pengujian kuisisioner dari sistem (sisi User)	77
Table 4.13 Hasil pengujian kuisisioner dari sistem (sisi Admin).....	78

ABSTRAK

Rancang Bangun Sistem Informasi Geografis Lokasi Pencarian Rute Tercepat Oleh Wombok Khusus Dengan Menggunakan Fungsi Waypoint Dan Metode Heuristik Greedy Di Kota Mataram Berbasis Web adalah aplikasi pemetaan peta Google berbasis web yang memetakan lokasi toko-toko oleh - oleh dan kategori toko yang berlokasi di kota Mataram. Tujuan pembuatan aplikasi ini adalah untuk memudahkan wisatawan domestik dan asing untuk mencari toko-toko souvenir di Lombok, di mana wisatawan dapat mencari 1-3 kategori toko yang ada dan mencari rute terdekat dari posisi pengguna berdasarkan kategori pencarian berdasarkan kategori. Sistem ini menggunakan metode heuristic greedy untuk pencarian rute, dimana geolocation digunakan untuk mendapatkan lokasi user yang digunakan sebagai lokasi awal untuk pencarian rute dengan toko oleh – oleh dan dijadikan rute dengan menggunakan fungsi waypoint yang digunakan untuk membuat rute dengan lokasi tujuan yang lebih dari satu. Penulis menggunakan blackbox, whitebox dan MOS (Mean Opinion Score) sebagai pengujian sistem. Hasil blackbox dan pengujian fungsi dalam sistem telah berjalan dengan baik tanpa ada kesalahan, hasil tes whitebox menunjukkan bahwa hasil pencarian sistem cocok dengan hasil pencarian secara manual, kami mendapat 84% setuju dan 16% pendapat netral menggunakan MOS dari sisi pengguna dan kami mendapat 74% setuju dan 26% pendapat netral dari sisi admin .

Kata kunci: CodeIgniter, *waypoint*, *Geolocation*, *google maps*, *Heuristic*, *Greedy*.

ABSTACT

Design Of Geographic Information System Fastest Route Search Location By Wombok Specialized By Using Waypoint Function And Greedy Heuristic Method In Mataram City Web-Based is a web-based Google map mapping application that maps the location of souvenir shops and shop categories especially those located in the city of Mataram . The purpose of making this application is to make it easier for domestic and foreign tourists to search for souvenir shops in Lombok, where tourists can search 1-3 categories of existing stores and search for the closest route from the user's position based on the search category. This system uses the heuristic greedy method for route search, where geolocation is used to get the user's location which is used as a starting location for route search by souvenirs and used as a route by using the waypoint function used to create routes with more than one destination location. We use blackbox , whitebox and MOS (Mean Opinion Score) as system testing. Result of blackbox and testing the function in the system has been running nicely without having any errors, whitebox test results show that the system search results match the search results manually we got 84% agreed and 16% neutral opinions using MOS from the user side and we got 74% agreed and 26% neutral opinions from the side admin

Keyword: CodeIgniter, waypoint, Geolocation, google maps, Heuristic, Greedy.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi informasi mengalami kemajuan yang sangat pesat sehingga memudahkan pengguna untuk mendapatkan informasi. Dengan semakin berkembangnya teknologi informasi, banyak instansi dan masyarakat maju yang telah memanfaatkan teknologi informasi untuk memperoleh informasi. Salah satu perkembangan teknologi yang telah berkembang pesat, yaitu teknologi internet. Berkat kemajuan teknologi internet kita diberikan *alternative* dalam memberikan suatu informasi, salah satunya adalah sistem informasi berbasis *web*. Selain memberikan manfaat dalam proses pengolahan informasi, sistem informasi berbasis *web* memberikan tampilan menarik, *user friendly* dan dapat diakses secara luas dengan internet, sehingga penyebaran informasi dapat berlangsung lebih cepat.

Saat ini pulau Lombok merupakan destinasi pariwisata yang diminati oleh banyak wisatawan. Banyak wisatawan domestik ataupun mancanegara memilih pulau Lombok sebagai tujuan wisata, berdasarkan data kunjungan wisatawan dari dinas pariwisata pada tahun 2019 Lombok memiliki jumlah kunjungan yang selalu bertambah setiap bulannya, melihat hal itu banyak penduduk lokal yang memulai usaha dengan menjual produk khas pulau Lombok untuk dijadikan oleh – oleh, mulai dari jajanan, pakaian, dan juga kesenian khas Lombok.

Kebutuhan akan informasi lokasi penjualan oleh – oleh khas Lombok tentunya dibutuhkan oleh wisatawan yang ingin membawa cinderamata khas pulau Lombok, akan tetapi kebutuhan akan informasi tentang lokasi dari toko oleh – oleh masih belum ada dan hanya terbatas pada sosial media dimana wisatawan hanya dapat mengetahui lokasi akan toko oleh – oleh yang direkomendasikan oleh wisatawan lainnya ataupun toko oleh – oleh yang melakukan promosi pada sosial media, sehingga wisatawan hanya mengetahui lokasi toko oleh – oleh tersebut walaupun lokasi toko tersebut jauh dari tempat wisatawan itu berada dan terkadang cinderamata yang diinginkan oleh wisatawan berbeda dengan informasi yang ada. Wisatawan juga mendapatkan informasi melalui *google maps* akan tetapi terdapat kekurangan dimana *google maps* dimana *google maps* hanya menampilkan toko oleh – oleh dari seluruh data yang ada dalam database dalam radius tertentu dari

posisi pengguna sehingga pengguna masih harus mencari toko oleh - oleh yang diinginkan pengguna serta memasukan titik kordinat dari tujuan toko secara manual untuk mendapat rute menuju lokasi toko yang diinginkan.

Oleh karena itu, diperlukan suatu sistem informasi yang dapat memberikan informasi berupa lokasi penjualan oleh – oleh khas Lombok sehingga dapat membantu wisatawan domestik maupun wisatawan mancanegara untuk mendapatkan informasi tentang lokasi penjualan oleh – oleh disekitarnya. Diperlukan juga fitur geografis yang mampu menggambarkan rute perjalanan menuju lokasi yang dapat memudahkan wisatawan untuk mengetahui arah menuju lokasi yang diinginkan .

Untuk menyelesaikan permasalahan tersebut, maka penulis membuat suatu sistem informasi geografis yang bernama **“Rancang Bangun Sistem Informasi Geografis Pencarian Lokasi Penjualan Oleh – oleh khas Lombok dengan menggunakan fungsi WayPoint dan metode Heuristik Greedy Di Kota Mataram Berbasis Web”**. Sistem ini merupakan aplikasi pemetaan lokasi toko oleh – oleh dan pencarian lokasi toko oleh – oleh menggunakan metode *heuristik Greedy* yang digunakan untuk menentukan lokasi toko terdekat. Pada sistem ini akan mengambil titik kordinat pengguna dengan menggunakan fungsi *geolocation* untuk digunakan sebagai titik awal pencarian toko dan pengguna akan memasukkan jenis kategori toko jajanan, kesenian, dan pakaian sesuai dengan kebutuhan maka sistem nantinya akan menampilkan rute dengan tujuan satu lokasi oleh – oleh atau lebih dari satu lokasi oleh - oleh sesuai dengan kategori yang dimasukkan pengguna. Sistem ini membutuhkan data dari lokasi oleh – oleh yang ada yang di kota Mataram sehingga pengumpulan data menggunakan cara observasi menuju lokasi untuk mendapatkan data lokasi serta data toko pada setiap toko oleh – oleh dikota Mataram.

1.2 Rumusan Masalah

Berdasarkan uraian latar belakang yang ada maka dirumuskan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana menganalisis dan menentukan kebutuhan, serta merancang sistem informasi geografis berbasis *website* yang dapat menggambarkan lokasi dan menentukan rute dari posisi *user* menuju lokasi penjualan oleh – oleh khas Lombok di Kota Mataram?

2. Bagaimana membuat sistem informasi geografis yang dapat memberikan informasi tentang berbagai lokasi penjualan oleh – oleh khas Lombok di Kota Mataram ?
3. Bagaimana mengimplementasikan metode *heuristik greedy* untuk menentukan rute tercepat sesuai dengan pilihan pengguna pada sistem?
4. Bagaimana cara menampilkan hasil dari metode *heuristic greedy* menjadi tampilan rute pada sistem?

1.3 Batasan Masalah

Batasan-batasan masalah yang digunakan dalam tugas akhir ini adalah sebagai berikut:

1. Daerah penelitian meliputi wilayah Kota Mataram.
2. Sistem dirancang dan dibangun dengan berbasis *web*.
3. Faktor yang diamati ditinjau dari lokasi dan jarak rute dari posisi pengguna menuju lokasi yang ingin dituju.
4. Penentuan rute pada aplikasi menggunakan metode *heuristic greedy* untuk mengeliminasi rute perjalanan sehingga didapatkan rute perjalanan yang tercepat.

1.4 Tujuan Penelitian

Tujuan dari pembuatan tugas akhir ini adalah sebagai berikut:

Membuat sistem informasi geografis berbasis *web* sebagai media pemetaan yang dapat:

1. Membuat sistem informasi geografis yang dapat memetakan titik lokasi penjualan oleh – oleh khas Lombok di Kota Mataram.
2. Mampu memberikan informasi lokasi dari tempat penjualan oleh – oleh khas Lombok di Kota Mataram yang berada didekat pengguna.
3. Membuat informasi berbasis *web* yang dapat digunakan untuk mencari lokasi toko oleh – oleh berdasarkan kategori yang diinginkan.
4. Mampu memberikan rute tercepat menuju lokasi dari toko oleh – oleh yang diinginkan pengguna.

1.5 Manfaat Penelitian

Manfaat dari penulisan tugas akhir ini adalah sebagai berikut:

1. Dapat memberikan informasi mengenai lokasi penjualan oleh – oleh khas Lombok secara cepat dan tepat.
2. Dapat menggambarkan lokasi penjualan oleh – oleh di Kota Mataram serta dapat mencari rute tercepat
3. Mempermudah pengguna atau masyarakat dalam hal pencarian lokasi penjualan oleh – oleh di Kota Mataram.
4. Dapat digunakan untuk mempromosikan UKM bisnis Kota Mataram dalam hal oleh – oleh khas Lombok.

1.6 Sistematika Penulisan

Laporan tugas akhir ini ditulis dengan atika penulisan sebagai berikut:

BAB I : PENDAHULUAN

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan, batasan masalah, metodologi penelitian serta atika penulisan yang digunakan untuk menyusun laporan tugas akhir.

BAB II : TINJAUAN PUSTAKA DAN DASAR TEORI

Bab ini berisi tentang teori-teori yang berkaitan langsung dengan judul yang diangkat yang mendukung pembahasan dan permasalahan. Teori-teori yang mendukung laporan penelitian ini bersumber dari buku, jurnal, *e-book*, dan dari situs-situs web resmi yang berkaitan dari pembahasan dan permasalahan yang diangkat.

BAB III : METODE PERANCANGAN

Pada bab ini berisi tentang alat dan bahan, jalannya perancangan sistem dengan membuat UML berupa *use case diagram*, *class diagram*, dan *sequence diagram*. Serta teknik pengujian yang digunakan yaitu *black box*, dan *white box*.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini memuat tentang analisa dan pembahasan hasil penelitian yang telah dilakukan.

BAB V : KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dan saran-saran berdasarkan hasil pembahasan yang diperoleh.

BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Tinjauan Pustaka

Pembuatan sistem informasi geografis serta penelitian tentang pencarian lokasi dan rute terdekat telah banyak dilakukan, adapun penelitian tentang pencarian lokasi dan rute terdekat yang dibuat oleh peneliti terdahulu diantaranya sebagai berikut:

Pada penelitian [1] merupakan aplikasi web-*GIS* yang menampilkan informasi kuliner terdekat dengan menggunakan fungsi *haversine* untuk mencari lokasi kuliner terdekat dengan radius yang telah ditentukan pengguna dan dengan menggunakan lokasi pengguna sebagai center untuk pencarian tersebut. Sehingga pengguna dapat mengetahui lokasi kuliner terdekat dari titik lokasi pengguna dalam radius tertentu.

Pada penelitian [2] dilakukan penelitian untuk mencari rute terdekat untuk menentukan jalur terpendek pengiriman barang ke kantor pos-kantor pos cabang medan dengan menggunakan metode *heuristik A**, sehingga memudahkan petugas atau kurir kantor pos untuk menentukan jalur mana yang terpendek untuk mengantarkan barang akan dikirim.

Pada penelitian [3] mengimplementasikan algoritma *heuristic Greedy* untuk optimasi kunjungan tempat wisata. Algoritma ini bekerja dengan mencari jalur terpendek antara setiap node dan menempatkannya pada solusi yang disediakan atau bisa dituliskan dengan notasi big-O, $O(n^2 \log^2(n))$. Dalam kasus untuk proyek akhir ini, akan diimplementasikan TSP (*Travelling Salesman Problem*) didalam algoritma *Greedy*. Dimana untuk kasus TSP memiliki persyaratan bahwa semua tujuan wisata akan dikunjungi semuanya, dan setiap tempat wisata hanya sekali dikunjungi.

Pada penelitian [4] menggunakan algoritma *Bellman-Ford*, dimana ketika *user* membuka aplikasi ini *user* dapat langsung melihat lokasi UMKM yang tersebar di Kota Medan kemudian ketika *user* sudah memilih lokasi UMKM yang ingin dituju, maka algoritma Bellman-Ford akan memproses permintaan *user* dan mencari 3 rute terdekat menuju UMKM yang ingin dituju *user* dengan menampilkan Google Maps disertai dengan garis yang menghubungkan antara lokasi *user* dengan lokasi UMKM sebagai

outputnya. Dengan hasil pengujian sistem bahwa perhitungan jarak terpendek sangat bergantung kepada relasi titik yang telah di-input ke dalam database. Jika salah satu titik menuju UMKM tersebut tidak terhubung maka tidak ada rute yang akan tampil pada sistem.

Pada penelitian [5] dilakukan penelitian untuk mencari rute terpendek menuju POM bensin terdekat di kota Semarang dengan menggunakan metode Dijkstra, sehingga memudahkan pengguna untuk mencari POM bensin terdekat. Dengan hasil pengujian sistem, maka diperoleh hasil pencarian jalur terpendek menggunakan algoritma Dijkstra berbasis sistem informasi geografis (GIS) tersebut dinyatakan berhasil dari peneliti yang paling tepat untuk menentukan jalur terpendek.

Pada penelitian [6] dilakukan penelitian *web GIS* yang mampu melakukan pencarian dan menyajikan informasi profil dari setiap objek sarana kesehatan masyarakat dengan memanfaatkan algoritma *Depth first search* sistem juga menyediakan proses pencarian rute terdekat menuju sarana kesehatan masyarakat, dengan hasil Sistem pemetaan informasi lokasi sarana pelayanan kesehatan masyarakat ini, dapat menjawab kebutuhan akan informasi alamat Rumah sakit terdekat yang dibutuhkan oleh masyarakat dan khususnya bagi pasien yang bersifat *Urgent*.

Pada penelitian [7] dilakukan penelitian untuk mencari rute jalur distribusi minuman PT.COCA-COLA Kota Padang dengan menggunakan metode Dijkstra, sehingga memudahkan pengguna untuk menentukan jalur distribusi. Dengan hasil sistem ini mampu menyalurkan informasi untuk PT. Distribusi Coca Cola tentang jalur mana yang bisa ditempuh untuk mempercepat proses pendistribusian dengan mengikuti jalur rute terpendek menggunakan algoritma Dijkstra yang telah dihasilkan.

Pada penelitian [8] dilakukan penelitian untuk mencari rute terdekat menuju indekos pada Kelurahan Kandang Limun dan Beringin Raya dengan menggunakan metode Dijkstra, sehingga memudahkan pengguna untuk menentukan rute menuju indekos terdekat. Dengan hasil penilaian terhadap variabel tampilan dengan nilai rata-rata 3.31 dari 4, penilaian terhadap variabel Kinerja Sistem dengan nilai rata-rata 3.1875 dari 4 dan penilaian terhadap variabel kemudahan pengguna dengan nilai rata-rata 3.125 dari 4.

Pada penelitian [9] dilakukan penelitian untuk mencari rute terdekat menuju perusahaan mebel di kota Jepara dengan menggunakan metode A^* , sehingga memudahkan pengguna untuk mencari rute menuju perusahaan mebel terdekat. Dengan hasil Pengujian Pencarian Lokasi Mebel Menggunakan Metode *Black Box Testing* Sistem dapat menentukan rute terpendek dari titik awal menuju titik akhir dengan menggunakan Algoritma $A^*(Star)$ dengan nilai *valid* / benar.

Pada penelitian [10] dilakukan penelitian untuk mencari rute terdekat menuju Apotek di kota Kendari dengan menggunakan metode *Floyd-Warshall*, sehingga memudahkan masyarakat untuk mencari rute menuju Apotek terdekat. Dengan hasil Pengujian dengan menggunakan metode *white box* diperoleh hasil dengan tingkat akurasi 100 % dari perbandingan keluaran sistem dengan perhitungan manual.

2.2 Landasan Teori

Dasar teori tentang konsep-konsep yang digunakan dalam perancangan dan pembuatan Sistem Informasi Geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – oleh khas Lombok mencakup pembahasan tentang definisi sistem informasi, definisi sistem informasi geografis, Google Maps API, *Geolocation*, *Geocoding* dan materi yang mendukung perancangan sistem yang akan dijelaskan pada subbab selanjutnya.

2.2.1 Algoritma *Heuristic Greedy*

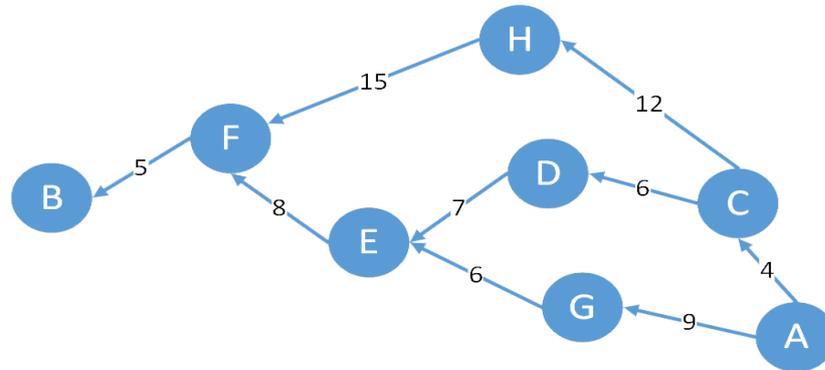
Algoritma *heuristic Greedy* adalah algoritma yang memecahkan masalah langkah per langkah, pada setiap langkah:

1. Mengambil pilihan yang terbaik yang dapat diperoleh pada saat itu tanpa memperhatikan konsekuensi ke depan.
2. Berharap bahwa dengan memilih optimum lokal pada setiap langkah akan berakhir dengan optimum global.

Skema umum Algoritma *heuristic Greedy*:

- a. Himpunan kandidat, himpunan ini berisi seluruh elemen pembentuk solusi
- b. Himpunan solusi, berisi kandidat-kandidat yang terpilih sebagai solusi persoalan
- c. Fungsi seleksi, fungsi yang pada setiap langkah memilih kandidat yang paling memungkinkan mencapai solusi optimal

- d. Fungsi kelayakan, fungsi yang memeriksa apakah suatu kandidat yang terpilih dapat memberikan solusi yang layak, yaitu kandidat bersama-sama dengan himpunan solusi yang sudah terbentuk tidak melanggar kendala yang ada.
- e. Fungsi objektif, fungsi yang memaksimumkan atau meminimumkan nilai solusi [11].



Gambar 2.1 Graph Berarah awal

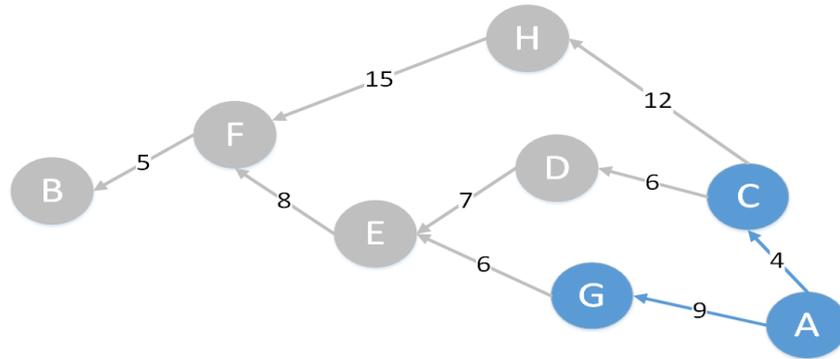
Graph Berarah dari Titik A ke B

Untuk mencari jarak terpendek dari A ke B, sebuah algoritma *heuristic Greedy* akan menjalankan langkah-langkah seperti berikut:

1. Kunjungi satu titik pada graph, dan ambil seluruh titik yang dapat dikunjungi dari titik sekarang.
2. Cari *local maximum* ke titik selanjutnya.
3. Tandai graph sekarang sebagai graph yang telah dikunjungi, dan pindah ke *local maximum* yang telah ditentukan.
4. Kembali ke langkah 1 sampai titik tujuan didapatkan.

Jika mengaplikasikan langkah-langkah di atas pada graph A ke B sebelumnya maka kita akan mendapatkan pergerakan seperti berikut:

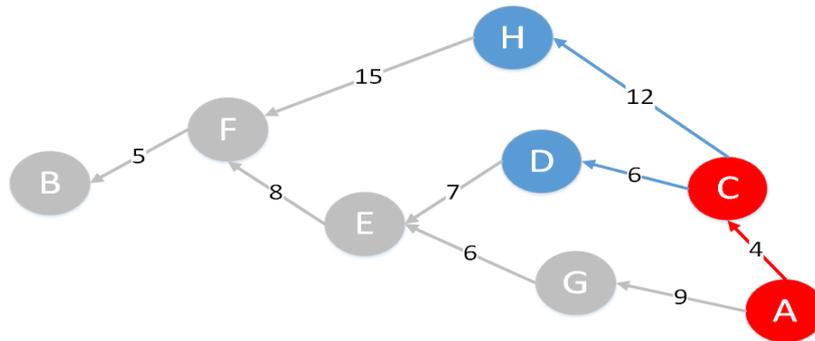
1. Mulai dari titik awal (A). Ambil seluruh titik yang dapat dikunjungi.



Gambar 2.2 Graph Berarah langkah pertama

Langkah Pertama *Greedy*

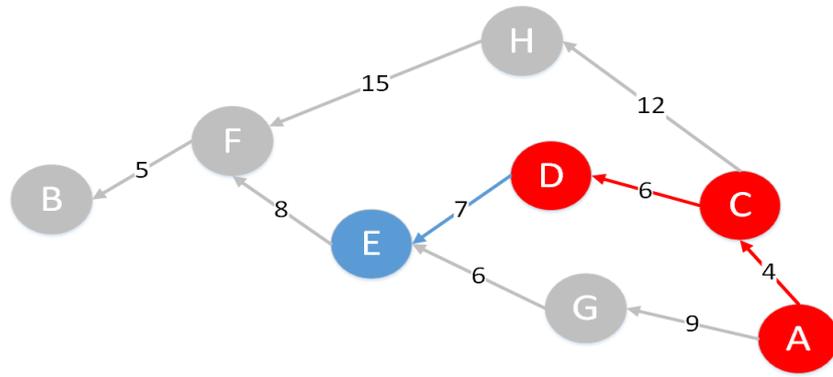
2. Local maximum adalah ke C, karena jarak ke C adalah yang paling dekat.
3. Tandai A sebagai titik yang telah dikunjungi, dan pindah ke C.
4. Ambil seluruh titik yang dapat dikunjungi dari C.



Gambar 2.3 Graph Berarah langkah kedua

Langkah Kedua *Greedy*

5. Local maximum adalah ke D, dengan jarak 6.
6. Tandai C sebagai titik yang telah dikunjungi, dan pindah ke D.



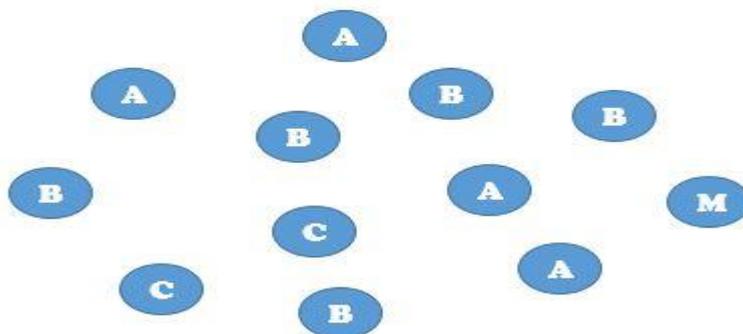
Gambar 2.4 Graph Berarah langkah ketiga

Langkah Ketiga *Greedy*

7. Local maximum adalah ke E, dengan jarak 7.
8. Tandai D sebagai titik yang telah dikunjungi, dan pindah ke E dan seterusnya.

Dengan menggunakan algoritma *greedy* pada graph di atas, hasil akhir yang akan didapatkan sebagai jarak terpendek adalah A-C-D-E-F-B. Hasil jarak terpendek yang didapatkan ini tidak tepat dengan jarak terpendek yang sebenarnya (A-G-E-F-B). Algoritma *greedy* memang tidak selamanya memberikan solusi yang optimal, dikarenakan pencarian *local maximum* pada setiap langkahnya, tanpa memperhatikan solusi secara keseluruhan. Gambar berikut memperlihatkan bagaimana algoritma *greedy* dapat memberikan solusi yang kurang optimal.

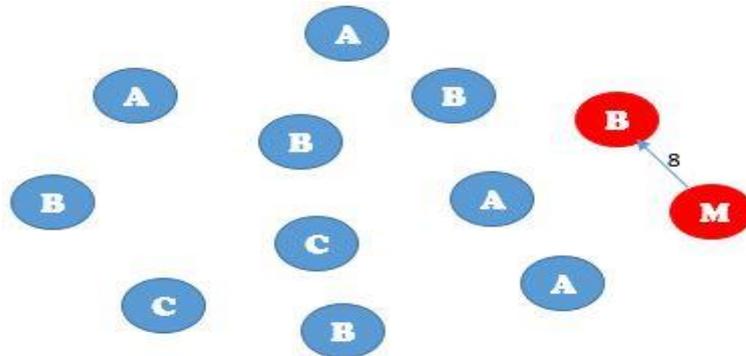
Penerapan metode Algoritma *Greedy* pada sistem dimana titik yang dikunjungi dibagi menjadi 3 kategori jajanan (A), pakaian (B) dan kesenian (C) dan titik awal (M):



Gambar 2.5 Graph Berarah penerapan awal mula

Langkah Pertama

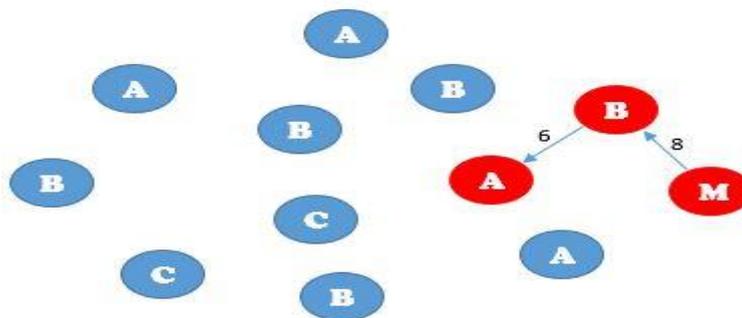
1. Mulai dari titik awal (M). Ambil seluruh titik yang dapat dikunjungi.



Gambar 2.6 Graph Berarah penerapan pertama

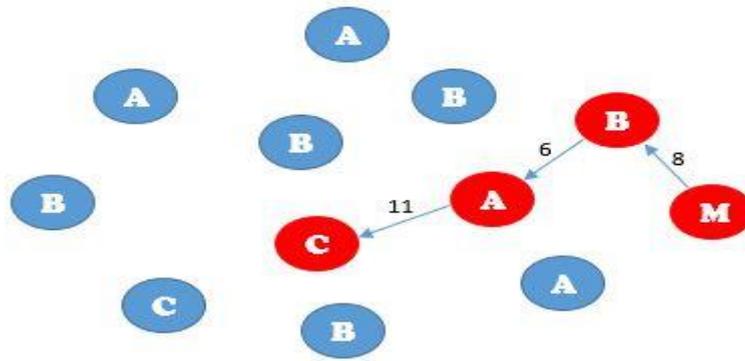
2. Local maximum adalah ke B, karena jarak ke B adalah yang paling dekat.
3. Tandai M sebagai titik yang telah dikunjungi, dan pindah ke B.
4. Ambil seluruh titik yang dapat dikunjungi dari B kecuali titik dengan kategori B.

Langkah Kedua



Gambar 2.7 Graph Berarah penerapan kedua

5. Local maximum adalah ke A, dengan jarak 6.
6. Tandai B sebagai titik yang telah dikunjungi, dan pindah ke A.
7. Ambil seluruh titik yang dapat dikunjungi dari A kecuali titik dengan kategori A dan B.



Gambar 2.8 Graph Berarah penerapan ketiga

Langkah Ketiga

8. Local maximum adalah ke C, dengan jarak 11.

Dengan menggunakan metode greedy pada graph di atas, hasil akhir yang akan didapatkan sebagai jarak terpendek adalah M-B-A-C dengan total jarak 25.

2.2.2 Sistem Informasi

Sistem Informasi adalah kumpulan dari elemen-elemen yang berinteraksi untuk mencapai suatu tujuan tertentu. Sistem informasi merupakan komponen yang saling bekerjasama untuk mengumpulkan, mengolah, menyimpan dan menyebarkan informasi untuk mendukung pengambilan keputusan, koordinasi, pengendalian, analisis masalah dan visualisasi dalam sebuah organisasi [12].

2.2.3 Sistem Informasi Geografis

Sistem Informasi geografis adalah suatu sistem berbasis komputer untuk menangkap, menyimpan, mengecek, mengintegrasikan, memanipulasi, dan menampilkan data dengan peta *digital* [13].

Sistem Informasi Geografis (SIG) mempunyai beberapa kelebihan serta kelemahan, adapun kelebihanannya menurut [14]:

1. Sistem Informasi geografis menggunakan data spasial maupun atribut secara terintegrasi.
2. Digunakan sebagai alat bantu interaktif yang menarik dalam usaha meningkatkan pemahaman mengenai konsep lokasi, ruang, kependudukan, dan unsur-unsur geografi yang ada dipermukaan bumi.
3. Memisahkan antara bentuk presentasi dan basis data.

4. Semua operasi SIG dapat dilakukan secara interaktif.
5. Bisa dengan mudah menghasilkan peta tematik.
6. Sangat membantu pekerjaan yang erat kaitannya dengan bidang spasial dan geoinformatika.

Adapun kekurangan dari sistem informasi geografis adalah:

1. Seringnya terdapat peta yang tidak diperbaharui.
2. Terkadang terdapat penggunaan skala yang tidak sesuai.

2.2.4 Google Maps API

Google Maps adalah layanan gratis Google yang menampilkan peta permukaan bumi secara *digital*. Google Maps API dapat digunakan untuk menambahkan berbagai fitur dalam Google maps yang terdapat dalam web yang dibangun. Google Maps API adalah *library* JavaScript. Menggunakan dan memprogram Google Maps API sangat mudah dan hanya dibutuhkan pengetahuan tentang HTML dan JavaScript, serta koneksi internet. Dengan menggunakan Google Maps API *programmer* dapat menghemat waktu dan biaya untuk membangun aplikasi peta *digital* yang handal, sehingga *programmer* dapat fokus pada data-data saja [15].

2.2.5 Geolocation

Geolocation adalah fitur spesifikasi HTML5 dan API. Hal ini digunakan untuk menemukan lokasi pengguna, meskipun akurasi lokasi tidak dijamin [16]. Berikut *interface* yang disediakan oleh HTML5 *geolocation* pada Tabel 2.1 dibawah ini:

Tabel 2.1 *Interface* yang disediakan HTML5 *Geolocation*.

No	<i>Interface</i>	Deskripsi
1	<code>Geolocation</code>	<code>Geolocation</code> merupakan objek utama dalam API dan digunakan untuk mendapatkan informasi lokasi perangkat dan pengguna.
2	<code>PositionOptions</code>	Fungsi <code>getCurrentPosition</code> dan <code>watchPosition</code> menerima <code>PositionOptions</code> objek sebagai argumen ketiga mereka.
3	<code>Position</code>	Sama seperti <code>Coordinates</code>
4	<code>Coordinates</code>	Koordinat dari lokasi geografis
5	<code>PositionError</code>	Informasi <i>error</i> jika lokasi geografis tidak dapat didapatkan.

2.2.6 Geocoding

Geocoding adalah proses untuk menentukan lokasi, biasanya dalam bentuk nilai koordinat ke alamat dengan membandingkan lokasi deskriptif elemen dalam alamat dengan bahan acuan. Alamat dapat ditampilkan dalam berbagai bentuk, mulai dari format alamat umum rumah, angka diikuti dengan nama jalan dan informasi deskripsi lokasi lainnya seperti zona pos. Pada dasarnya sebuah alamat termasuk jenis informasi yang membedakan lokasi [17].

Permintaan *Google Maps Geocoding API* harus dengan bentuk <https://maps.googleapis.com/maps/api/geocode/output?parameters> dalam hal ini, *output* bisa berupa salah satu nilai berikut:

- 3 Json (disarankan) menampilkan *output* dalam *JavaScript Object Notation* (JSON)
- 4 Xml menampilkan *output* berupa XML

Parameter yang diperlukan dalam permintaan *geocoding* dapat dilihat sebagai berikut:

- 3 *Address*, yaitu alamat jalan yang ingin di-*geocode*, dalam format yang digunakan oleh layanan pos nasional pada negara yang bersangkutan. Elemen alamat tambahan seperti nama bisnis dan unit, *suite* atau nomor lantai harus dihindari.
- 4 *Components*, yaitu filter komponen yang ingin didapatkan *geocode*-nya.
- 5 *Key*, yaitu kunci API aplikasi. Kunci ini mengidentifikasi aplikasi yang dibuat untuk keperluan manajemen kuota.

2.2.7 WayPoint

Waypoint adalah titik referensi dalam ruang fisik yang digunakan untuk tujuan navigasi. *Waypoint* merupakan koordinat yang mengidentifikasi titik dalam ruang fisik. Koordinat yang digunakan dapat bervariasi tergantung pada aplikasi. Untuk navigasi darat koordinat berupa bujur dan lintang, sedangkan untuk navigasi udara juga mencakup ketinggian. *Waypoint* biasanya digunakan untuk sistem navigasi pada *Global Positioning System* (GPS) dan jenis - jenis tertentu dari radio navigasi. *Waypoint* yang terletak di permukaan bumi biasanya didefinisikan dalam dua dimensi (misalnya, bujur dan lintang), sedangkan yang digunakan dalam atmosfer bumi atau di luar angkasa didefinisikan dalam setidaknya tiga dimensi atau empat jika waktu merupakan salah satu koordinat untuk

beberapa titik yang berada di luar Bumi. *Waypoint* ini digunakan untuk membantu menentukan jalur routing yang tak terlihat untuk navigasi [18].

2.2.8 Entity Relationship Diagram (ERD)

ERD didasarkan pada persepsi terhadap dunia nyata yang tersusun atas kumpulan objek-objek dasar yang disebut entitas dan hubungan antar objek [19].

1. Elemen-elemen Diagram ER

a. Entitas (*Entity*)

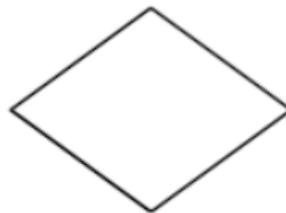
Entitas adalah sesuatu atau objek dalam dunia nyata yang dapat dibedakan dengan objek lainnya. Simbol dari entitas adalah seperti Gambar 2.9 dibawah ini:



Gambar 2.9 Simbol entitas

b. Relasi (*Relationship*)

Relasi adalah hubungan antara satu atau lebih entitas. Kardinalitas menentukan kejadian suatu entitas untuk satu kejadian pada entitas yang berhubungan. Contohnya Mahasiswa bisa mengambil banyak matakuliah. Adapun simbol dari relasi dalam diagram-ER adalah seperti pada Gambar 2.10.



Gambar 2.10 Simbol relasi.

c. Atribut (*Atribut*)

Atribut adalah ciri umum yang mengidentifikasi entitas secara unik dan dapat mendeskripsikan karakteristik dari suatu entitas. Simbol dari atribut adalah seperti pada Gambar 2.11.



Gambar 2.11 Simbol Atribut.

2.2.9 UML

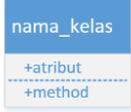
UML (*Unified Modelling Language*) adalah bahasa yang menggambarkan cara kerja maupun struktur dari suatu yang dibuat terlebih menggunakan konsep OOP (*Object Oriented Programming*). Dengan UML sebuah dapat dilihat dari struktur data yang diberikan, tingkah laku apa saja dari suatu objek hingga aktivitas–aktivitas yang ada di dalamnya. UML menyediakan sembilan jenis diagram, yang lain menyebutkan delapan karena ada beberapa diagram yang digabung misalnya diagram komunikasi, diagram urutan dan diagram pewaktuan digabung menjadi diagram interaksi. Namun demikian model-model itu dapat dikelompokkan berdasar sifatnya yaitu statis atau dinamis [20]. Jenis diagram itu antara lain:

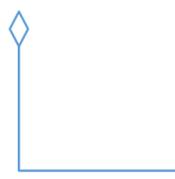
1. Class Diagram

Class Diagram adalah gambar yang menjelaskan struktur dari program yang akan dibuat menggunakan konsep OOP (*Object Oriented Programming*). *Class diagram* menggambarkan bagaimana objek pada dunia nyata digambarkan pada struktur yang biasa memiliki atribut dan *method*. Pada umumnya *class diagram* juga dapat menunjukkan hubungan antara relasi dari setiap *class* yang dibuat dan memperlihatkan objek – objek yang menyusun dari tersebut.

Pada *class diagram* terdapat beberapa simbol dan beberapa cara penulisan diantaranya sebagai berikut seperti pada Tabel 2.2:

Tabel 2.2 Tabel simbol *class diagram*

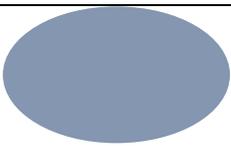
Simbol	Nama	Keterangan
	<i>Class</i>	<i>Class</i> adalah sebuah objek yang menggambarkan sebuah keadaan nyata. <i>Class</i> memiliki :Nama kelas , atribut , dan <i>method</i>
	<i>Association</i>	Garis yang menghubungkan antara dua kelas atau lebih dan menunjukkan bahwa kelas yang terhubung menunjukkan sebuah relasi seperti : <i>one-to-one</i> (1 – 1) , <i>one-to-many</i> (1 – M) , <i>many-to-many</i> (M – M)

	<i>Composition</i>	Menggambarkan sebuah relasi dari sebuah <i>class</i> yang tidak bisa berdiri sendiri
	<i>Aggregation</i>	Menunjukkan sebuah bagian relasi agregasi
	<i>Dependency</i>	Menunjukkan operasi sebuah <i>class</i> yang menggunakan sebuah <i>class</i> lain.

2. Usecase Diagram

Usecase menurut Martin Fowler adalah teknik untuk merekam persyaratan fungsional. Sebuah *Usecase* mendeskripsikan interaksi tipikal antara para pengguna dengan itu sendiri, dengan memberi sebuah narasi tentang bagaimana tersebut digunakan. *Usecase Diagram* menampilkan aktor mana yang menggunakan *usecase* mana, *usecase* mana yang memasukkan *usecase* lain dan hubungan antara aktor dan *usecase* [21] seperti pada Tabel 2.3.

Tabel 2.3 Tabel simbol *usecase* diagram

Simbol	Nama	Keterangan
	<i>Actor</i>	<i>Actor</i> tersebut mempresentasikan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
	<i>Use Case</i>	Gambaran fungsionalitas dari suatu sistem, sehingga <i>customer</i> atau pengguna paham dan

		mengerti mengenai kegunaan yang akan dibangun.
	<i>Association</i>	Menghubungkan <i>link</i> antar <i>element</i> .
	<i>Include</i>	Kejadian yang harus terpenuhi agar sebuah <i>event</i> dapat terjadi, dimana pada kondisi ini sebuah <i>usecase</i> adalah bagian dari <i>usecase</i> lainnya.
	<i>Dependency</i>	Sebuah <i>element</i> bergantung dalam beberapa cara ke <i>element</i> lainnya.
	<i>Generalization</i>	Disebut juga <i>inheritance</i> (pewarisan).

3. Activity Diagram

Activity diagram adalah cara untuk menggambarkan aktivitas – aktivitas yang terdapat pada suatu sistem yang akan dibuat. Dengan *activity* diagram, alur dari kerja suatu sistem dapat dibuat dari awal hingga akhir. *Activity* diagram memperlihatkan siapa saja yang terlibat baik aktor maupun sistem serta memberikan keterangan proses dari tiap bagian tersebut. Pada *Activity* Diagram terdapat beberapa simbol dan beberapa cara penulisan diantaranya sebagai berikut seperti pada Tabel 2.4:

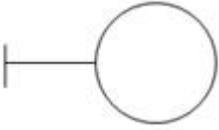
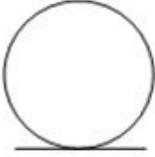
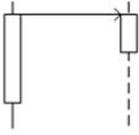
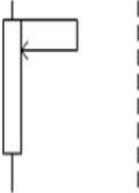
Tabel 2.4 Tabel simbol *activity* diagram

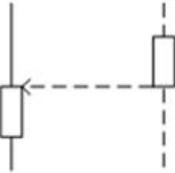
Simbol	Nama	Keterangan
	<i>Activity</i>	Menunjukkan sebuah aktivitas yang dilakukan.
	<i>Initial Node</i>	Awal dari aktivitas suatu sistem atau program.
	<i>Activity Final Node</i>	Akhir dari suatu aktivitas.
	<i>Decision</i>	Menunjukkan sebuah pilihan.
	<i>Line Connector</i>	Menghubungkan antara simbol yang satu dan yang lainnya.

4. Sequence Diagram

Sequence diagram adalah grafik dua dimensi dimana obyek ditunjukkan dalam dimensi horizontal, sedangkan *lifeline* ditunjukkan dalam dimensi vertikal seperti pada Tabel 2.5.

Tabel 2.5 Tabel simbol *sequence* diagram.

Simbol	Nama	Keterangan
	<i>Actor</i>	Menggambarkan seseorang atau sesuatu (seperti perangkat, sistem lain) yang berinteraksi dengan sistem.
	<i>Boundary</i>	Menggambarkan interaksi antara satu atau lebih <i>actor</i> dengan sistem, memodelkan bagian dari yang bergantung pada pihak lain disekitarnya dan merupakan pembatas dengan dunia luar.
	<i>Control</i>	Menggambarkan “perilaku mengatur”, mengkoordinasikan perilaku dan dinamika dari suatu sistem, menangani tugas utama dan mengontrol alur kerja suatu sistem.
	<i>Entity</i>	Menggambarkan informasi yang harus disimpan oleh sistem (struktur data dari sebuah sistem).
	<i>Object Message</i>	Menggambarkan pesan/hubungan antar obyek yang menunjukkan urutan kejadian yang terjadi.
	<i>Generalization</i>	Menggambarkan pesan/hubungan obyek itu sendiri, yang menunjukkan urutan kejadian yang terjadi.

	<i>Return Message</i>	Menggambarkan pesan/hubungan antar obyek, yang menunjukkan urutan kejadian yang terjadi.
	<i>Lifeline</i>	Eksekusi obyek selama <i>sequence</i> (<i>message</i> dikirim atau diterima dan aktifasinya).

2.2.10 CodeIgniter

CodeIgniter adalah sebuah *framework* PHP yang dapat membantu mempercepat developer dalam pengembangan aplikasi web berbasis PHP dibanding menulis program dari awal [22].

MVC adalah sebuah pendekatan yang ditempuh untuk memisahkan aplikasi menjadi 3 bagian, yaitu Model, *View* dan *Controller*. MVC memberikan struktur kepada aplikasi, sehingga dapat dicapai “*code reusability*”

1. Model

Model merepresentasikan data yang digunakan aplikasi. Contoh data misalnya *database*, RSS, atau data yang diperoleh dari pemanggilan API, dan aksi yang melibatkan operasi *Create, Read, Update, dan Delete* (CRUD) data.

2. View

View adalah informasi yang ditampilkan kepada *user* melalui *browser*. Informasi yang ditampilkan biasanya berupa file HTML atau kode PHP yang menyusun *template* untuk sebuah *website*.

3. Controller

Controller adalah “*business logic*” bertugas sebagai jembatan antara model dan *view*. *Controller* akan merespon HTTP *request* yang datang dari *user* melalui *browser*, dari *request* ini *controller* akan menentukan proses yang akan dilakukan [23].

2.2.11 XAMPP

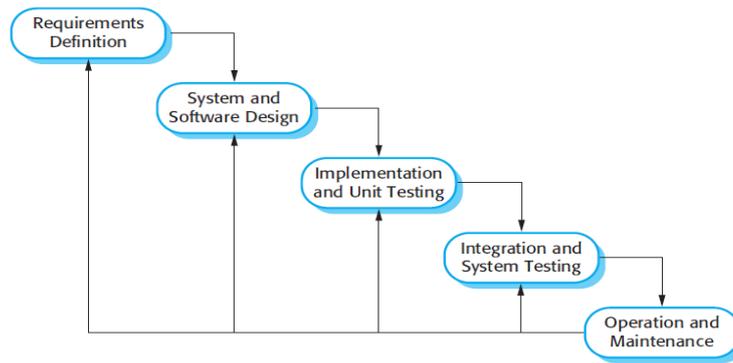
XAMPP adalah sebuah *software* web server apache didalamnya sudah tersedia *database* server mysql dan *support php programming*. XAMPP merupakan *software* yang mudah digunakan, gratis dan mendukung instalasi di Linux dan Windows.

phpMyAdmin adalah perangkat lunak bebas yang ditulis dalam bahasa pemrograman PHP yang digunakan untuk menangani administrasi MySQL melalui Jejaring Jagat Jembar (*World Wide Web*). *phpMyAdmin* mendukung berbagai operasi MySQL, diantaranya (mengelola basis data, tabel-tabel, bidang (*fields*), relasi (*relations*), indeks, pengguna (*users*), perijinan (*permissions*), dan lain-lain). Pada dasarnya, mengelola basis data dengan MySQL harus dilakukan dengan cara mengetikkan baris-baris perintah yang sesuai (*command line*) untuk setiap maksud tertentu. Jika seseorang ingin membuat basis data (*database*), ketikkan baris perintah yang sesuai untuk membuat basis data. Jika seseorang menghapus tabel, ketikkan baris perintah yang sesuai untuk menghapus tabel. Hal tersebut tentu saja sangat menyulitkan karena seseorang harus hafal dan mengetikkan perintahnya satu per satu.

Fungsi XAMPP sendiri adalah sebagai *server* yang berdiri sendiri (*localhost*), yang terdiri beberapa program antara lain : Apache HTTP Server, MySQL *database*, dan penerjemah bahasa yang ditulis dengan bahasa pemrograman PHP dan Perl. Nama XAMPP sendiri merupakan singkatan dari X (empat operasi sistem apapun), Apache, MySQL, PHP dan Perl. Program ini tersedia dalam GNU *General Public License* dan bebas, merupakan web server yang mudah untuk digunakan yang dapat menampilkan halaman web yang dinamis. [24].

2.2.12 Metode Pengembangan *Waterfall*

Tahapan utama dari *waterfall model* langsung mencerminkan aktifitas pengembangan dasar. Terdapat 5 tahapan pada *waterfall model*, yaitu *requirement analysis and definition*, *system and software design*, *implementation and unit testing*, *integration and system testing*, dan *operation and maintenance* [25].



Gambar 2.12 Model pengembangan *waterfall*.

Berikut adalah penjelasan dari tahapan-tahapan tersebut :

1. *Requirement Analysis and Definition*

Merupakan tahapan penetapan fitur, kendala dan tujuan melalui konsultasi dengan pengguna. Semua hal tersebut akan ditetapkan secara rinci dan berfungsi sebagai spesifikasi .

2. *System and Software Design*

Dalam tahapan ini akan dibentuk suatu arsitektur berdasarkan persyaratan yang telah ditetapkan. Dan juga mengidentifikasi dan menggambarkan abstraksi dasar perangkat lunak dan hubungan-hubungannya.

3. *Implementation and Unit Testing*

Dalam tahapan ini, hasil dari desain perangkat lunak akan direalisasikan sebagai satu set program atau unit program. Setiap unit akan diuji apakah sudah memenuhi spesifikasinya.

4. *Integration and System Testing*

Dalam tahapan ini, setiap unit program akan diintegrasikan satu sama lain dan diuji sebagai satu yang utuh untuk memastikan sudah memenuhi persyaratan yang ada. Setelah itu akan dikirim ke pengguna .

5. *Operation and Maintenance*

Dalam tahapan ini, diinstal dan mulai digunakan. Selain itu juga memperbaiki *error* yang tidak ditemukan pada tahap pembuatan. Dalam tahap ini juga dilakukan pengembangan seperti penambahan fitur dan fungsi baru.

BAB III

METODE PENELITIAN

Dalam bagian ini akan dijelaskan tentang alat dan bahan, metode pengembangan sistem, output sistem, evaluasi dan dokumentasi dari Sistem Informasi Geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – oleh khas Lombok dengan menggunakan fungsi *WayPoint* dan metode *Heuristik Greedy* Di Kota Mataram Berbasis *Web*. Berikut ini akan diberikan penjelasan dari masing-masing bagian tersebut.

3.1 Alat dan Bahan

Pada pembuatan sistem informasi geografis pencarian rute tercepat lokasi penjualan oleh – oleh khas Lombok dengan menggunakan fungsi *waypoint* dan metode *heuristik greedy* di Kota Mataram berbasis *web* ini membutuhkan beberapa alat dan bahan yang terdiri dari perangkat keras (*hardware*) dan perangkat lunak (*software*). Perangkat keras yang dibutuhkan untuk membuat aplikasi yaitu:

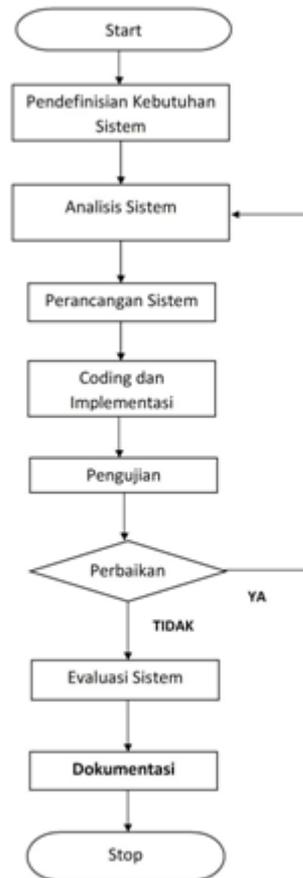
Laptop yang akan digunakan sebagai media pembangunan aplikasi dan untuk uji coba yang akan digunakan oleh *admin*.

Perangkat lunak yang dibutuhkan untuk membuat aplikasi yaitu:

1. OS (*operating system*), pada pembuatan aplikasi digunakan Windows 7 32-bit sebagai media untuk menjalankan *software-software* yang lain yang akan digunakan untuk membuat sistem.
2. Sublime Text merupakan *text editor* yang akan digunakan untuk membuat *code* program dari aplikasi yang akan dibuat pada tugas akhir ini.
3. Menggunakan PHP sebagai bahasa pemrogramannya
4. CodeIgniter digunakan sebagai *framework* pembuatan aplikasi berupa *website* yang akan mempermudah dalam pembangunan aplikasi.
5. XAMPP merupakan *software web server* apache yang didalamnya terdapat *server* MySQL yang akan digunakan sebagai *server* dalam pembuatan aplikasi dan digunakan untuk mengelola *database* yang dibutuhkan aplikasi.
6. Microsoft office 2007 digunakan untuk penyusunan laporan.

3.2 Metode Pengembangan

Proses pengembangan sistem informasi geografis pencarian rute tercepat lokasi penjualan oleh – oleh khas Lombok berbasis *web* menggunakan metode pengembangan *waterfall* yang terdiri dari 5 tahap, yaitu pendefinisian kebutuhan, perancangan, *coding* dan implementasi, pengujian serta dokumentasi. Tahap-tahap pengembangan tersebut diilustrasikan pada *diagram* alir atau *flowchart* pada Gambar 3.1 berikut:



Gambar 3.1 *Flowchart* pengembangan sistem.

3.2.1 Pendefinisian Kebutuhan Sistem

Tahap pertama dalam pembangunan sistem informasi geografis pencarian rute tercepat lokasi penjualan oleh – oleh khas Lombok dengan menggunakan fungsi *waypoint* dan metode *heuristik greedy* di Kota Mataram berbasis *web* ini adalah pendefinisian kebutuhan sistem. Pada tahap ini dilakukan pendeskripsian, pengumpulan data dan penjabaran berbagai kebutuhan *admin* dan *user* sehingga sistem dapat dirancang secara objektif dan dapat berjalan dengan optimal.

3.2.1.1 Pengumpulan Data

Pengumpulan data adalah tahap mengumpulkan seluruh data-data yang diperlukan dalam pembangunan sistem. Pada pembangunan sistem informasi geografis pencarian rute tercepat lokasi penjualan oleh – oleh khas Lombok ini menggunakan data lokasi penjualan oleh –oleh khas Lombok dimana data yang diperlukan adalah data lokasi berupa *latitude* dan *longititude*, nama toko, alamat, dan jenis oleh - oleh yang digunakan untuk menampilkan data tersebut kedalam *google maps API* dan untuk proses pencarian rute terdekat sesuai dengan yang diinginkan pengguna sehingga pengumpulan data dilakukan dengan cara observasi pada setiap toko oleh – oleh di kota Mataram.

3.2.1.2 Penjabaran Kebutuhan Sistem

Penjabaran kebutuhan sistem adalah tahapan untuk menjabarkan dan menganalisis berbagai perilaku yang diperlukan sistem. Kebutuhan sistem harus sesuai dengan tujuan yang akan dibangun dan harus dapat memenuhi kebutuhan *end user* yang akan menggunakan sistem tersebut. Kebutuhan sistem yang telah dianalisis dalam sistem informasi geografis pencarian rute tercepat lokasi penjualan oleh – oleh khas Lombok dapat dirincikan sebagai berikut:

1. Sistem mampu memberikan informasi mengenai letak tempat penjualan oleh khas Lombok di Kota Mataram dan Kabupaten Lombok Barat dengan lokasi koordinat *latitude* dan *longitude* berbasis peta digital.
2. Sistem mampu memberikan informasi dalam hal untuk mencari lokasi penjualan oleh – oleh khas Lombok dan menentukan rute tercepat ke lokasi menggunakan fungsi *waypoint* dengan metode *heuristik greedy*.
3. Sistem mampu memberikan informasi oleh-oleh yang dijual oleh tempat penjualan oleh khas lombok di Kota Mataram.
4. Sistem mampu memberikan informasi rute tercepat untuk dua atau tiga lokasi sesuai dengan yang diinginkan pengguna.

3.2.1.3 Penjabaran Kebutuhan *End User*

Pendefinisian kebutuhan *end user* adalah menjabarkan dan menganalisis berbagai kebutuhan *end user* yang akan dipenuhi oleh sistem yang akan dibangun. Dalam sistem informasi geografis pencarian rute tercepat lokasi penjualan oleh – oleh khas Lombok dengan menggunakan fungsi *waypoint* dan metode *heuristik greedy* di Kota Mataram berbasis *web*, terbagi atas tiga *user* yaitu pemerintah sebagai *admin* yang mengelola data

lokasi toko dan pemiliknya, pemilik toko yang mengelola data barang oleh-oleh yang dijual dan masyarakat umum sebagai *end user*. Berbagai kebutuhan *end user* dapat didefinisikan dalam rincian sebagai berikut:

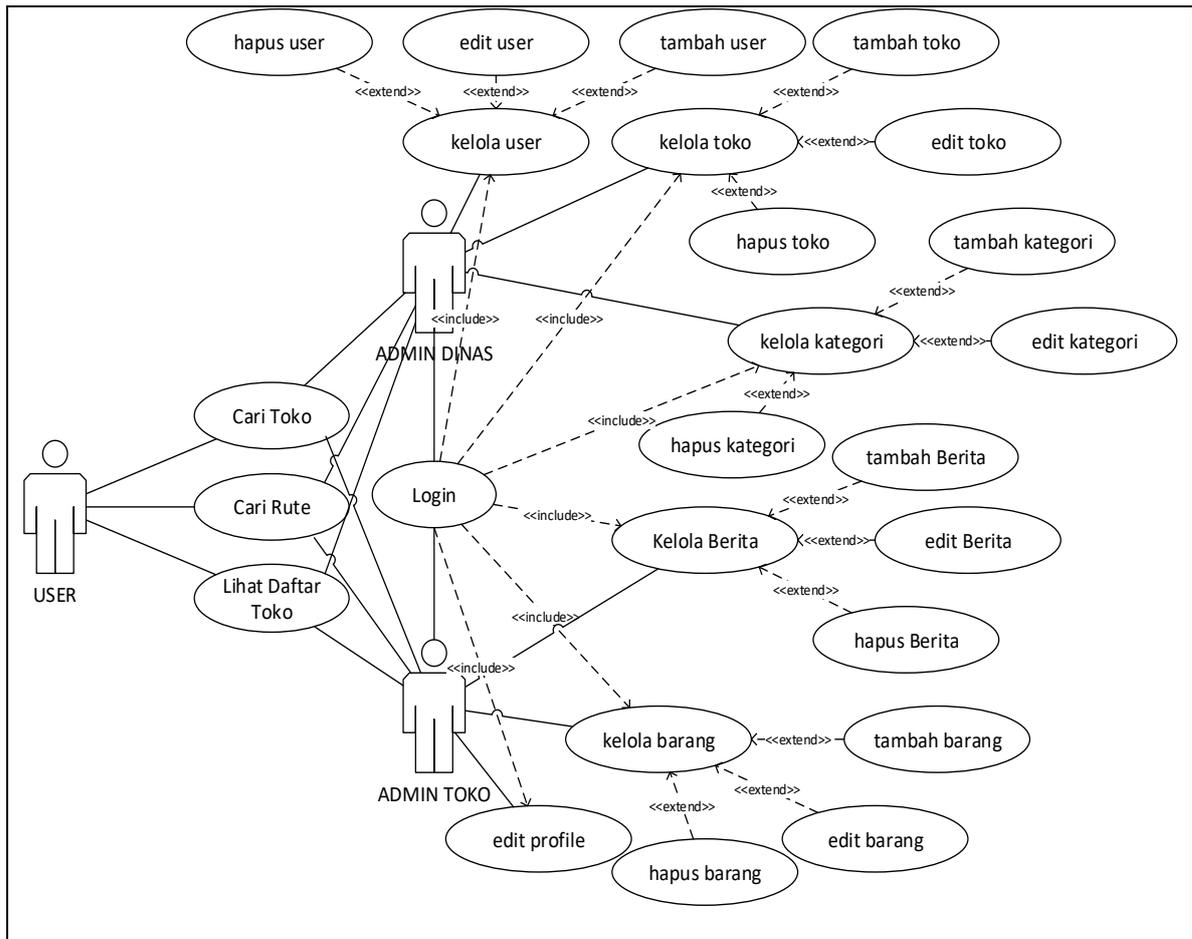
1. *End user* memerlukan suatu media yang menyediakan data dan informasi mengenai lokasi penjualan oleh – oleh khas Lombok yang dapat diakses secara *online* yang *user friendly*.
2. *End user* memerlukan media yang menyediakan informasi mengenai rute tercepat untuk dua atau tiga lokasi sesuai dengan yang diinginkan pengguna yang dapat diakses secara *online* yang *user friendly*.

3.2.2 Perancangan Sistem

Dalam tahap perancangan ini, terdapat enam perancangan yang dilakukan, yaitu perancangan *behaviour*, perancangan *class*, perancangan aktivitas aktor, perancangan *deployment*, perancangan *database* dan perancangan *layout*

3.2.2.1 Perancangan Behaviour

Pada perancangan sistem, diperlukan suatu proses yang menggambarkan kelakuan (*behavior*) sistem yang akan dibuat, yaitu dengan menggunakan *usecase diagram*. *Usecase diagram* mendeskripsikan interaksi antara aktor (pengguna aplikasi) dengan sistem yang akan dibuat. Dalam perancangan sistem informasi geografis pencarian lokasi oleh-oleh tercepat ini, terdapat tiga *actor* yaitu Dinas Kota Mataram sebagai super *admin*, toko oleh-oleh di Kota Mataram sebagai *admin* serta masyarakat umum sebagai *user*. *Behaviour* dari setiap aktor akan diilustrasikan dalam *usecase diagram* pada Gambar 3.2. Pada Gambar 3.2 terlihat beberapa *behaviour* dari *admin* dinas, diantaranya mengelola data lokasi, dan data *admin*, *behaviour* dari *admin* toko, diantaranya mengelola data barang, dan data berita . Untuk dapat menjalankan *behavior* dan fungsinya (ditunjukkan dengan tanda panah garis-garis berlabel *extend*), *admin* diharuskan untuk *login* terlebih dahulu (ditunjukkan dengan tanda panah garis-garis berlabel *include*) sehingga mendapatkan hak akses.

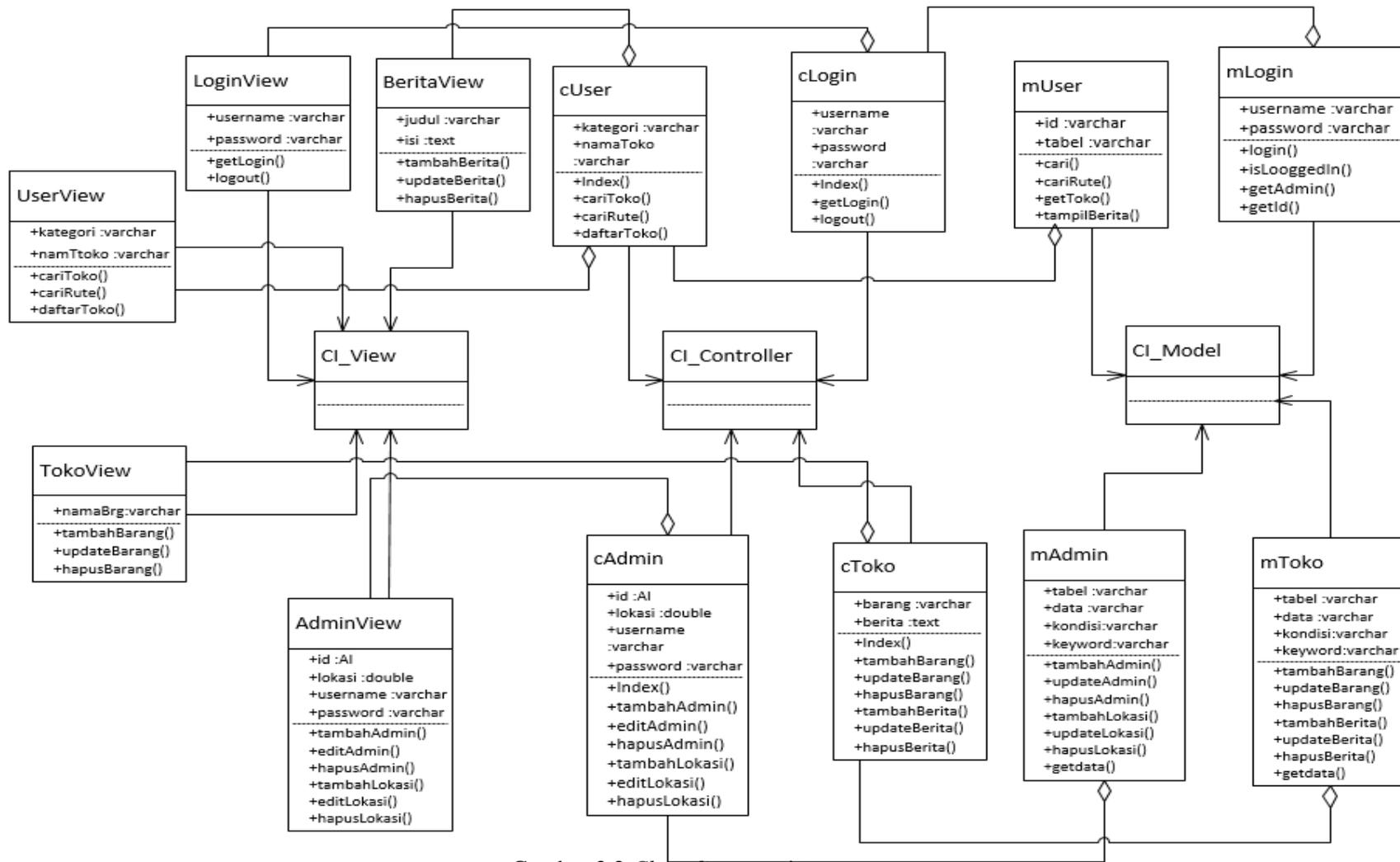


Gambar 3.2 Usecase diagram pencarian lokasi oleh-oleh terdekat.

Terdapat juga *behavior* dari *admin* dan masyarakat sebagai *user* mencari toko oleh-oleh, mencari toko terdekat, melihat daftar toko dan *behaviour* lainnya dapat dilakukan tanpa melakukan *login* untuk masyarakat.

3.2.2.2 Perancangan *Class*

Pada perancangan *class* ini, dilakukan dengan menggunakan *Class diagram* yaitu suatu rancangan kelas yang akan diimplementasikan dalam bentuk *interface* pada sistem yang akan dibangun. Perancangan *class* sistem informasi geografis pencarian lokasi oleh-oleh terdekat di Kota Mataram, digambarkan dalam *diagram* pada Gambar 3.3 berikut:



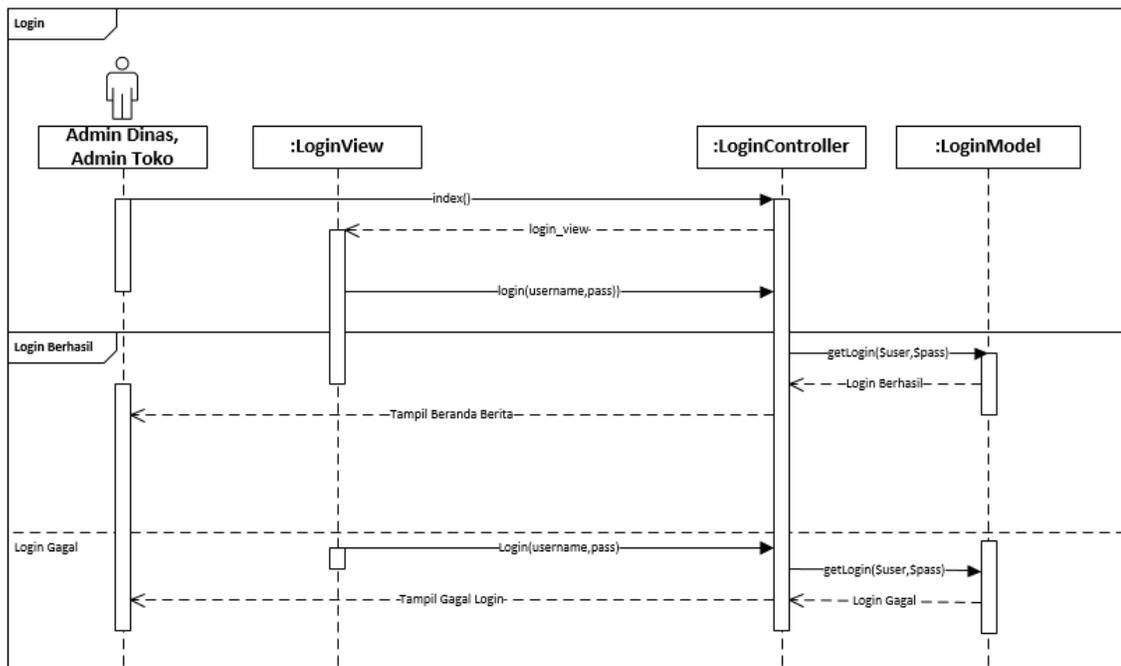
Gambar 3.3 Class diagram sistem.

3.2.2.3 Perancangan Aktifitas Aktor

Rancangan aktivitas aktor diilustrasikan dengan *Sequence diagram*, yaitu *diagram* yang menggambarkan interaksi objek dengan sistem. *Sequence diagram* menggambarkan secara bertahap aktivitas yang dilakukan oleh aktor berdasarkan urutan waktu. Rancangan *sequence diagram* pada sistem informasi geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – Oleh khas Lombok dengan menggunakan Fungsi Way Point dan Metode *Heuristik Greedy* di Kota Mataram dapat dirincikan sebagai berikut:

1. *Sequence diagram* proses login

Sequence diagram dari proses login yang dilakukan oleh *admin* Dinas dan Toko adalah seperti pada Gambar 3.4. Proses login dimulai ketika *admin* mengakses fungsi `index()` yang ada pada *controllerlogin*, kemudian fungsi `index()` akan me-load view `login_view` yang berisi halaman *form login*. Kemudian fungsi `login()` pada *controllerlogin* akan diakses, selanjutnya akan dicocokkan *username* dan *password* yang telah dimasukkan oleh *end user* dengan data yang ada di *database* beserta dengan level untuk menentukan *admin* dimana apabila level = 1 maka akan diarahkan ke beranda *admin* dinas apabila level = 2 maka akan diarahkan ke halaman *admin* toko.



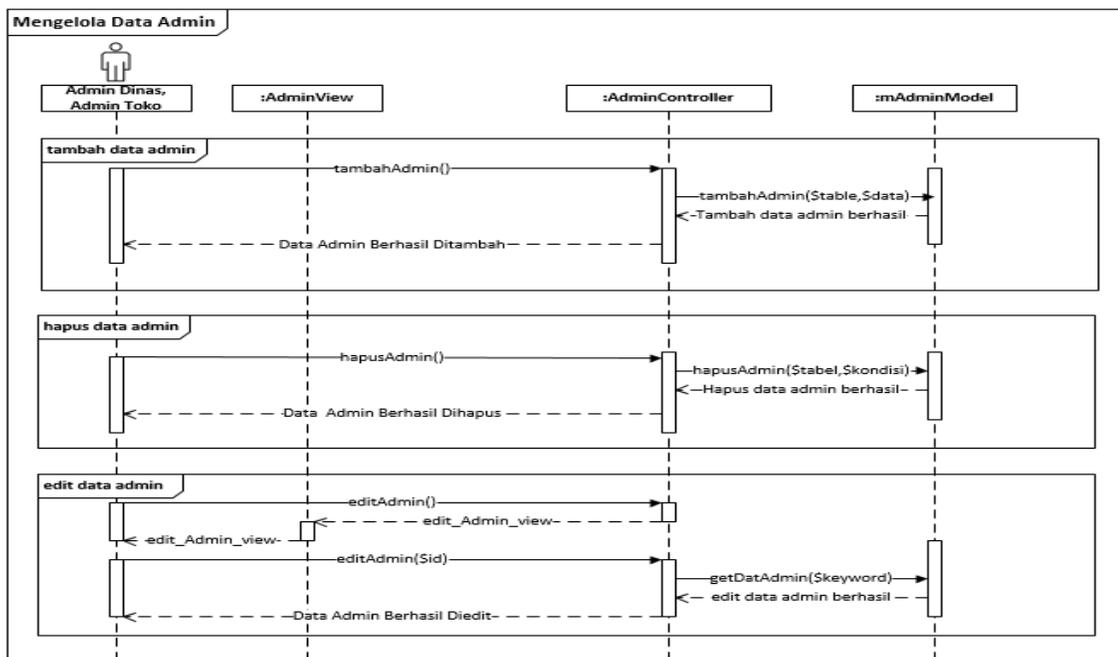
Gambar 3.4 *Sequence diagram* login.

2. Sequence diagram mengelola data admin

Sequence diagram dari mengelola data admin yang dilakukan oleh admin dinas seperti Gambar 3.5. Untuk proses tambah data admin dimulai ketika admin mengakses fungsi `index()` pada *controller admin*, yang selanjutnya akan me-load ke *admin_view* yang merupakan halaman untuk menampilkan data admin pada database. Pada halaman *admin_view* terdapat form tambah data admin. Ketika admin meng-klik tombol *submit* pada form tersebut maka akan dijalankan fungsi `tambahAdmin()`, selanjutnya data akan disimpan pada database dengan menggunakan fungsi `tambahAdmin($table,$data)` yang ada pada model *mAdmin*.

Untuk melakukan hapus data, maka fungsi `hapusAdmin()` yang ada pada *controller admin* yang akan dijalankan. Dari fungsi tersebut maka akan didapat kondisi dari data yang akan dihapus, kemudian kondisi tersebut akan dikirim ke kelas *mAdminModel* dan akan dieksekusi oleh fungsi `hapusAdmin()`.

Untuk melakukan edit data, maka fungsi pertama yang dijalankan yaitu fungsi `editAdmin()` pada *controller admin*, fungsi tersebut akan menampilkan data admin sebelumnya yang diambil berdasarkan *id*. Ketika admin melakukan perubahan pada data admin, data tersebut akan disimpan pada tabel *admin* di database. Untuk proses akses ke database, fungsi `getDatAdmin()` yang akan dijalankan.



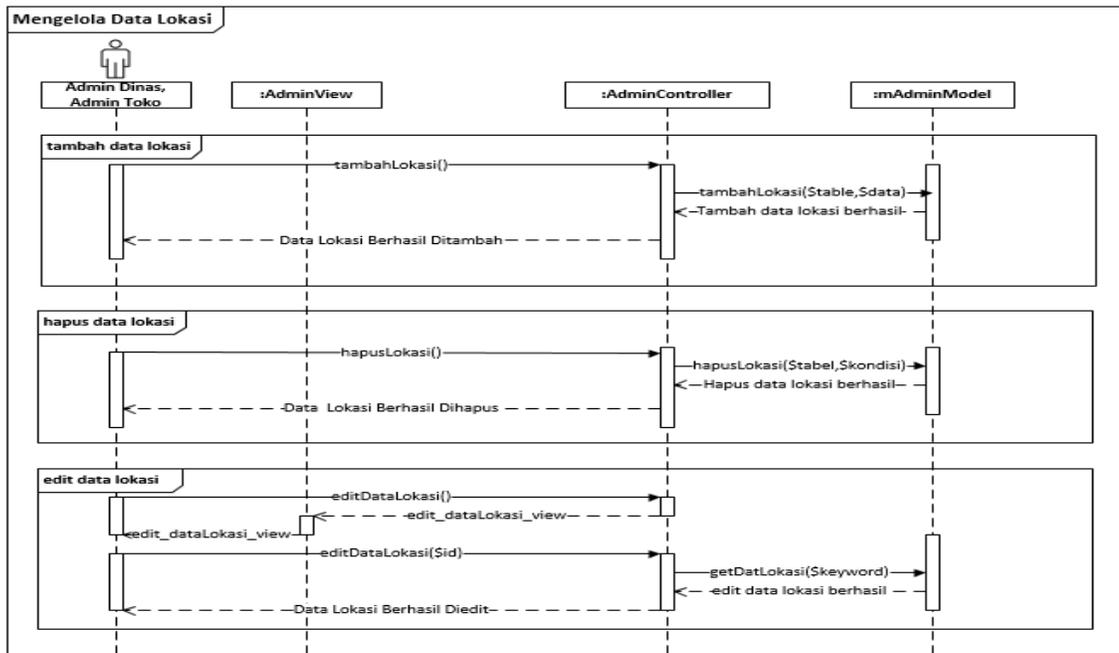
Gambar 3.5 Sequence diagram mengelola data admin.

3. Sequence diagram mengelola data lokasi

Berikut *sequence diagram* dari proses mengelola data lokasi yang dapat dilakukan oleh *admin* dinas yang dapat dilihat pada Gambar 3.6. Untuk proses tambah data lokasi dimulai ketika *admin* mengakses fungsi `tambahLokasi()` pada *controller admin*, yang selanjutnya akan me-load view *admin* yang merupakan halaman untuk menampilkan *form* tambah lokasi. Ketika *admin* meng-klik tombol *submit* maka akan dijalankan fungsi `tambahLokasi()` pada model `mAdmin` yang berfungsi untuk menyimpan data kedalam *database*.

Untuk melakukan hapus data, maka fungsi `hapusLokasi()` yang ada pada *controller admin* yang akan dijalankan. Dari fungsi tersebut maka akan didapat kondisi dari data yang akan dihapus, kemudian kondisi tersebut akan dikirim ke kelas `mAdminModel` dan akan dieksekusi oleh fungsi `hapusLokasi()`.

Untuk melakukan edit data, maka fungsi pertama yang dijalankan yaitu fungsi `editLokasi()` pada *controller admin*, fungsi tersebut akan menampilkan data *admin* sebelumnya yang diambil berdasarkan *id*. Ketika *admin* melakukan perubahan pada data *admin*, data tersebut akan disimpan pada tabel *admin* di *database*. Untuk proses akses ke *database*, fungsi `getDatLokasi()` yang akan dijalankan.



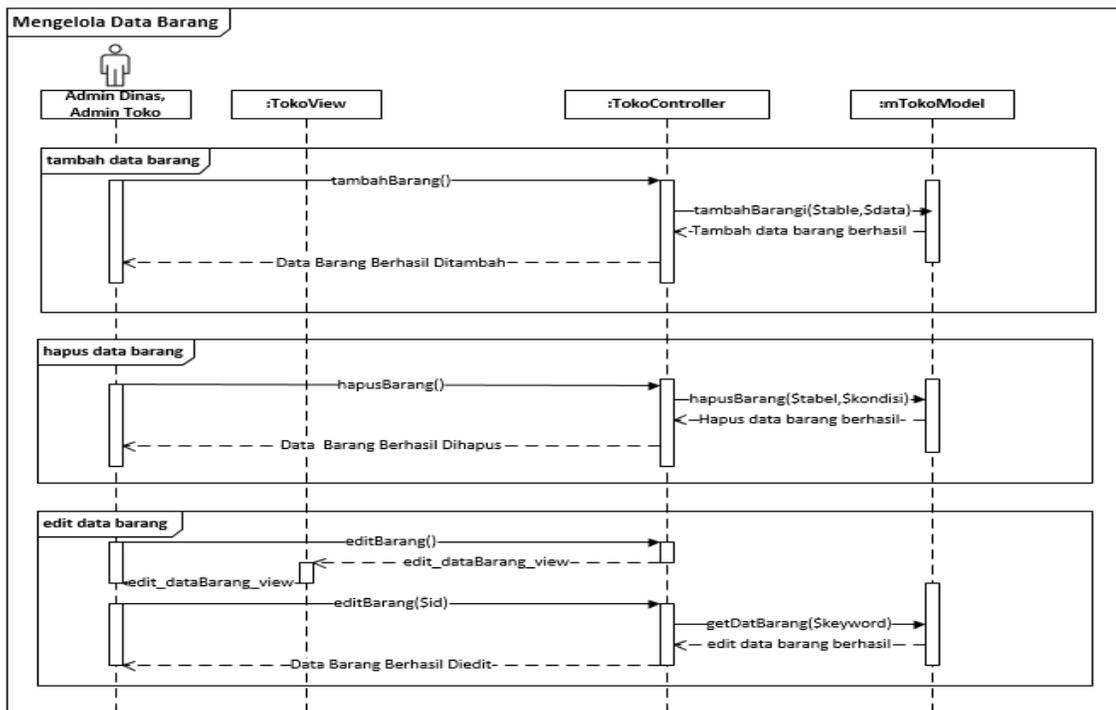
Gambar 3.6 *Sequence diagram* mengelola data lokasi.

4. Sequence diagram mengelola data barang

Berikut *sequence diagram* dari proses mengelola data lokasi yang dapat dilakukan oleh *admin* toko yang dapat dilihat pada Gambar 3.7. Untuk proses tambah data barang dimulai ketika *admin* mengakses fungsi `tambahBarang()` pada *controller* Toko, yang selanjutnya akan me-load view Toko yang merupakan halaman untuk menampilkan *form* tambah barang. Ketika *admin* meng-klik tombol *submit* maka akan dijalankan fungsi `tambahBarang()` pada model `mToko` yang berfungsi untuk menyimpan data kedalam *database*.

Untuk melakukan hapus data, maka fungsi `hapusBarang()` yang ada pada *controller* Toko yang akan dijalankan. Dari fungsi tersebut maka akan didapat kondisi dari data yang akan dihapus, kemudian kondisi tersebut akan dikirim ke kelas model `mToko` dan akan dieksekusi oleh fungsi `hapusBarang()`.

Untuk melakukan edit data, maka fungsi pertama yang dijalankan yaitu fungsi `editBarang()` pada *controller* Toko, fungsi tersebut akan menampilkan data barang sebelumnya yang diambil berdasarkan *id*. Ketika *admin* melakukan perubahan pada data barang, data tersebut akan disimpan pada tabel barang di *database*. Untuk proses akses ke *database*, fungsi `getDatBarang()` yang akan dijalankan.



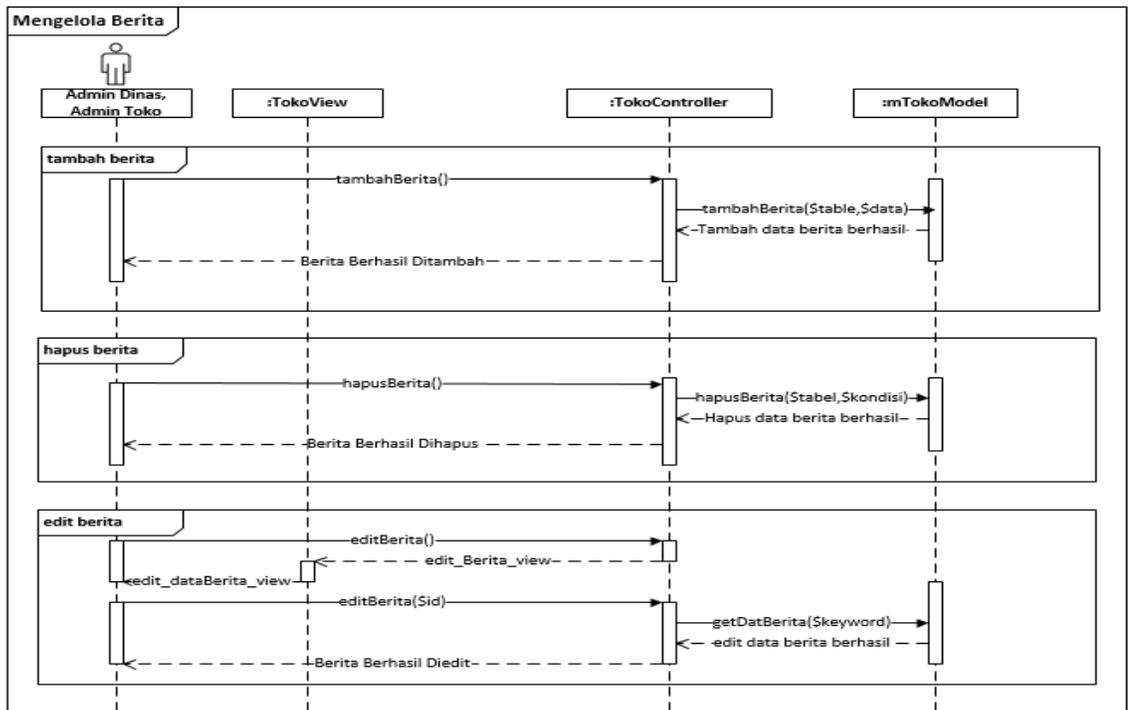
Gambar 3.7 Sequence diagram mengelola data barang.

5. Sequence diagram mengelola berita

Berikut *sequence diagram* dari proses mengelola data lokasi yang dapat dilakukan oleh *admin* toko yang dapat dilihat pada Gambar 3.8. Untuk proses tambah data barang dimulai ketika *admin* mengakses fungsi `tambahBerita()` pada *controller* Toko, yang selanjutnya akan me-load view Toko yang merupakan halaman untuk menampilkan *form* tambah berita. Ketika *admin* meng-klik tombol *submit* maka akan dijalankan fungsi `tambahBerita()` pada model `mToko` yang berfungsi untuk menyimpan data kedalam *database*.

Untuk melakukan hapus data, maka fungsi `hapusBerita()` yang ada pada *controller* Toko yang akan dijalankan. Dari fungsi tersebut maka akan didapat kondisi dari data yang akan dihapus, kemudian kondisi tersebut akan dikirim ke kelas model `mToko` dan akan dieksekusi oleh fungsi `hapusBerita()`.

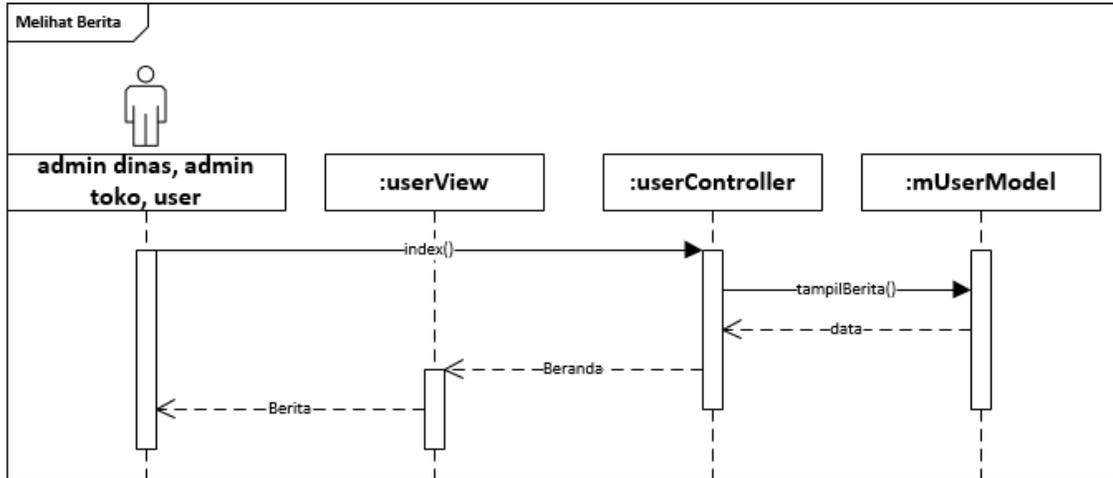
Untuk melakukan edit data, maka fungsi pertama yang dijalankan yaitu fungsi `editBerita()` pada *controller* Toko, fungsi tersebut akan menampilkan data barang sebelumnya yang diambil berdasarkan *id*. Ketika *admin* melakukan perubahan pada data berita, data tersebut akan disimpan pada tabel berita di *database*. Untuk proses akses ke *database*, fungsi `getDatBerita()` yang akan dijalankan.



Gambar 3.8 Sequence diagram mengelola berita.

6. Sequence diagram melihat berita

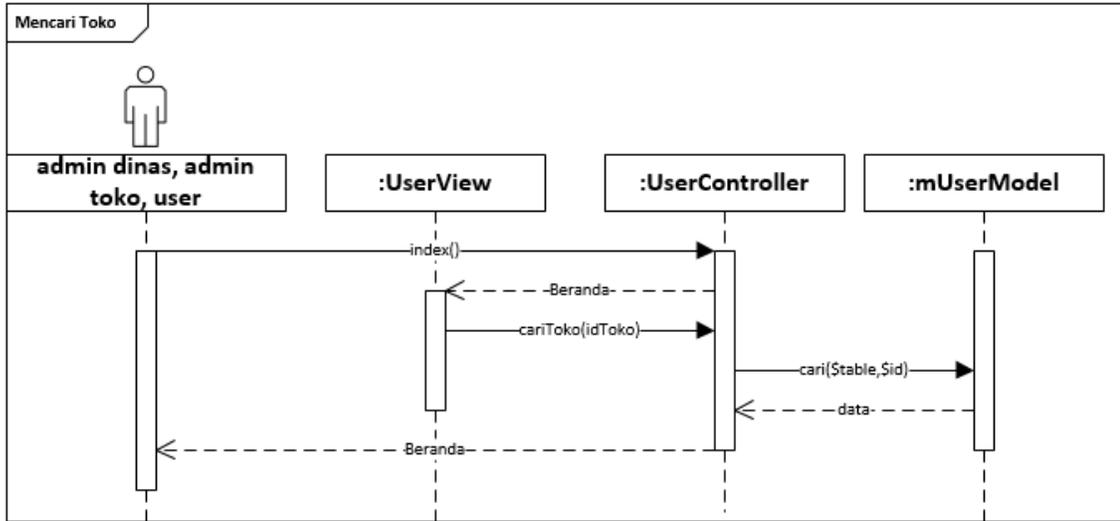
Sequence diagram dari proses melihat berita yang dilakukan oleh *admin* dan *user* adalah seperti pada Gambar 3.9. Proses dimulai ketika *user* dan *admin* mengakses fungsi `index()` pada *controller User*, selanjutnya akan dijalankan fungsi `tampilBerita()` berdasarkan data pada kelas `mUserModel`, selanjutnya akan ditampilkan data berita pada *view user*.



Gambar 3.9 *Sequence diagram* melihat berita.

7. Sequence diagram cari toko

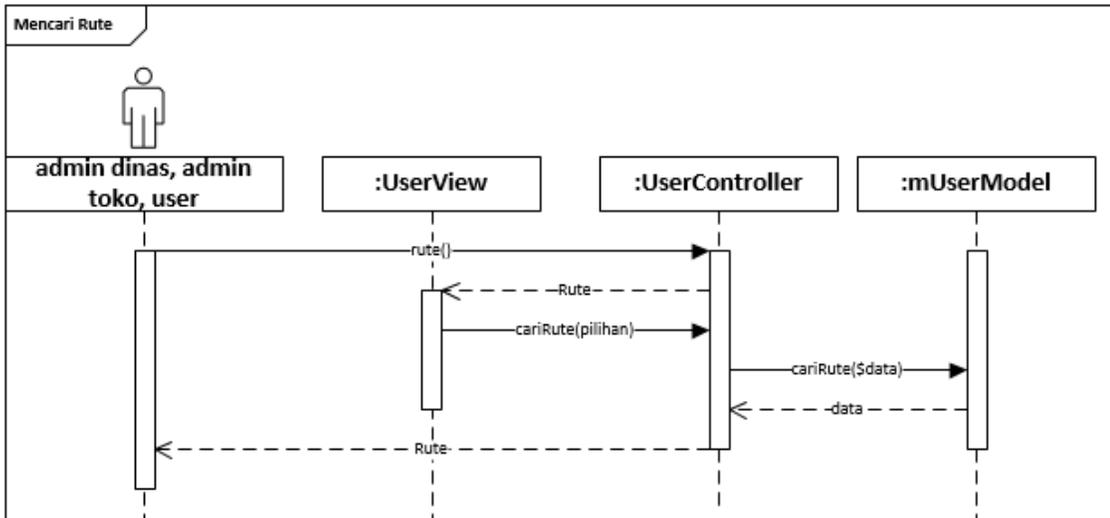
Sequence diagram dari proses mencari toko yang dilakukan oleh *admin* dan *user* adalah seperti pada Gambar 3.10. Proses dimulai ketika *user* dan *admin* mengakses fungsi `index()` pada *controller User* yang akan menampilkan *view* Beranda, selanjutnya akan dijalankan fungsi `cariToko()` pada *controller User* ketika *user* atau *admin* ingin mencari toko yang diinginkan dimana fungsi `cari($table, $id)` yang terhubung pada *UserModel* akan menampilkan hasil dari cari toko pada kelas *UserView*.



Gambar 3.10 *Sequence diagram* cari toko.

8. *Sequence diagram* cari rute

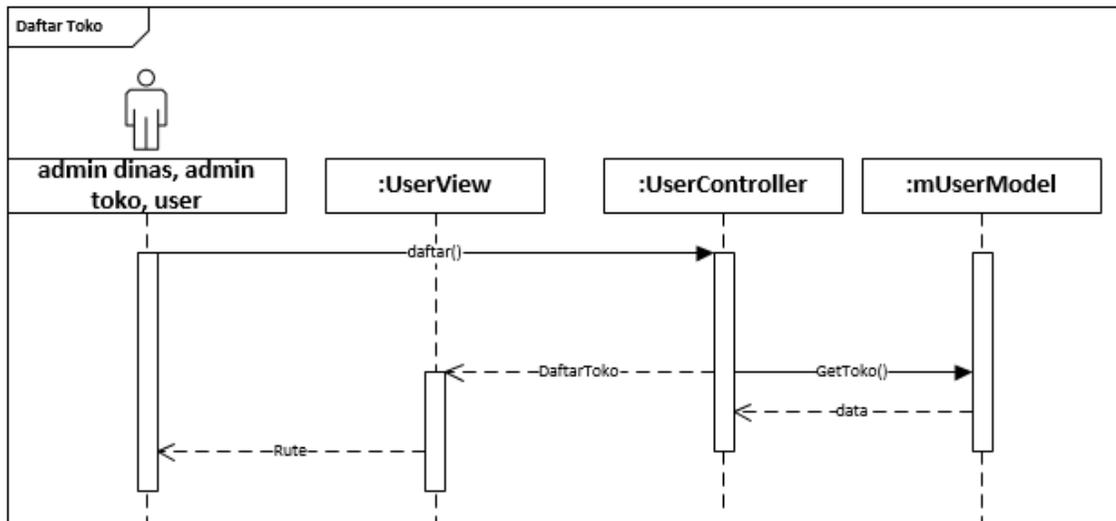
Sequence diagram dari proses mencari rute yang dilakukan oleh *admin* dan *user* adalah seperti pada Gambar 3.11. Proses dimulai ketika *user* dan *admin* mengakses fungsi `route()` pada *controller User* yang akan menampilkan *view Rute*, selanjutnya akan dijalankan fungsi `cariRute()` sesuai dengan data yang dimasukkan *user* pada kelas *controller User*. Selanjutnya dijalankan fungsi `cariRute()` pada model *user* yang akan menampilkan hasil dari cari rute pada kelas *UserView*.



Gambar 3.11 *Sequence diagram* cari rute.

9. Sequence diagram melihat daftar toko

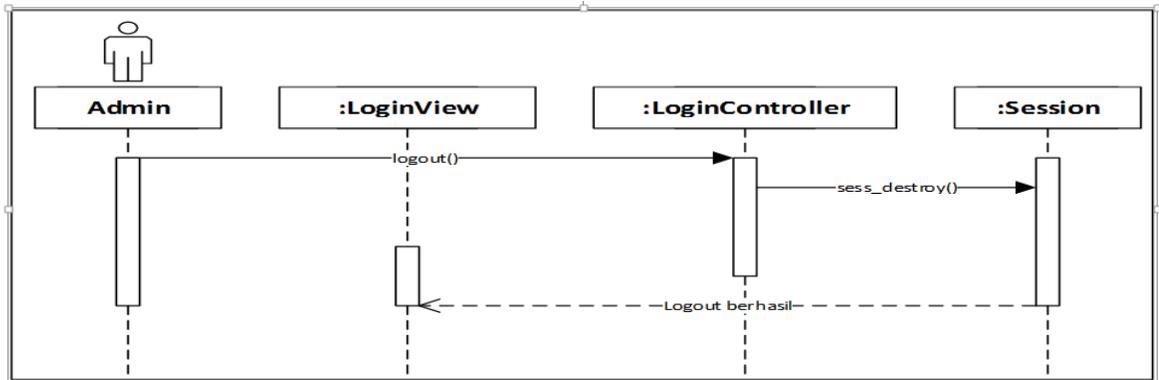
Sequence diagram dari proses melihat berita yang dilakukan oleh *admin* dan *user* adalah seperti pada Gambar 3.12. Proses dimulai ketika *user* dan *admin* mengakses fungsi `daftar()` pada *controller User*, selanjutnya akan dijalankan fungsi `GetToko()` berdasarkan data pada kelas `mUserModel`, selanjutnya akan ditampilkan data toko pada *view user*.



Gambar 3.12 *Sequence diagram* melihat daftar toko.

10. Sequence diagram logout

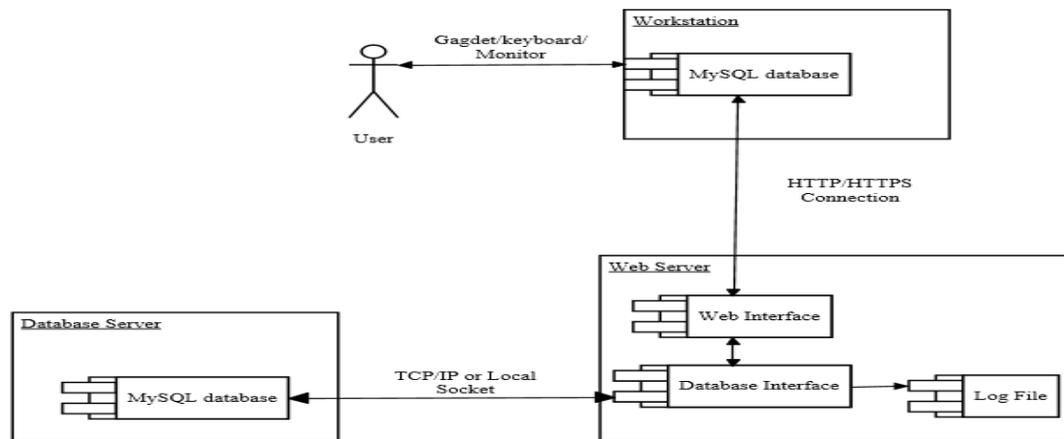
Sequence diagram dari proses *logout* yang dilakukan oleh *admin* adalah seperti pada Gambar 3.14. *Admin* akan mengakses fungsi `logout()` pada *controllerlogin*, kemudian fungsi `sess_destroy()` akan dijalankan untuk menghapus *session*, sehingga untuk masuk ke *sistem admin* harus *login* dengan memasukkan *username* dan *password* kembali.



Gambar 3.13 Sequence diagram logout

3.2.2.4 Perancangan Deployment diagram

Deployment diagram adalah salah satu model diagram dalam UML untuk mengerahkan *artifact* dalam node. Deployment diagram digunakan untuk memvisualisasikan, menspesifikasikan dan mendokumentasikan proses yang terjadi pada suatu sistem perangkat lunak berbasis objek yang akan dibangun. Berikut perancangan deployment yang digambarkan dalam deployment diagram pada Gambar 3.13:

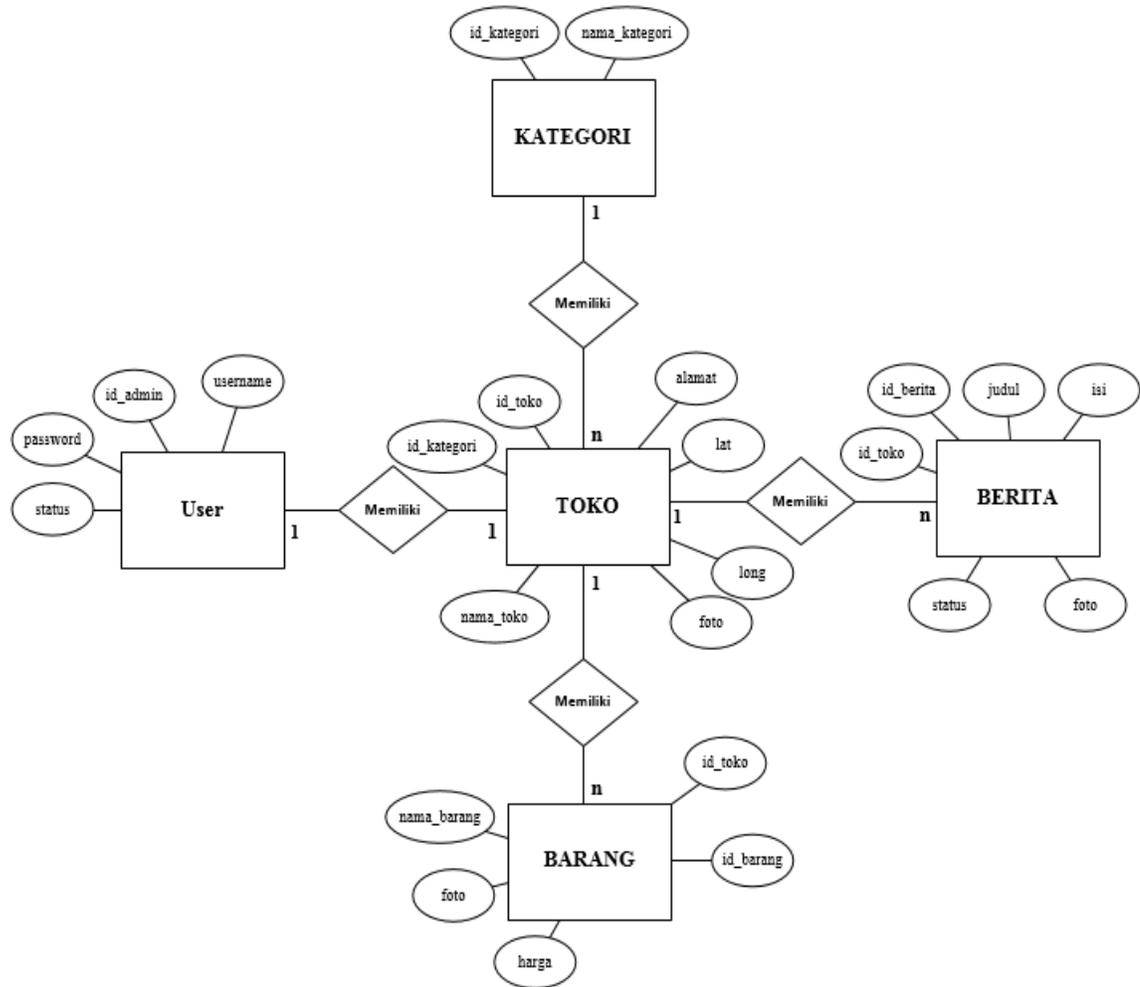


Gambar 3.14 Deployment diagram.

3.2.2.5 Perancangan Database

Tahap selanjutnya setelah merancang alur kerja dan *behaviour* setiap *user* adalah perancangan *database*. Data-data yang ada di dalam ini, seluruhnya akan disimpan di dalam satu *database*, sehingga data menjadi terintegrasi. Perancangan *database* perlu dilakukan agar penyimpanan data dapat dilakukan secara optimal dan mengurangi adanya

duplikasi data yang juga dapat meminimumkan media penyimpanan. Berikut adalah rancangan *database* dalam informasi geografis pencarian lokasi oleh-oleh terdekat di Kota Mataram yang diilustrasikan pada Gambar 3.1.



Gambar 3.15 Entity relationship diagram.

Berdasarkan rancangan ERD pada Gambar 3.15, terdapat lima entitas yang ada di dalam *database* pencarian rute lokasi oleh-oleh, yaitu *admin*, berita, barang, kategori, dan toko. Struktur tabel *database* dari ERD tersebut dapat dilihat dalam rincian tabel berikut:

Tabel 3.1 Tabel admin

No	Field	Tipe Data	Panjang	Key	Keterangan
1	<u>Id_admin</u>	VARCHAR	9	PK	Primary Key tabel admin

2	<i>Username</i>	VARCHAR	25		<i>Username untuk login ke sistem</i>
3	<i>Password</i>	VARCHAR	50		<i>Password untuk login ke sistem</i>
4	<i>Status</i>	VARCHAR	2		<i>Status user</i>

Tabel 3.2 Tabel berita.

No	<i>Field</i>	Tipe Data	Panjang	<i>Key</i>	Keterangan
1	<u>Id_berita</u>	VARCHAR	9	PK	<i>Primary Key</i> tabel Berita
2	Id_toko	VARCHAR	9	FK	Id toko
3	foto	VARCHAR	40		foto
4	Judul	VARCHAR	50		Judul berita
5	Isi	TEXT			Isi berita
6	status	VARCHAR	6		status

Tabel 3.3 Tabel barang.

No	<i>Field</i>	Tipe Data	Panjang	<i>Key</i>	Keterangan
1	<u>Id_barang</u>	VARCHAR	9	PK	<i>Primary Key</i> tabel barang
2	Id_toko	VARCHAR	9	FK	Id toko
3	nama_barang	VARCHAR	25		Nama barang
4	harga	INT	10		Harga dari barang oleh-oleh
5	foto	VARCHAR	40		Foto barang

Tabel 3.4 Tabel kategori.

No	<i>Field</i>	Tipe Data	Panjang	<i>Key</i>	Keterangan
1	Id_kategori	VARCHAR	9	PK	<i>Primary Key</i> tabel kategori
2	nama_kategori	VARCHAR	10		Nama kategori

Tabel 3.5 Tabel toko.

No	<i>Field</i>	Tipe Data	Panjang	<i>Key</i>	Keterangan
1	Id_toko	VARCHAR	9	PK	<i>Primary Key</i> tabel pendidikan
2	Id_kategori	VARCHAR	9	FK	Id kategori
3	Nama_toko	VARCHAR	20		Nama toko
4	lat	DOUBLE			Latitude dari lokasi toko
5	long	DOUBLE			Longitude dari lokasi toko

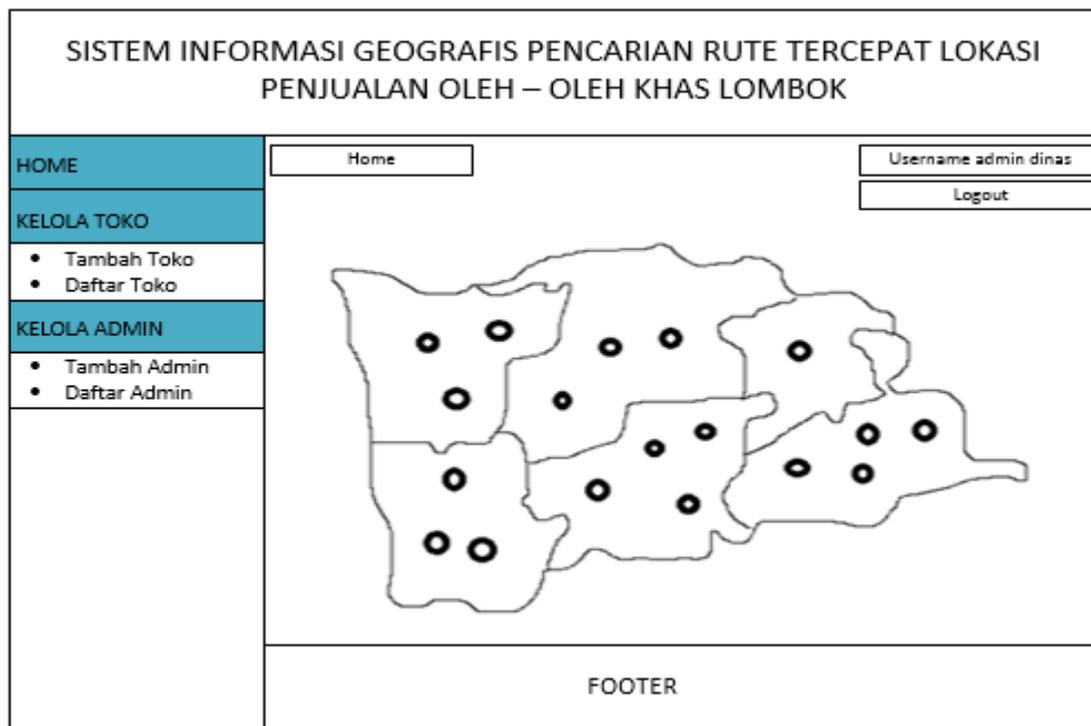
6	alamat	VARCHAR	100		Alamat dari toko oleh-oleh
7	foto	VARCHAR	40		Foto toko oleh-oleh

3.2.2.6 Perancangan *Layout Website*

Tahap desain *layout website* adalah tahap perancangan terhadap desain *interface website* yang akan berkomunikasi dengan *user*. Pada tahap ini, rancangan yang dilakukan harus mementingkan kebutuhan *user*, yaitu *user* membutuhkan sebuah *website* yang sederhana, mudah dimengerti (*user friendly*) namun tetap menarik secara visual. Berikut merupakan perancangan *layout website* pada sistem informasi geografis pencarian rute lokasi oleh-oleh terdekat di Kota Mataram:

1. Halaman *Home admin*.

Pada halaman *home admin* menampilkan peta yang berisikan data lokasi dengan fitur kelola toko dan kelola *admin* yang dapat dilihat pada Gambar 3.16.



Gambar 3.16 Halaman *Home admin*.

2. Halaman Tambah *admin*

Halaman tambah *admin* pada sisi *admin* dinas perindag. Pada *interface* ini akan menampilkan *form* untuk menambahkan data *admin* yang dapat ditambahkan oleh *admin* dinas perindag yang dapat dilihat pada Gambar 3.17.

SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK		
HOME	Home > Tambah Admin	Username admin Logout
KELOLA TOKO	Tambah Admin	
KELOLA ADMIN	<input type="text" value="Nama"/> <input type="text" value="Lokasi"/> <input type="checkbox"/> <input type="text" value="Username"/> <input type="text" value="Password"/>	
<ul style="list-style-type: none"> • Tambah Admin • Daftar Admin 	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>	
FOOTER		

Gambar 3.17 Halaman Tambah Admin.

3. Halaman Daftar Admin

Pada halaman daftar *admin* pada sisi *admin* dinas Perindag. Pada *interface* daftar *admin* menampilkan daftar dari *admin* yang terdaftar pada sistem yang dapat dilihat pada Gambar 3.18.

SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK																																												
HOME	Home > Data Admin	Username admin Logout																																										
KELOLA TOKO	Data Admin																																											
KELOLA ADMIN	<table border="1"> <thead> <tr> <th>No</th> <th>Nama</th> <th>Lokasi</th> <th>Username</th> <th>Password</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Edit Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Edit Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Edit Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Edit Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Edit Hapus</td> </tr> <tr> <td></td> <td></td> <td></td> <td></td> <td></td> <td>Edit Hapus</td> </tr> </tbody> </table>		No	Nama	Lokasi	Username	Password	Action						Edit Hapus																														
No	Nama	Lokasi	Username	Password	Action																																							
					Edit Hapus																																							
					Edit Hapus																																							
					Edit Hapus																																							
					Edit Hapus																																							
					Edit Hapus																																							
					Edit Hapus																																							
<ul style="list-style-type: none"> • Tambah Admin • Daftar Admin 	FOOTER																																											

Gambar 3.18 Halaman Daftar Admin.

4. Halaman Daftar Toko

Pada halaman daftar toko pada sisi *admin* dinas perindag . Pada halaman tersebut akan ditampilkan daftar toko oleh – oleh yang sudah didaftarkan kedalam sistem yang dapat dilihat pada Gambar 3.19.

SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK						
HOME	Home > Data Toko		Username admin dinas			
			Logout			
KELOLA TOKO	Data Toko					
<ul style="list-style-type: none">• Tambah Toko• Daftar Toko						
KELOLA ADMIN						
	FOOTER					

No	Nama Toko	Alamat	lat	long	status	Action
						Edit Hapus
						Edit Hapus
						Edit Hapus
						Edit Hapus
						Edit Hapus
						Edit Hapus

Gambar 3.19 Halaman Daftar Toko.

5. Halaman Tambah Toko

Pada halaman tambah toko pada sisi *admin* dinas perindag. Pada halaman tersebut akan ditampilkan *form* untuk menambahkan data toko dan lokasi dari toko tersebut yang dapat dilihat pada Gambar 3.20.

**SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI
PENJUALAN OLEH – OLEH KHAS LOMBOK**

HOME	Home > Data Toko	Username admin dinas
		Logout
KELOLA TOKO	Tambah Toko	
<ul style="list-style-type: none"> • Tambah Toko • Daftar Toko 	<input type="text" value="Nama Toko"/>	<input type="text"/>
	<input type="text" value="Alamat"/>	<input type="text"/>
KELOLA ADMIN	<input type="text" value="lat"/>	<input type="text"/>
	<input type="text" value="long"/>	<input type="text"/>
	<input type="text" value="Status"/>	<input type="text"/>
	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>	
FOOTER		

Gambar 3.20 Halaman Tambah Toko.

6. Halaman Daftar Barang

Pada halaman daftar barang pada sisi *admin* toko. Pada halaman tersebut akan ditampilkan daftar barang yang sudah ditambahkan oleh *admin* toko yang dapat dilihat pada Gambar 3.21.

**SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI
PENJUALAN OLEH – OLEH KHAS LOMBOK**

HOME	Home > Barang	Username admin toko																																			
		Logout																																			
BERITA	Barang Toko																																				
BARANG TOKO	<table border="1" style="width: 100%; border-collapse: collapse; text-align: center;"> <thead> <tr> <th>No</th> <th>Nama Barang</th> <th>Foto</th> <th>Harga</th> <th>Action</th> </tr> </thead> <tbody> <tr> <td></td> <td></td> <td></td> <td></td> <td>Edit Hapus</td> </tr> </tbody> </table>		No	Nama Barang	Foto	Harga	Action					Edit Hapus																									
No	Nama Barang	Foto	Harga	Action																																	
				Edit Hapus																																	
				Edit Hapus																																	
				Edit Hapus																																	
				Edit Hapus																																	
				Edit Hapus																																	
				Edit Hapus																																	
<ul style="list-style-type: none"> • Tambah Barang • Daftar Barang 																																					
FOOTER																																					

Gambar 3.21 Halaman Daftar Barang.

7. Halaman Tambah Barang

Pada halaman tambah barang menampilkan *form input* untuk penambahan data barang yang dilakukan oleh *admin* toko yang dapat dilihat pada Gambar 3.22.

SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK	
HOME	Home > Barang Toko Username admin toko
	<input type="button" value="Logout"/>
BERITA	Tambah Barang Toko
BARANG TOKO	<input type="text" value="Nama Barang"/>
	<input type="text" value="Foto"/>
	<input type="text" value="Harga"/>
<ul style="list-style-type: none">• Tambah Barang• Daftar Barang	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
	FOOTER

Gambar 3.22 Halaman Tambah Barang.

8. Halaman kelola berita

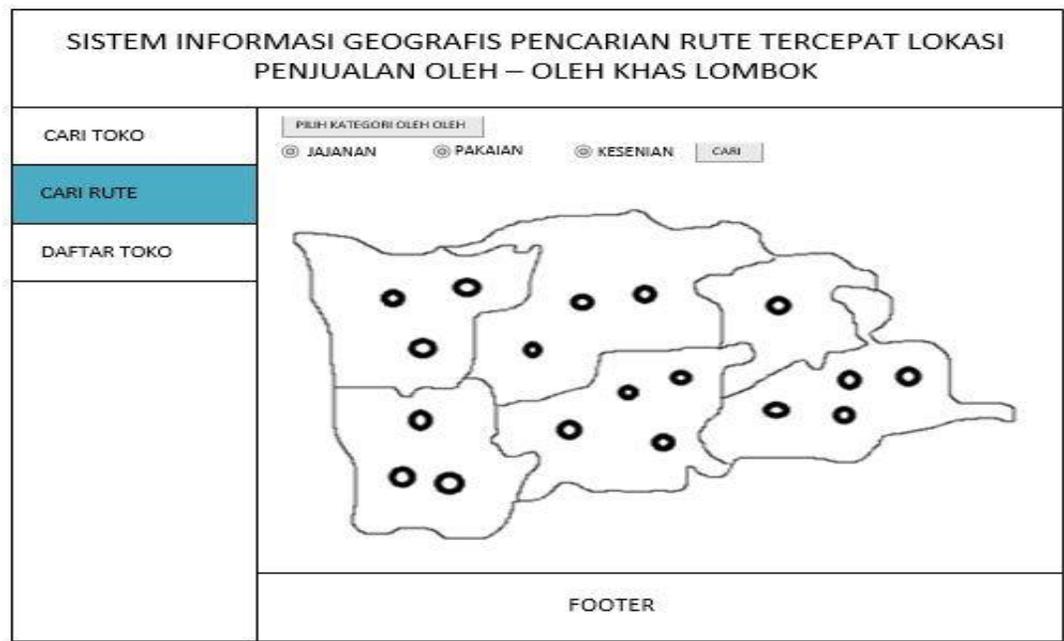
Pada halaman daftar berita menampilkan daftar dari berita yang sudah ditambahkan oleh *admin* toko yang dapat dilihat pada Gambar 3.23.



Gambar 3.23 Halaman Kelola Berita.

9. Halaman Cari Rute

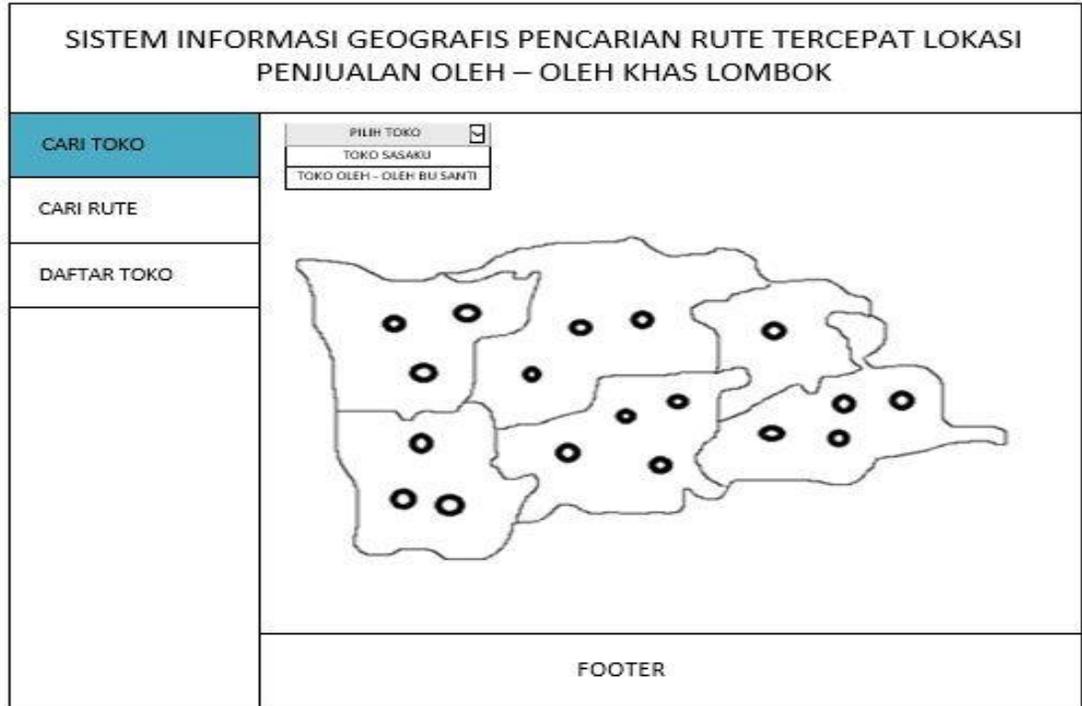
Pada halaman pencarian rute menampilkan halaman peta dengan pilihan kategori untuk fitur mencari rute terpendek sesuai dengan kategori dari toko yang diinginkan oleh user yang dapat dilihat pada Gambar 3.24.



Gambar 3.24 Halaman Cari Rute.

10. Halaman Cari Toko

Pada halaman cari toko menampilkan lokasi dari toko yang dicari dengan memasukan nama toko yang dicari pada kolom *searching* yang dapat dilihat pada Gambar 3.25.



Gambar 3.25 Halaman Cari Toko.

11. Halaman Daftar Toko

Pada halaman daftar toko menampilkan data tentang toko yang sudah terdaftar dalam sistem yang dapat dilihat pada Gambar 3.26.

SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK											
CARI TOKO	<table border="1"> <thead> <tr> <th>No</th> <th>Nama Toko</th> <th>Alamat</th> <th>Keterangan</th> <th>status</th> </tr> </thead> <tbody> <tr> <td> </td> <td> </td> <td> </td> <td> </td> <td> </td> </tr> </tbody> </table>	No	Nama Toko	Alamat	Keterangan	status					
No		Nama Toko	Alamat	Keterangan	status						
CARI RUTE											
DAFTAR TOKO											
FOOTER											

Gambar 3.26 Halaman Daftar Toko.

12. Halaman *Home* Berita

Pada halaman *home* berita menampilkan berita – berita yang dipublikasikan oleh *admin* toko yang dapat dilihat pada Gambar 3.27.

SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK	
HOME	<input type="text" value="Home"/> <input type="text" value="Username admin toko"/> <input type="button" value="Logout"/>
BERITA	<div style="border: 1px solid black; padding: 10px; text-align: center;">Slide Berita</div> <div style="border: 1px solid black; padding: 10px; text-align: center; margin-top: 10px;">Berita</div>
<ul style="list-style-type: none"> • Tambah Berita • Daftar Lokasi • Publikasi 	
BARANG TOKO	FOOTER
<ul style="list-style-type: none"> • Tambah Barang • Daftar Barang 	

Gambar 3.27 Halaman *Home* Berita.

13. Halaman Tambah Berita

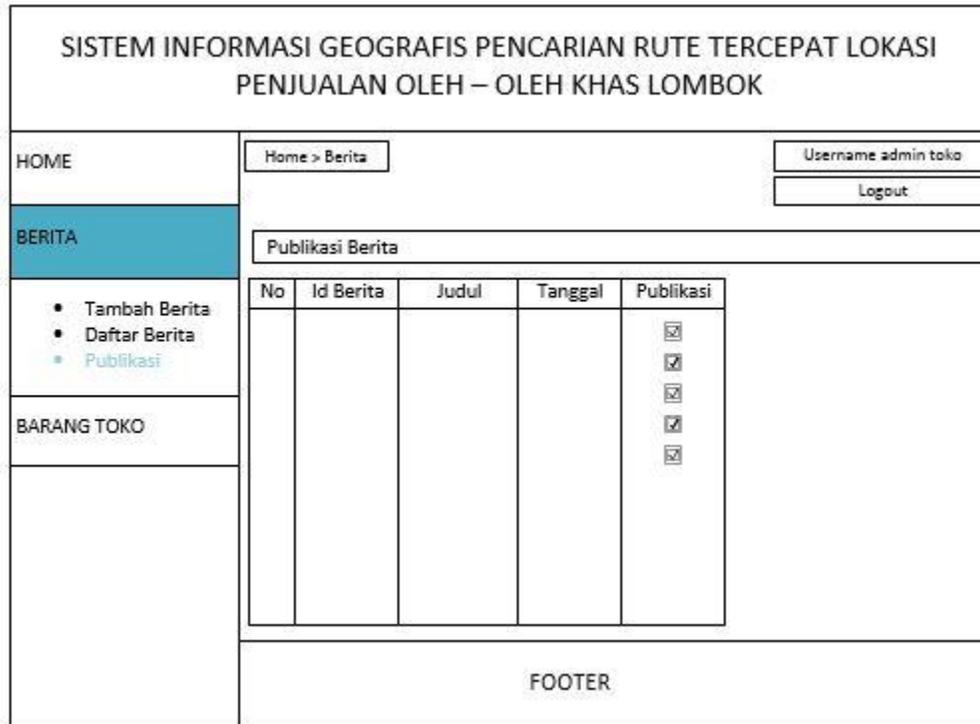
Pada halaman tambah berita pada sisi *admin* toko. Pada halaman tersebut akan menampilkan *form* pengisian untuk menambah berita yang dilakukan oleh *admin* toko yang dapat dilihat pada Gambar 3.28.

SISTEM INFORMASI GEOGRAFIS PENCARIAN RUTE TERCEPAT LOKASI PENJUALAN OLEH – OLEH KHAS LOMBOK	
HOME	Home > Berita Username admin toko Logout
BERITA	Tambah Berita
<ul style="list-style-type: none">• Tambah Berita• Daftar Berita• Publikasi	Judul Berita <input type="text"/> Isi <input type="text"/>
BARANG TOKO	<input type="button" value="Simpan"/> <input type="button" value="Reset"/> <input type="button" value="Batal"/>
FOOTER	

Gambar 3.28 Halaman Tambah Berita.

14. Halaman Publikasi Berita

Pada halaman publikasi berita yang dapat diakses oleh *admin* toko. Pada halaman tersebut ditampilkan list berita yang sudah ditambahkan dan ditentukan oleh *admin* toko untuk dipublikasikan yang dapat dilihat pada Gambar 3.29.



Gambar 3.29 Halaman Publikasi Berita.

3.2.3 Coding / Implementasi

Tahapan selanjutnya adalah proses implementasi rancangan sistem yang sudah dilakukan ke dalam bentuk bahasa pemrograman (*coding*). Pada pembuatan sistem informasi geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – Oleh khas Lombok dengan menggunakan Fungsi Way Point dan Metode *Heuristik Greedy* di Kota Mataram ini, menggunakan CodeIgniter sebagai *framework*-nya dimana dalam pengaplikasiannya menggunakan bahasa pemrograman HTML dan PHP. Seluruh perancangan yang telah dilakukan, akan ditransformasikan ke dalam bentuk M-V-C sesuai dengan kerangka dari CodeIgniter.

3.2.4 Pengujian

Setelah tahap *coding* dan implementasi selesai, maka tahap selanjutnya adalah melakukan pengujian terhadap aplikasi. Pengujian dilakukan dengan melihat keseluruhan fungsi-fungsi yang telah selesai dan membandingkan dengan rancangan yang dibuat sebelumnya. Apabila sistem yang dihasilkan telah sesuai dengan rancangan dan harapan, maka dapat melanjutkan ke tahap selanjutnya yaitu tahap evaluasi, namun apabila masih ada yang perlu diperbaiki dan diperbaharui, maka akan kembali ke tahap *coding* dan implementasi.

Tahap pengujian bertujuan untuk memastikan bahwa sistem informasi yang dibuat memenuhi tujuan yang diinginkan. Pada tahap ini informasi akan diuji dari aspek fungsionalitas dengan menggunakan metode *black box* dan untuk pengujian *user interface* akan digunakan pengujian secara subjektif dengan menggunakan *Mean Opinion Score* (MOS) dengan melibatkan beberapa orang responden dari wisatawan domestik sebanyak 15 orang, wisatawan mancanegara sebanyak 15 orang, *admin* toko sebanyak 15 orang dan *admin* dinas sebanyak 5 orang.

Serta menggunakan metode *white box* untuk melakukan pengujian terhadap hasil pengolahan aplikasi untuk penentuan rute terdekat yang akan dibandingkan dengan hasil perhitungan manual.

3.2.5 Evaluasi

Setelah dilakukan pengujian, tahapan selanjutnya adalah evaluasi. Tahapan evaluasi merupakan tahapan untuk melihat pencapaian yang telah dibangun, fungsi apa saja yang sudah berjalan dan bagaimana hasil keluarannya serta verifikasi terhadap data yang ada. Apakah fungsi berjalan sesuai dengan perhitungan dan metode yang digunakan. Selain pencapaian, hal yang dibahas pada tahap evaluasi ini adalah berbagai kendala yang ada pada, yang bisa menjadi kritik maupun saran di waktu yang akan datang.

3.2.6 Dokumentasi

Fitur sistem dan fungsi yang telah berfungsi dengan baik dan sesuai dengan rancangan yang ada, akan didokumentasikan dalam bentuk laporan. Laporan tersebut terdiri dari latar belakang, tahap perancangan, dan juga yang dihasilkan.

BAB IV HASIL DAN PEMBAHASAN

4.1 Implementasi

Pada bab ini, akan dibahas hasil dari penelitian yang dilakukan yaitu hasil implementasi “Sistem Informasi Geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – oleh khas Lombok dengan menggunakan fungsi *WayPoint* dan metode *Heuristik Greedy* Di Kota Mataram Berbasis *Web*”. Implementasi yang dilakukan telah dibuat sesuai dengan perancangan yang dijabarkan pada bab sebelumnya. Pembahasan yang akan dijelaskan meliputi implementasi *database*, implementasi *class* sistem dan implementasi *interface*. Selain itu, pada bab ini juga akan dibahas mengenai hasil sistem serta mengevaluasi sistem yang telah berjalan. Evaluasi dilakukan untuk mengetahui kekurangan sistem sehingga dapat menjadi lebih baik untuk pengembangan kedepannya.

4.1.1 Implementasi *Database*

Pada implementasi *database* hal pertama yang dilakukan adalah membuat *database* yang bernama “oleh_oleh”. Pada *database* tersebut terdiri dari lima tabel yaitu, tabel barang , tabel berita, tabel kategori, tabel toko dan tabel *user*. Struktur dari *database* “oleh_oleh” dapat dilihat pada Gambar 4.1.



Tabel	Tindakan	Baris	Jenis	Penyortiran	Ukuran	Beban
barang	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	~4	InnoDB	latin1_swedish_ci	16 KB	-
berita	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	~1	InnoDB	latin1_swedish_ci	16 KB	-
kategori	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	~3	InnoDB	latin1_swedish_ci	16 KB	-
toko	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	~5	InnoDB	latin1_swedish_ci	16 KB	-
user	Jelajahi Struktur Cari Tambahkan Kosongkan Hapus	~7	InnoDB	latin1_swedish_ci	16 KB	-
5 tabel	Jumlah	20	InnoDB	latin1_swedish_ci	80 KB	0 B

Gambar 4.1. Struktur *Database* “oleh_oleh”

Masing – masing tabel pada *database* “oleh_oleh” menyimpan data-data yang akan digunakan pada sistem informasi yang dibangun. Setiap tabel akan menyimpan data yang berbeda sesuai dengan struktur tabel yang dimiliki. Spesifikasi tabel akan dijabarkan sebagai berikut:

1. Tabel Barang

Pada struktur tabel absensi memiliki lima *field* yaitu *id_barang*, *id_toko*, *nama_barang*, *harga* dan *foto*. Tabel ini berfungsi untuk menyimpan data barang dari toko oleh – oleh yang telah terdaftar. Tabel barang berelasi dengan tabel toko. *Primary key* pada tabel absensi adalah *id_barang* dengan tipe VARCHAR(9). Rincian struktur tabel barang dapat dilihat pada Gambar 4.2.



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id_barang	varchar(9)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
2	id_toko	varchar(9)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
3	nama_barang	varchar(25)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
4	harga	int(7)		Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
5	foto	varchar(40)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya

Gambar 4.2. Struktur Tabel barang

2. Tabel Berita

Pada struktur tabel berita memiliki 6 *field*, yaitu *id_toko*, *id_berita*, *judul*, *isi*, *foto*, dan *status* dan. Tabel ini berfungsi untuk menyimpan data berita dibuat oleh toko. *Primary key* pada table mode adalah *id_toko* dengan tipe VARCHAR(9). Rincian struktur tabel berita dapat dilihat pada Gambar 4.3.



#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id_toko	varchar(9)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
2	id_berita	varchar(9)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
3	judul	varchar(20)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
4	isi	text	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
5	foto	varchar(40)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
6	status	varchar(6)	latin1_swedish_ci	Tidak	Tidak ada			Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya

Gambar 4.3. Struktur Tabel berita

3. Tabel Kategori

Pada struktur tabel kategori memiliki dua *field*, yaitu *id_kategori* dan *nama_kategori*. Tabel ini berfungsi untuk menyimpan data kategori dari jenis toko. *Primary key* pada tabel kategori adalah *id_kategori* dengan tipe VARCHAR(9). Rincian struktur tabel kategori dapat di lihat pada Gambar 4.4.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id_kategori	varchar(9)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
2	nama_kategori	varchar(10)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya

Gambar 4.4. Struktur Kategori

4. Tabel Toko

Pada struktur tabel Toko memiliki tujuh *field*, yaitu *id_toko*, *id_kategori*, *nama_toko*, *latitude*, *longitude*, *alamat* dan *foto*. Tabel ini berfungsi untuk menyimpan data toko yang sudah terdaftar pada sistem. *Primary key* pada tabel toko adalah *id_toko* dengan tipe VARCHAR(9). Rincian struktur tabel toko dapat dilihat pada Gambar 4.5.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id_toko	varchar(9)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
2	id_kategori	varchar(9)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
3	nama_toko	varchar(20)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
4	latitude	double		Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
5	longitude	double		Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
6	alamat	varchar(100)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya
7	foto	varchar(40)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Lainnya

Gambar 4.5. Struktur Tabel Toko

5. Tabel User

Pada struktur tabel *user* memiliki empat *field*, yaitu *id_admin*, *username*, *password* dan *status*. Tabel ini berfungsi untuk menyimpan data *user* yang sudah terdaftar pada sistem. *Primary key* pada tabel *user* adalah *id_admin* dengan tipe VARCHAR(9). Rincian struktur tabel *user* dapat dilihat pada Gambar 4.6.

#	Nama	Jenis	Penyortiran	Atribut	Kosong	Bawaan	Ekstra	Tindakan
1	id_admin	varchar(9)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
2	username	varchar(10)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
3	password	varchar(150)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya
4	status	varchar(2)	latin1_swedish_ci	Tidak	Tidak	Tidak ada		Ubah Hapus Kunci Utama Unik Indeks Spasial Teks penuh Lainnya

Gambar 4.6. Struktur Tabel User

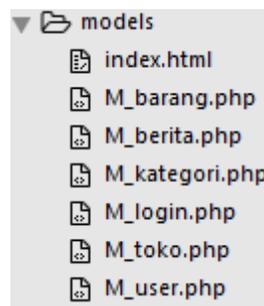
4.1.2 Implementasi Class Sistem

Berdasarkan perancangan *class diagram* yang telah dilakukan sebelumnya, implementasi *class* yang dilakukan sudah sesuai dengan perancangan tersebut. Dalam perancangan tersebut, *class* dibangun berdasarkan struktur data pada *framework CodeIgniter* yaitu terdiri dari MVC (Model, View, Controller). *Class-class* ini merupakan *class* yang berisi bahasa pemrograman yang

dibuat untuk mengimplementasikan sistem sesuai dengan perancangan yang dilakukan berupa *sourcecode* (*coding*). Hasil dari proses *coding* tersebut akan menghasilkan *interface* yang akan berinteraksi langsung dengan *user* sistem. Berikut merupakan implementasi *class* yang dilakukan dalam pengembangan Sistem Informasi Geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – oleh khas Lombok dengan menggunakan fungsi *WayPoint* dan metode *Heuristik Greedy* Di Kota Mataram Berbasis *Web*. Berikut rincian *class* yang digunakan pada sistem.

1. Implementasi *class* pada model

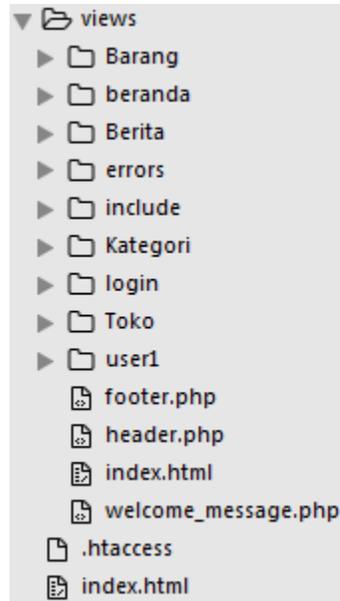
Model digunakan untuk menghubungkan sistem dengan *query-query database*. Keseluruhan fungsi yang terdapat pada *class-class* di dalam model merupakan berbagai *query* yang digunakan oleh sistem untuk memanipulasi data yang dibutuhkan yaitu *M_barang*, *M_user*, *M_login*, *M_toko*, *M_berita*, dan *M_kategori*. Berikut rincian *class* model dapat dilihat pada Gambar 4.7.



Gambar 4.7 *Class* model

2. Implementasi *class* pada *view*

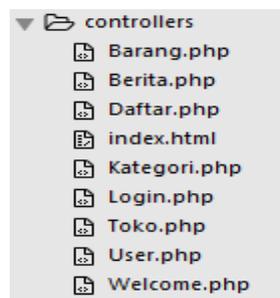
View merupakan kumpulan *source code* yang akan memberikan tampilan di dalam sistem dan akan berinteraksi dengan *user*. *View* akan menampilkan data-data yang berasal dari *model* yang telah diterjemahkan oleh *controller*. Dalam pembangunan Sistem Informasi Geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – oleh khas Lombok dengan menggunakan fungsi *WayPoint* dan metode *Heuristik Greedy* Di Kota Mataram Berbasis *Web*. Berikut rincian *class view* dapat dilihat pada Gambar 4.8.



Gambar 4.8 *Class view*

3. Implementasi *class* pada *controller*

Controller merupakan penghubung antara *model* dengan *view*. *Query-query* yang telah berjalan pada *model* akan ditampung kedalam suatu *variable* oleh *controller* yang kemudian akan didefinisikan di dalam *view*. Selain menghubungkan *view* dengan *model*, *controller* juga berisi fungsi-fungsi lainnya yang digunakan dalam membangun sistem. Berikut rincian *class controller* terlihat pada Gambar 4.9.



Gambar 4.9 *Class controller*

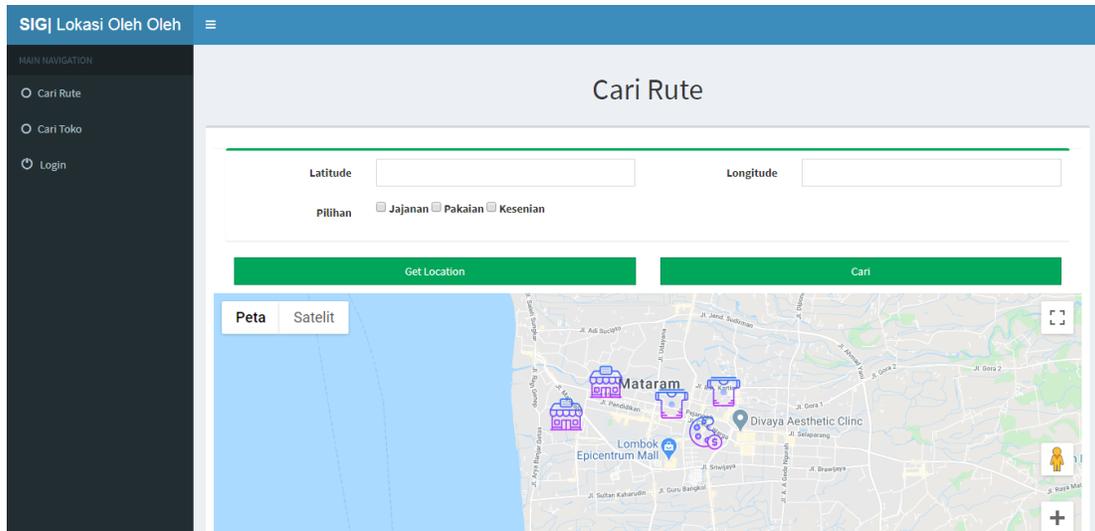
4.1.3 Implementasi *Interface* Sistem

Interface merupakan antarmuka yang akan berinteraksi secara langsung dengan pengguna sistem. Dalam implementasi *interface*, telah dikembangkan berdasarkan perancangan yang telah dibuat sebelumnya. Pada sistem ini terdapat 2 macam sisi *interface*, yaitu pada sisi *admin* dan sisi *user* sebagai berikut :

1. *Interface admin*

Pada sistem ini terdapat beberapa *interface* yang ditujukan untuk *admin* yang dapat dilihat setelah melakukan *login* yang terdiri dari halaman kelola *user*, kelola kategori, dan kelola toko. Untuk lebih jelas halaman tampilan untuk *admin* sebagai berikut :

a. Halaman *Home*

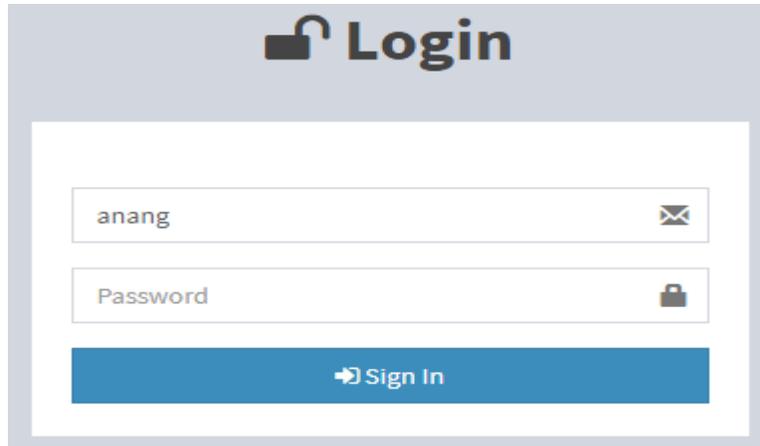


Gambar 4.10 Halaman *Home*

Halaman *Home* merupakan halaman pertama yang dapat di tampilkan ketika mengakses sistem. Pada halaman *home* ini dapat terlihat beberapa menu yang terdapat pada sistem diantaranya cari rute, cari toko dan *login*. Halaman *home* dapat dilihat pada Gambar 4.10.

b. *Login*

Login merupakan suatu *interface* yang digunakan oleh admin atau *user* untuk melakukan proses *login* sebelum dapat masuk ke dalam sistem dengan memilih *form login* di beranda. Untuk melakukan *login*, *admin* perlu memasukkan data *username* dan *password*. Berikut tampilan *login* dapat dilihat pada Gambar 4.11.



Gambar 4.11. Halaman *Login*

c. Halaman *Kelola User*

Pada halaman *kelola user* menampilkan semua *user* yang ada di sistem. Pada Halaman ini *admin* dapat menghapus dan mengelola data *user*. Halaman *kelola user* dapat dilihat pada Gambar 4.12.

ID User	Username	Password	Status		
AD1000001	adi	21232f297a57a5a743894a0e4a801fc3	AD	Edit	Hapus
AD1000002	anang	21232f297a57a5a743894a0e4a801fc3	AD	Edit	Hapus
TK1000001	aldi	21232f297a57a5a743894a0e4a801fc3	TK	Edit	Hapus
TK1000002	yana	21232f297a57a5a743894a0e4a801fc3	TK	Edit	Hapus
TK1000003	Rian	21232f297a57a5a743894a0e4a801fc3	TK	Edit	Hapus
TK1000004	Yosi	21232f297a57a5a743894a0e4a801fc3	TK	Edit	Hapus
TK1000005	Yogj	21232f297a57a5a743894a0e4a801fc3	TK	Edit	Hapus

Gambar 4.12 Halaman *Kelola User*

1. Halaman *Tambah Data User*

Pada halaman *tambah data user* menampilkan *form* untuk menambahkan *user* baru dalam sistem. Halaman *tambah data user* dapat dilihat pada Gambar 4.13.

Gambar 4.13. Halaman Tambah data *user*

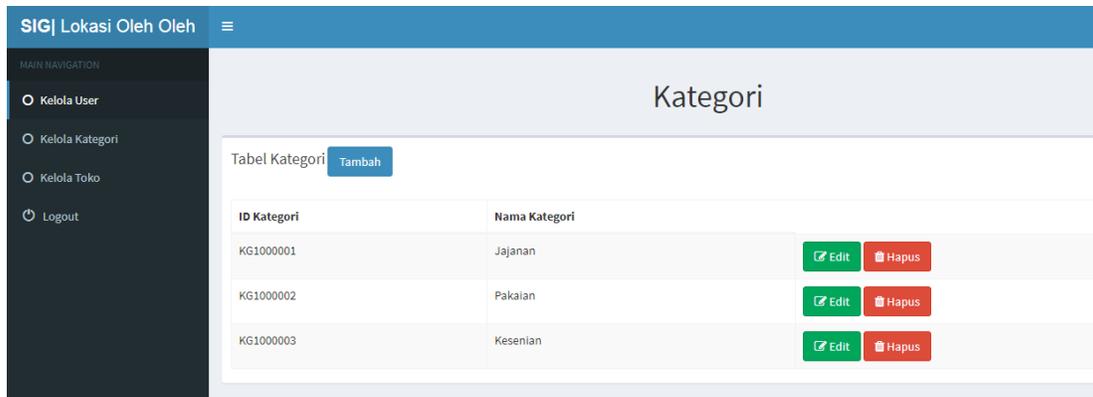
2. Halaman Edit Data *User*

Pada halaman edit data *user* menampilkan *form* data dari satu *user* yang dipilih yang dapat diganti dengan data yang lebih *valid*. Halaman edit data *user* dapat dilihat pada Gambar 4.14.

Gambar 4.14. Halaman Edit *User*

d. Halaman Kelola Kategori

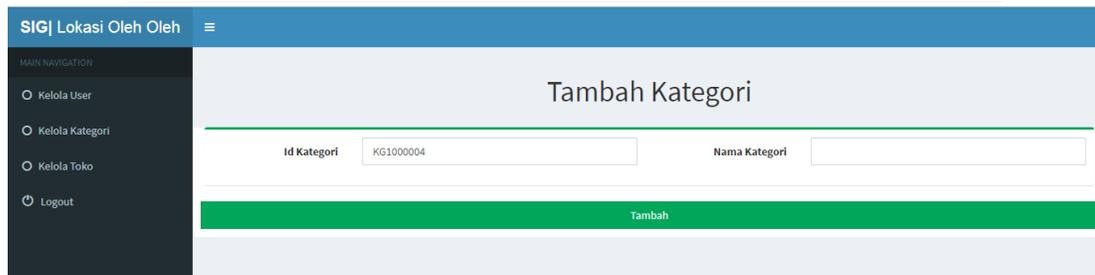
Pada halaman kelola kategori terdapat informasi tentang kategori jenis toko yang ada pada sistem. Pada Halaman ini *admin* dapat mengelola jenis kategori dari toko sesuai dengan kebutuhan yang dapat dilihat pada Gambar 4.15.



Gambar 4.15 Halaman Kelola Kategori

1. Halaman Tambah Data Kategori

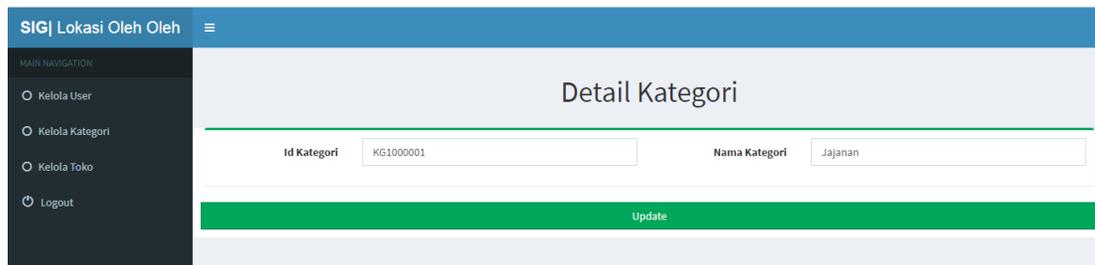
Pada halaman tambah data kategori menampilkan *form* untuk menambahkan Kategori baru dalam sistem. Halaman tambah data kategori dapat dilihat pada Gambar 4.16.



Gambar 4.16. Halaman *Import* data user

2. Halaman Edit Data Kategori

Pada halaman edit data kategori menampilkan *form* data dari satu kategori yang dipilih yang dapat diganti dengan data yang lebih *valid*. Halaman edit data kategori dapat dilihat pada Gambar 4.17.



Gambar 4.17. Halaman Edit Kategori

e. Halaman Kelola Toko

Pada halaman kelola toko, *admin* dapat mengelola data dari toko yang ingin didaftarkan kedalam sistem seperti menambahkan data, mengedit data dan menghapus data. Halaman kelola toko dapat dilihat pada Gambar 4.18.

ID Toko	Nama Toko	Latitude	Longitude	Kategori	Alamat	Foto		
TK1000001	Lombok	-8.587712285353932	116.08293771743774	Jajanan	mataram	toko_TK1000001.png	Edit	Hapus
TK1000002	Sasaku	-8.585370772467893	116.10523223876953	Pakaian	mataram	toko_TK1000002.png	Edit	Hapus
TK1000003	Rasaku	-8.591523718930041	116.11244201660156	Kesenian	mataram	toko_TK1000003.png	Edit	Hapus
TK1000004	Nut	-8.580903193435963	116.09124183654785	Jajanan	mataram	toko_TK1000004.png	Edit	Hapus
TK1000005	Lombok 2	-8.582917190937433	116.11619710922241	Pakaian	mataram	toko_TK1000005.png	Edit	Hapus

Gambar 4.18. Halaman Kelola Toko

1. Halaman Tambah Toko

Pada halaman tambah data toko menampilkan *form* untuk menambahkan data toko baru dalam sistem. Halaman tambah data toko dapat dilihat pada Gambar 4.19.

Tambah Toko

Id Toko:

Nama Toko:

Latitude:

Longitude:

Kategori:

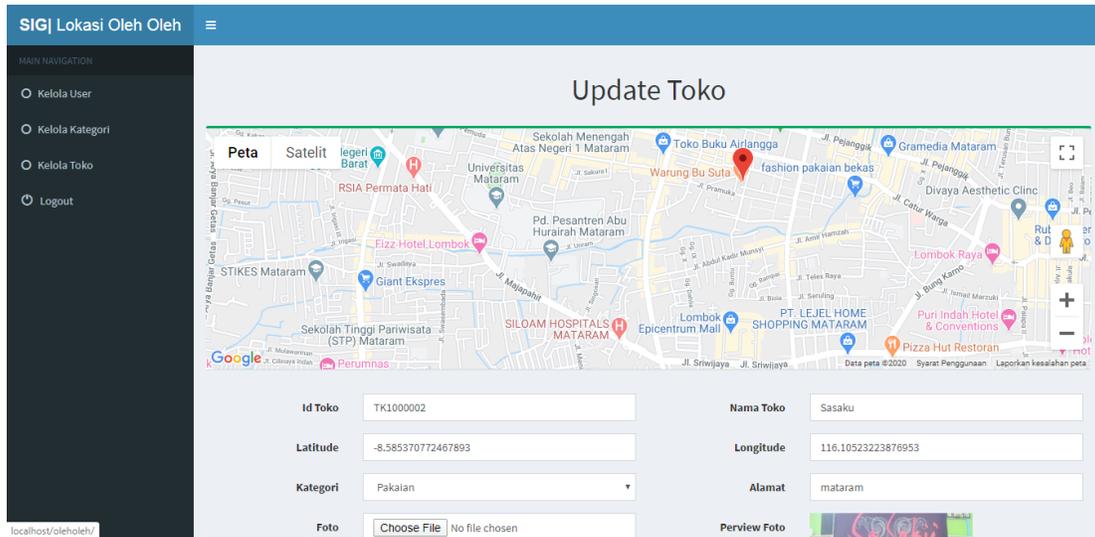
Alamat:

Foto: No file chosen

Gambar 4.19 Halaman Tambah Toko

2. Halaman Edit Data Toko

Pada halaman edit data toko menampilkan *form* data dari satu data toko yang dipilih yang dapat diganti dengan data yang lebih *valid*. Halaman edit data toko dapat dilihat pada Gambar 4.20.



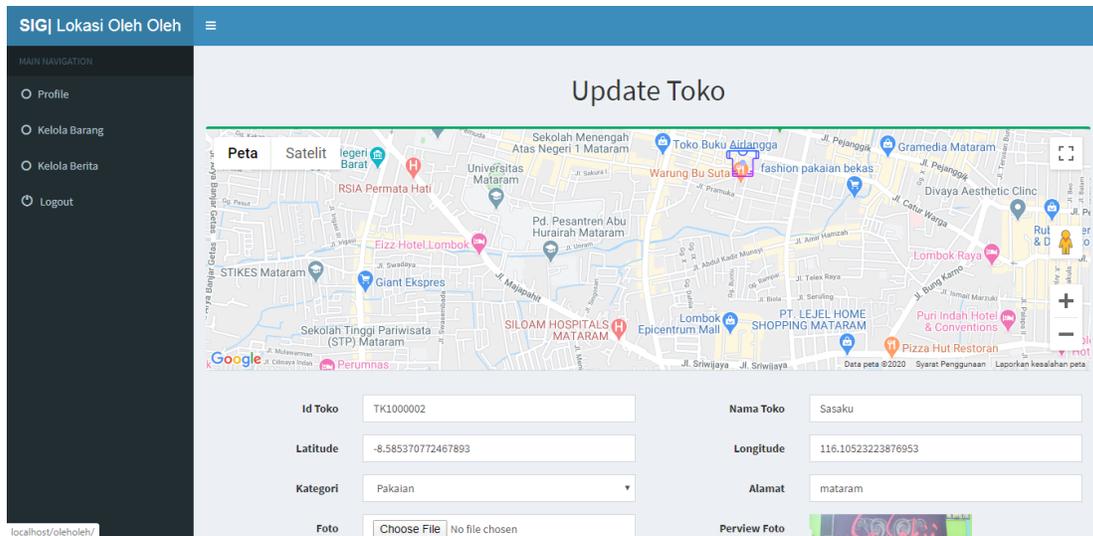
Gambar 4.20 Halaman Edit data Toko

2. Interface Admin Toko

Pada sistem ini terdapat beberapa *interface* yang ditujukan pemilik toko yang dapat dilihat sebelum dan setelah melakukan *login* yang terdiri dari halaman profile, kelola barang, dan kelola berita. Untuk lebih jelas halaman tampilan untuk *user* adalah sebagai berikut :

a. Halaman *profile*

Pada halaman *profile* menampilkan informasi dari toko yang data dapat diubah dan disesuaikan oleh *admin* toko. Pada Halaman *profile* dapat dilihat pada Gambar 4.21.



Gambar 4.21 Halaman *profile*

b. Halaman Kelola Barang

Pada halaman kelola barang, *admin* dapat mengelola data barang dari toko yang ingin dimasukkan kedalam sistem seperti menambahkan data, mengedit data dan menghapus data barang. Halaman kelola barang dapat dilihat pada Gambar 4.22.



Gambar 4.22. Halaman Kelola Barang

1. Halaman Tambah Barang

Pada halaman tambah data barang menampilkan *form* untuk menambahkan data barang baru dalam sistem. Halaman tambah data barang dapat dilihat pada Gambar 4.23.

Gambar 4.23 Halaman Tambah Barang

2. Halaman Edit Data Barang

Pada halaman edit data barang menampilkan *form* data dari satu data barang yang dipilih yang dapat diganti dengan data yang lebih *valid*. Halaman edit data barang dapat dilihat pada Gambar 4.24.

Gambar 4.24 Halaman Edit data Barang

c. Halaman Kelola Berita

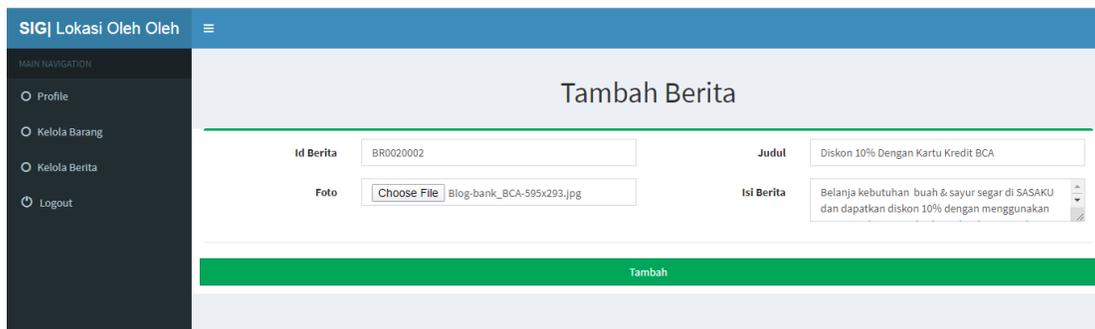
Pada halaman kelola berita, *admin* dapat mengelola data berita dari toko yang ingin dimasukkan kedalam sistem seperti menambahkan data, mengedit data dan menghapus data berita. Halaman kelola berita dapat dilihat pada Gambar 4.25.



Gambar 4.25. Halaman Kelola Berita

1. Halaman Tambah Berita

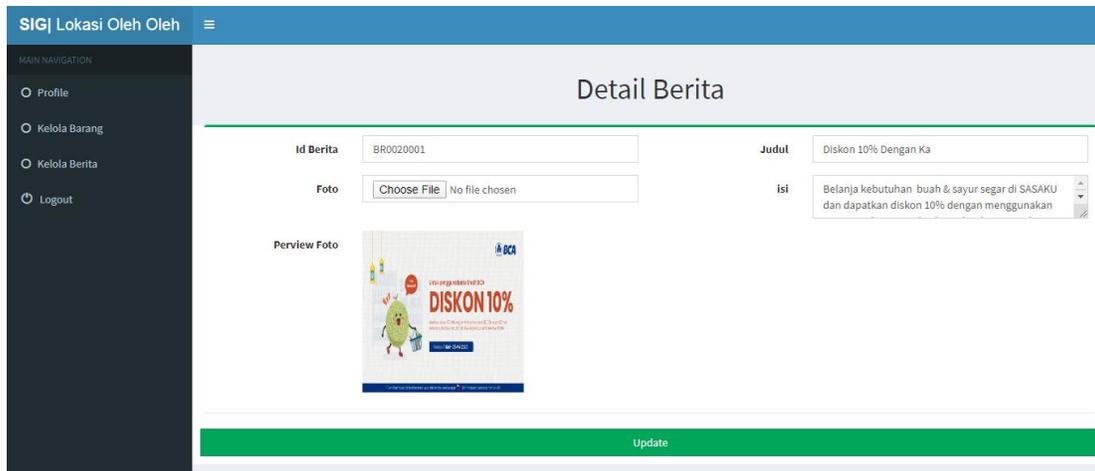
Pada halaman tambah data berita menampilkan *form* untuk menambahkan data berita baru dalam sistem. Halaman tambah data berita dapat dilihat pada Gambar 4.26.



Gambar 4.26 Halaman Tambah berita

2. Halaman Edit Data Barang

Pada halaman edit data barang menampilkan *form* data dari satu data barang yang dipilih yang dapat diganti dengan data yang lebih *valid*. Halaman edit data berita dapat dilihat pada Gambar 4.27.



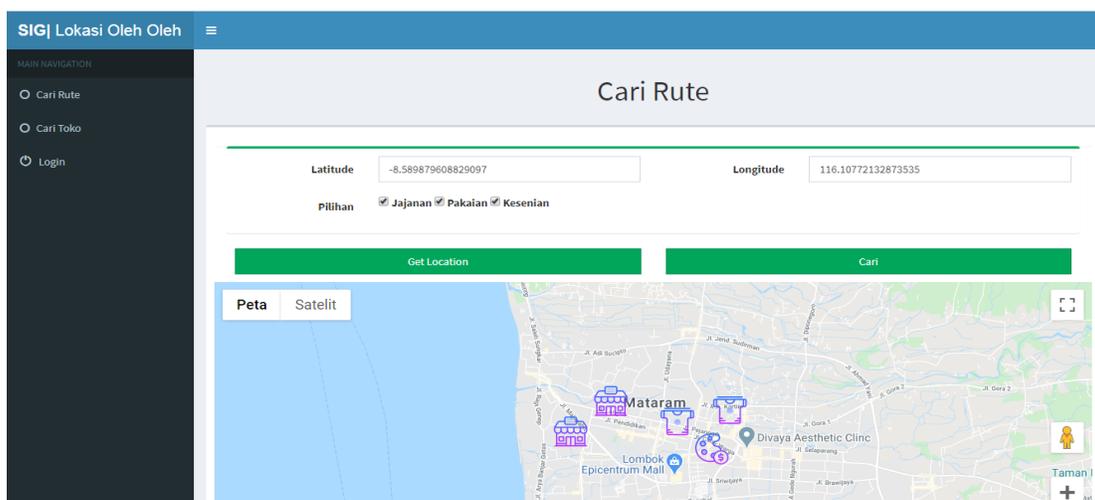
Gambar 4.27 Halaman Edit data Berita

3. *Interface User*

Pada sistem ini terdapat beberapa *interface* yang ditujukan untuk *user* yang menampilkan halaman cari rute (*Home*), dan cari toko,. Untuk lebih jelas halaman tampilan untuk *user* adalah sebagai berikut :

a. Halaman Cari Rute(*Home*)

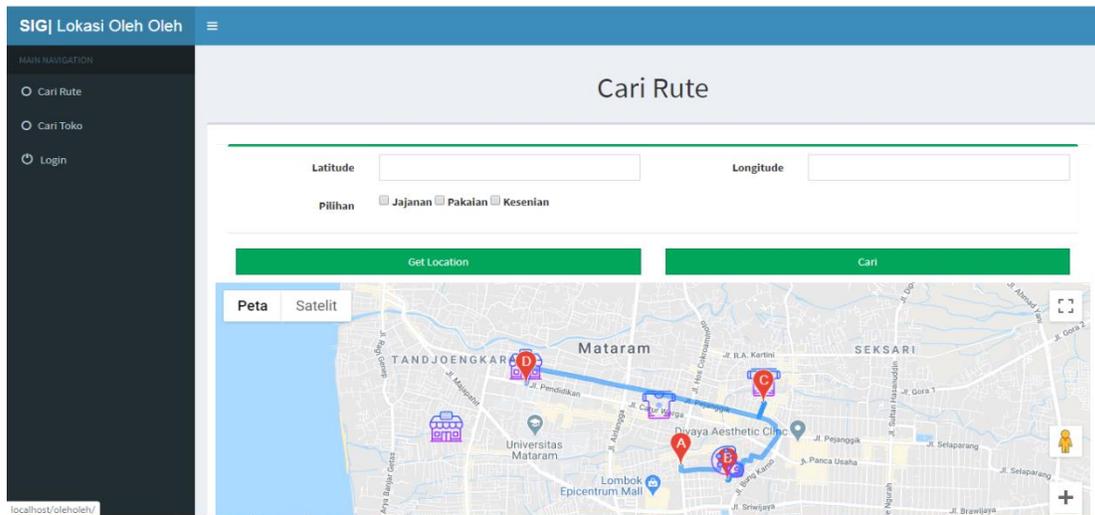
Pada halaman kelola cari rute(*Home*), *user* dapat melakukan pencarian untuk mencari toko oleh-oleh terdekat dengan memasukan jenis kategori dari toko yang diinginkan *user* sehingga sistem dapat menampilkan rute serta toko yang dicari oleh pengguna. Halaman cari rute dapat dilihat pada Gambar 4.28.



Gambar 4.28. Halaman Home

1. Halaman Hasil Cari Rute

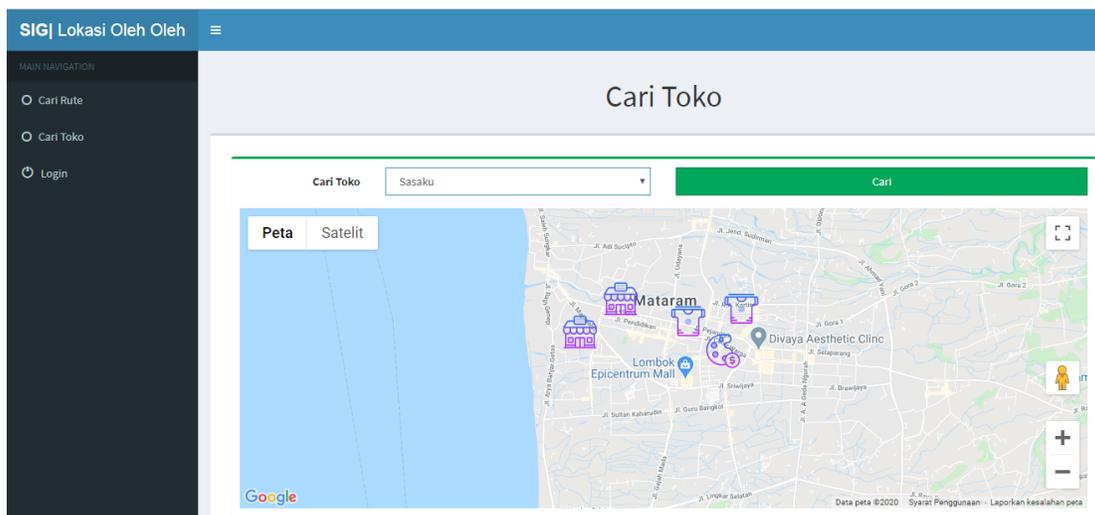
Pada halaman ini menampilkan hasil dari pencarian toko oleh-oleh dengan menggunakan metode *heuristic greedy* yang menampilkan rute dengan tujuan toko oleh – oleh sesuai dengan kategori toko yang dimasukkan oleh *user* pada sistem yang dapat dilihat pada Gambar 4.30.



Gambar 4.30 Halaman Hasil Cari Rute

b. Halaman Cari Toko

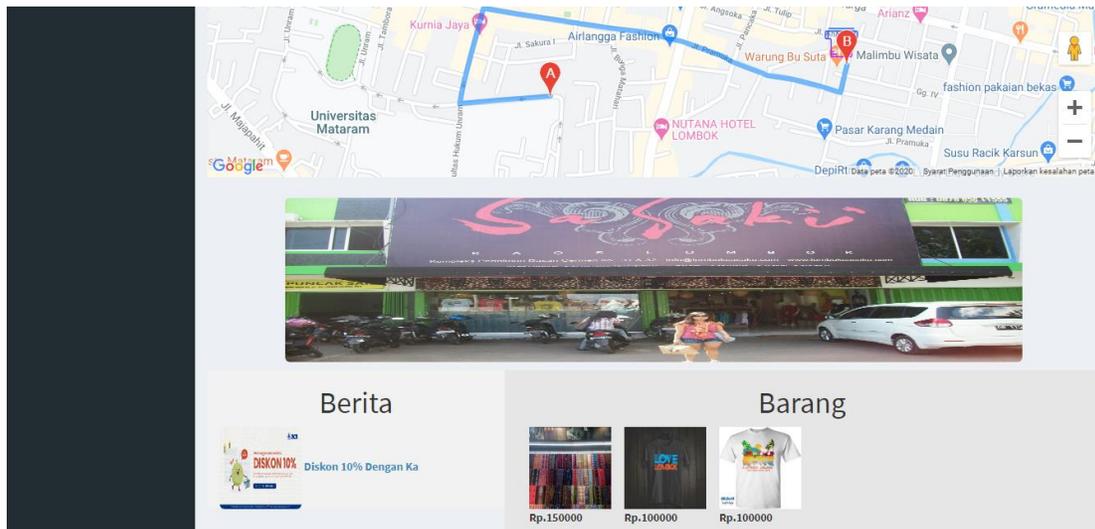
Pada halaman cari toko, *user* dapat mencari dan melihat data dari toko yang diinginkan oleh *user*. Halaman cari toko dapat dilihat pada Gambar 4.31.



Gambar 4.31. Halaman Cari Toko

1. Halaman Hasil Cari Toko

Pada halaman ini menampilkan hasil pencarian toko sesuai dengan toko dari daftar toko yang dipilih oleh *user* yang dapat dilihat pada Gambar 4.32.



Gambar 4.32. Halaman Hasil Cari Toko.

2. Halaman Berita

Pada halaman ini menampilkan halaman berita yang ada dalam data toko yang merupakan digunakan untuk melakukan promosi dan pemberitahuan lainnya yang dapat dilihat pada Gambar 4.33.



Gambar 4.33 Halaman Berita

4.2 Pengujian Sistem

Sistem akan diuji dari aspek fungsionalitas dengan menggunakan metode *black box* dan *white box*, sedangkan untuk pengujian *user interface* akan digunakan pengujian secara subjektif dengan menggunakan *Mean Opinion Score* (MOS) dengan melibatkan 40 responden, terdiri dari wisatawan, *admin* toko dan *staff* dinas.

4.2.1 Hasil Pengujian *Black Box*

Metode pengujian *black box* merupakan metode pengujian yang menguji fungsi-fungsi di dalam sistem untuk menentukan apakah fungsi-fungsi tersebut sudah berjalan sesuai harapan atau tidak. Berikut merupakan jabaran fungsi-fungsi di Sistem Informasi Geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – oleh khas Lombok dengan menggunakan fungsi *WayPoint* dan metode *Heuristik Greedy* Di Kota Mataram Berbasis *Web* yang akan diuji menggunakan metode *black box*:

1. Pengujian Fungsi pada *Admin* Dinas

a. Fungsi *Login Admin*

Fungsi *login admin* dilakukan untuk menguji apakah fungsi yang berjalan pada halaman *login admin* sudah berjalan dengan baik dan benar pada pengujian fungsi *login Admin* terdapat 2 skenario pengujian. Pengujian fungsi *login admin* akan dijelaskan pada Tabel 4.1.

Tabel 4.1 Pengujian *black box* pada fungsi *login admin*.

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
<i>Form username</i> dan <i>password</i> di isi dengan data yang tidak sesuai	Proses <i>login</i> tidak berhasil dan akan dirujuk kembali ke halaman <i>login admin</i>	Sesuai	Valid
<i>Form username</i> dan <i>password</i> di isi dengan data yang benar dan terdapat di dalam database.	Proses <i>login</i> berhasil <i>admin</i> akan memasuki sistem dan akan dirujuk ke halaman beranda <i>admin</i> .	Sesuai	Valid

Pada pengujian pertama, terdapat beberapa variasi dalam input data, pertama apabila *form username* diisi dan *form password* di kosongkan, kedua apabila *form password* diisi dan *form username* di kosongkan dan yang ketiga apabila *form username* dan *password* dikosongkan. Hasil dari ketiga kondisi tersebut adalah proses *login* tidak berhasil dikarenakan tidak lengkapnya data dan akan dirujuk kembali ke halaman *login admin*.

b. Fungsi pada Kelola *User*

Fungsi Kelola *User* pada sistem ini adalah untuk melihat data semua yang ada di sistem serta bisa melakukan edit, hapus dan tambah data *user* ke dalam sistem. Pengujian terhadap fungsi kelola *user* akan dijelaskan pada Tabel 4.2.

Tabel 4.2 Pengujian *black box* Kelola *User*

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Pada halaman kelola <i>user</i> dapat dilakukan <i>edit</i> terhadap data <i>user</i> yang sudah tersedia.	Data <i>user</i> sudah terdaftar dapat di <i>edit</i> sesuai dengan kebutuhan.	Sesuai	Valid
Pada halaman Kelola <i>user</i> dapat melakukan hapus <i>user</i>	Data <i>user</i> yang sudah terdaftar dapat di hapus sesuai dengan kebutuhan	Sesuai	Valid
Pada halaman Kelola <i>user</i> dapat melakukan tambah data <i>User</i> ke dalam sistem	Sistem dapat melakukan tambah data <i>User</i> ke dalam sistem	Sesuai	Valid

c. Fungsi Kelola Toko

Fungsi kelola toko pada sistem ini adalah untuk melihat data semua yang ada di sistem serta bisa melakukan edit, hapus dan tambah data toko ke dalam sistem. Pengujian terhadap fungsi kelola toko akan dijelaskan pada Tabel 4.3.

Tabel 4.3 Pengujian *black box* pada fungsi Kelola Toko

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Pada halaman kelola toko dapat dilakukan <i>edit</i> terhadap data toko yang sudah tersedia.	Data toko sudah terdaftar dapat di <i>edit</i> sesuai dengan kebutuhan.	Sesuai	Valid
Pada halaman Kelola toko dapat melakukan hapus toko	Data toko yang sudah terdaftar dapat di hapus sesuai dengan kebutuhan	Sesuai	Valid
Pada halaman Kelola toko dapat melakukan tambah data toko ke dalam sistem	Sistem dapat melakukan tambah data toko ke dalam sistem	Sesuai	Valid

d. Fungsi Kelola Kategori

Fungsi kelola kategori pada sistem ini adalah untuk melihat data semua yang ada di sistem serta bisa melakukan edit, hapus dan tambah data kategori ke dalam sistem. Pengujian terhadap fungsi kelola kategori akan dijelaskan pada Tabel 4.4.

Tabel 4.4 Pengujian *black box* pada fungsi Kelola Kategori

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Pada halaman kelola kategori dapat dilakukan <i>edit</i> terhadap data kategori yang sudah tersedia.	Data kategori sudah terdaftar dapat di <i>edit</i> sesuai dengan kebutuhan.	Sesuai	Valid
Pada halaman Kelola kategori dapat melakukan hapus kategori	Data kategori yang sudah terdaftar dapat di hapus sesuai dengan kebutuhan	Sesuai	Valid

Pada halaman Kelola kategori dapat melakukan tambah data kategori ke dalam sistem	Sistem dapat melakukan tambah data kategori ke dalam sistem	Sesuai	Valid
---	---	--------	-------

2. Pengujian Fungsi pada *Admin Toko*

a. Fungsi Edit *Profile*

Fungsi Edit *Profile* dilakukan untuk menguji apakah fungsi yang berjalan pada halaman *profile* sudah berjalan dengan baik dan benar pada pengujian fungsi *profile Admin toko* terdapat 2 skenario pengujian. Pengujian fungsi Edit *Profile* akan dijelaskan pada Tabel 4.5.

Tabel 4.5 Pengujian *black box* pada fungsi Edit *Profile*.

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Data tidak dilakukan perubahan	Data dari <i>Profile</i> toko yang ada tidak ada perubahan	Sesuai	Valid
Dilakukan perubahan terhadap data <i>profile</i> dari data yang sudah ada.	Data dari <i>Profile</i> toko menampilkan data sesuai data yang dilakukan perubahan.	Sesuai	Valid

b. Fungsi pada Kelola Barang

Fungsi Kelola barang pada sistem ini adalah untuk melihat data semua yang ada di sistem serta bisa melakukan edit, hapus dan tambah data barang ke dalam sistem. Pengujian terhadap fungsi kelola barang akan dijelaskan pada Tabel 4.6.

Tabel 4.6 Pengujian *black box* Kelola Barang

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Pada halaman kelola barang dapat dilakukan <i>edit</i> terhadap data barang yang sudah tersedia.	Data barang sudah terdaftar dapat di <i>edit</i> sesuai dengan kebutuhan.	Sesuai	Valid
Pada halaman Kelola barang dapat melakukan hapus barang	Data barang yang sudah terdaftar dapat di hapus sesuai dengan kebutuhan	Sesuai	Valid
Pada halaman Kelola barang dapat melakukan tambah data barang ke dalam sistem	Sistem dapat melakukan tambah data barang ke dalam sistem	Sesuai	Valid

c. Fungsi Kelola Berita

Fungsi kelola berita pada sistem ini adalah untuk melihat data semua yang ada di sistem serta bisa melakukan edit, hapus dan tambah data berita ke dalam sistem. Pengujian terhadap fungsi kelola berita akan dijelaskan pada Tabel 4.7.

Tabel 4.7 Pengujian *black box* pada fungsi Kelola Berita

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Pada halaman kelola berita dapat dilakukan <i>edit</i> terhadap data berita yang sudah tersedia.	Data berita sudah terdaftar dapat di <i>edit</i> sesuai dengan kebutuhan.	Sesuai	Valid
Pada halaman Kelola berita dapat melakukan hapus berita	Data berita yang sudah terdaftar dapat di hapus sesuai dengan kebutuhan	Sesuai	Valid

Pada halaman Kelola berita dapat melakukan tambah data berita ke dalam sistem	Sistem dapat melakukan tambah data berita ke dalam sistem	Sesuai	Valid
---	---	--------	-------

d. Fungsi Kelola Kategori

Fungsi kelola kategori pada sistem ini adalah untuk melihat data semua yang ada di sistem serta bisa melakukan edit, hapus dan tambah data kategori ke dalam sistem. Pengujian terhadap fungsi kelola kategori akan dijelaskan pada Tabel 4.8.

Tabel 4.8 Pengujian *black box* pada fungsi Kelola Kategori

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Pada halaman kelola kategori dapat dilakukan <i>edit</i> terhadap data kategori yang sudah tersedia.	Data kategori sudah terdaftar dapat di <i>edit</i> sesuai dengan kebutuhan.	Sesuai	Valid
Pada halaman Kelola kategori dapat melakukan hapus kategori	Data kategori yang sudah terdaftar dapat di hapus sesuai dengan kebutuhan	Sesuai	Valid
Pada halaman Kelola kategori dapat melakukan tambah data kategori ke dalam sistem	Sistem dapat melakukan tambah data kategori ke dalam sistem	Sesuai	Valid

3. Pengujian Fungsi pada *User*

a. Fungsi Cari Rute

Fungsi Cari Rute dilakukan untuk menguji apakah fungsi yang berjalan pada halaman Cari Rute sudah berjalan dengan baik dan benar pada pengujian fungsi Cari Rute terdapat 2 skenario pengujian. Pengujian fungsi Cari Rute akan dijelaskan pada Tabel 4.9.

Tabel 4.9 Pengujian *black box* pada fungsi Cari Rute.

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
<i>Form</i> pencarian di isi dengan data yang tidak sesuai	Proses Cari Rute tidak berhasil dan akan dirujuk untuk mengisi <i>field</i> yang ingin dicari	Sesuai	Valid
<i>Form</i> pencarian di isi dengan data yang sesuai	Proses Cari Rute berhasil dan akan dirujuk ke halaman Hasil Cari Rute.	Sesuai	Valid

b. Fungsi Cari Toko

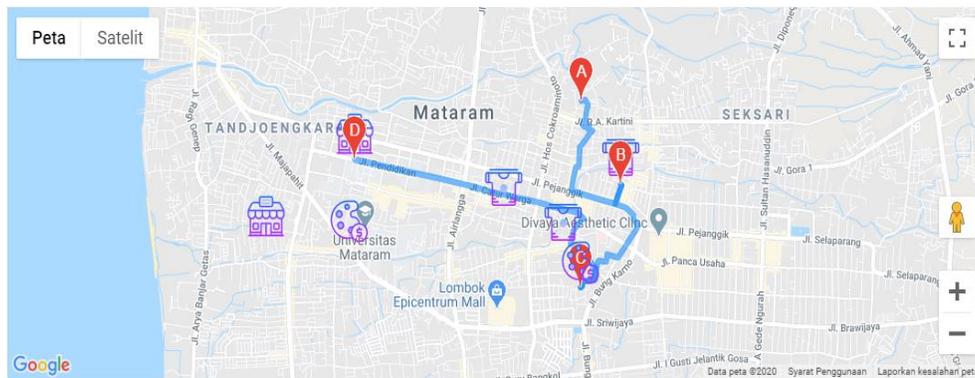
Fungsi Cari Toko dilakukan untuk menguji apakah fungsi yang berjalan pada halaman Cari Toko sudah berjalan dengan baik dan benar pada pengujian fungsi Cari Toko terdapat 2 skenario pengujian. Pengujian fungsi Cari Toko akan dijelaskan pada Tabel 4.10.

Tabel 4.10 Pengujian *black box* pada fungsi Cari Toko

Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Kesimpulan
Data tidak dilakukan perubahan	Proses Cari Toko tidak berhasil dan akan dirujuk untuk mengisi <i>field</i> yang ingin dicari	Sesuai	Valid
<i>Form</i> pencarian di isi dengan data yang sesuai.	Proses Cari Toko berhasil dan akan dirujuk ke halaman Hasil Cari Toko.	Sesuai	Valid

4.2.2 Hasil Pengujian *White Box*.

Metode pengujian *white* metode perancangan *test case* yang menggunakan struktur kontrol dari perancangan prosedural untuk mendapatkan *test case*. Dengan menggunakan metode *white box*, analisis sistem akan dapat memperoleh *test case* yang menjamin seluruh *independent path* dieksekusi paling sedikit satu kali. *Independent path* adalah jalur dalam program yang menunjukkan paling sedikit satu kumpulan proses ataupun kondisi baru. Berikut fungsi di Sistem Informasi Geografis Pencarian Rute Tercepat Lokasi Penjualan Oleh – oleh khas Lombok dengan menggunakan fungsi *WayPoint* dan metode *Heuristik Greedy* Di Kota Mataram Berbasis *Web* yang akan diuji menggunakan metode *white box*:



Gambar 4.34. Gambar Hasil fungsi pencarian rute

Tabel 4.11 Tabel Ketetangaan Graf Masalah.

Simpul	KG	J	P	K	J	P	P	K
	SL	1	2	3	4	5	6	7
SL	0	5.1km	2km	2.5km	3.3km	1.8km	1.9km	4.9km
1	x	x	x	x	x	x	x	x
2	x	x	x	x	x	x	x	x
3	0	4.9km	0	0	3.4km	0	0	0
4	x	x	x	x	x	x	x	x
5	0	5.4km	0	1.6km	3.7km	0	0	4.8km
6	x	x	x	x	x	x	x	x
7	x	x	x	x	x	x	x	x

Dengan Menggunakan algoritma *Greedy* pada permasalahan pencarian rute terpendek berdasarkan kategori yang dipilih dapat dipecahkan dengan memperhatikan tabel 4.11 graf permasalahan dimana data yang ada merupakan data dari jarak tempuh sebenarnya dari titik lokasi pengguna menuju lokasi toko ataupun titik lokasi toko menuju titik lokasi toko lainnya yang didapatkan dengan menggunakan *google maps* untuk mendapat jarak tempuh yang paling pendek sebagai jalur yang digunakan. Berikut cara penyelesaiannya :

1. Pada awalnya node SL akan dijadikan node pertama yang digunakan sebagai titik awal.
2. Lalu dari SL dicari jarak tempuh yang paling dekat dengan node yang lainnya dimana didapat yaitu node 5 dengan jarak 1.8 dan jadikan node 5 sebagai node awal.
3. Lalu dari node 5 dicari jarak tempuh yang paling dekat dengan node lainnya dengan kategori selain Pakaian yang merupakan kategori node 5, dan didapatkan node 3 dengan kategori kesenian dengan jarak tempuh 1.6 serta jadikan node 3 sebagai titik awal.
4. Lalu dari node 3 dicari jarak tempuh yang paling dekat dengan node yang berkategori jajanan dan didapatkan node 4 dengan jarak tempuh 3.4.
5. Jadi bila disimpulkan maka rute terpendek yang didapatkan berdasarkan *independent path* pada sistem adalah **SL-5-3-4** dan hasil dari fungsi pada sistem dengan hitungan manual dengan pengujian *white box* bernilai benar/ valid.

4.2.3 Hasil Pengujian Kuesioner dengan Metode *Mean Opinion Score (MOS)*.

Tujuan dari pengujian ini adalah mengetahui bagaimana kualitas sistem dilihat dari sisi pengguna (*admin* dan *user*). Pengujian ini dilakukan oleh 30 responden yang dipilih secara acak terdiri dari 20 wisatawan, 5 *admin* toko dan 5 *staff* dinas. Pengujian dilakukan dengan mempresentasikan program kepada responden tersebut, responden diberikan kesempatan untuk mencoba sistem secara langsung, kemudian setelah itu, responden mengisi kuisisioner untuk menilai sistem.

Hasil dari jawaban responden nantinya akan dihitung dan ditarik kesimpulan mengenai hasil pengujian sistem. Kuesioner pengujian sistem yang diberikan untuk *admin* toko dan *admin* dinas terdiri dari terdiri dari 5 pernyataan, yaitu:

1. Apakah *website* yang dibuat menarik untuk dilihat?
2. Apakah sistem pada *website* yang dibuat *user friendly*(mudah digunakan)?
3. Apakah sistem dapat memberikan informasi berdasarkan titik persebaran toko oleh-oleh di Kota Mataram?
4. Apakah sistem ini dapat membantu *admin* dalam hal pengelelolaan data toko oleh-oleh di Kota Mataram?
5. Apakah sistem ini bermanfaat dan dapat membantu pada Dinas Perindag dan Toko oleh-oleh di Kota Mataram?

Kuesioner pengujian sistem yang diberikan untuk wisatawan atau *user* terdiri dari terdiri dari 5 pernyataan, yaitu:

1. Apakah *website* yang dibuat menarik untuk dilihat?
2. Apakah sistem pada *website* yang dibuat *user friendly*(mudah digunakan)?
3. Apakah sistem dapat memberikan informasi mengenai data toko oleh-oleh yang diinginkan?
4. Apakah Sistem ini dapat membantu dalam pencarian rute tercepat untuk toko oleh-oleh ?
5. Apakah Sistem ini dapat membantu dalam hal pemetaan dan informasi tentang lokasi oleh-oleh ?

Lalu dari pernyataan tersebut, responden diminta untuk menjawab dengan nilai-nilai tertentu, diantaranya:

1. 1 = Sangat tidak setuju.
2. 2 = Tidak setuju.
3. 3 = Cukup.
4. 4 = Setuju.
5. 5 = Sangat setuju

Selanjutnya akan dihitung persentase jawaban yang diberikan pengguna di setiap pertanyaan.

Rumus untuk menghitung persentase nilai adalah sebagai berikut :

$$\text{Persentase} : J / N * 100\%$$

Di mana:

J : total nilai jawaban responden pada setiap nomor.

N : Jumlah responden.

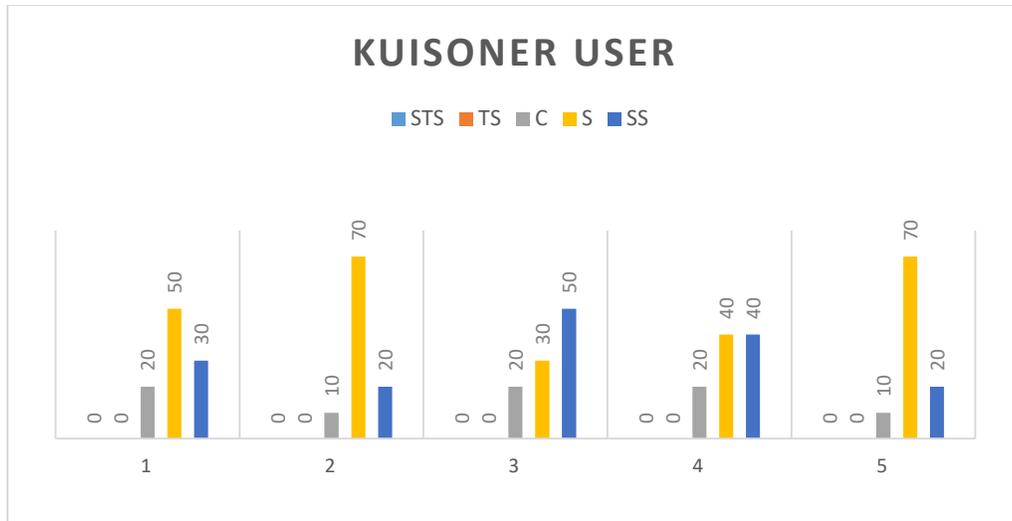
Untuk menghitung persentase rata rata dari keseluruhan jawaban, perhitungan yang dilakukan adalah menggunakan rumus sebagai berikut:

$$\text{Rata - rata} = (\text{persentase ke-1} + \dots + \text{persentase ke -5}) / 5$$

Dengan menggunakan rumus perhitungan diatas perhitungan untuk hasil jawaban dari setiap pertanyaan kuisisioner diatas dapat dilihat pada rincian table 4.12 dan table 4.13.

Tabel 4.12 Hasil pengujian kuisisioner dari sistem (sisi *User*)

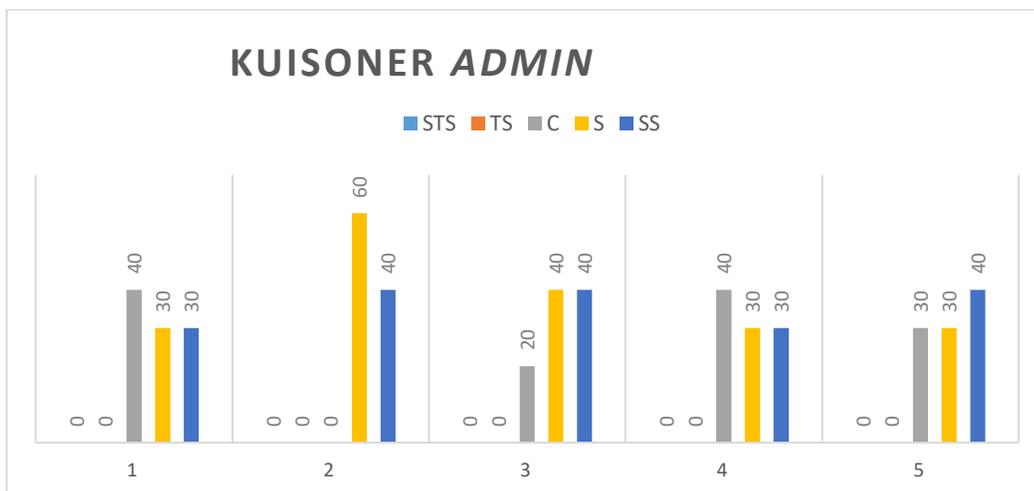
Pertanyaan	Kategori Jawaban					Persentase(%)				
	STS	TS	C	S	SS	STS	TS	C	S	SS
1	0	0	4	10	6	0	0	20	50	30
2	0	0	2	14	4	0	0	10	70	20
3	0	0	4	6	10	0	0	20	30	50
4	0	0	4	8	8	0	0	20	40	40
5	0	0	2	14	4	0	0	10	70	20
Rata - Rata						0	0	16	52	32



Gambar 4.35. Grafik persentase jawaban kuisoner untuk setiap pertanyaan

Tabel 4.13 Hasil pengujian kuisoner dari sistem (sisi Admin)

Pertanyaan	Kategori Jawaban					Persentase(%)				
	STS	TS	C	S	SS	STS	TS	C	S	SS
1	0	0	4	3	3	0	0	40	30	30
2	0	0	0	6	4	0	0	0	60	40
3	0	0	2	4	4	0	0	20	40	40
4	0	0	4	3	3	0	0	40	30	30
5	0	0	3	3	4	0	0	30	30	40
Rata - Rata						0	0	26	38	36



Gambar 4.36. Grafik persentase jawaban kuisoner untuk setiap pertanyaan

Dengan menggunakan rumus persentase, perhitungan hasil jawaban responden dapat di lihat pada Tabel 4.12 dan Tabel 4.13, sangat setuju dengan rata – rata persentasenya adalah 32%, setuju dengan jumlah rata – rata 52%, sedangkan cukup setuju dengan jumlah rata – rata persentasenya adalah 16% dari jumlah 20 responden, sedangkan pada sisi admin: sangat setuju dengan rata – rata persentasenya adalah 36%, setuju dengan jumlah rata – rata 38%, sedangkan cukup setuju dengan jumlah rata – rata persentasenya adalah 26% dari jumlah 10 responden.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan penelitian dan pengembangan sistem yang dilakukan terdapat beberapa hal yang dapat disimpulkan, antara lain:

1. Berdasarkan hasil pengujian sistem dari sisi pengguna disimpulkan bahwa sistem ini dapat membantu pengguna untuk mencari toko oleh-oleh khas lombok dan menampilkan jalur yang dilalui sesuai kategori toko yang dicari dalam bentuk peta *google maps*.
2. Berdasarkan hasil pengujian sistem dari sisi admin toko disimpulkan bahwa sistem ini dapat digunakan oleh pemilik toko sebagai media promosi untuk pemasaran produk yang diperdagangkan.
3. Berdasarkan hasil pengujian sistem dari sisi admin dinas disimpulkan sistem ini dapat digunakan untuk pendataan lokasi oleh-oleh khas Lombok dalam bentuk kordinat titik lokasi dalam *google maps*.
4. Berdasarkan hasil pengujian sistem ini dapat menampilkan rute pencarian toko dengan menggunakan titik lokasi pengguna dan 20 titik lokasi oleh – oleh dengan keberhasilan 100%.
5. Berdasarkan hasil pengujian MOS yang telah dilakukan menunjukkan bahwa rata – rata penilain responden *user* dari jumlah 20 responden : sangat setuju dengan rata – rata persentasenya adalah 32%, setuju dengan jumlah rata – rata 52%, sedangkan cukup setuju dengan jumlah rata – rata persentasenya adalah 16% dan *admin* dari jumlah 10 responden: sangat setuju dengan rata – rata persentasenya adalah 36%, setuju dengan jumlah rata – rata 38%, sedangkan cukup setuju dengan jumlah rata – rata persentasenya adalah 26%.

5.2 Saran

Terdapat beberapa saran yang dapat disampaikan untuk mengembangkan sistem informasi geografis ini yaitu:

1. Untuk pengembangan lainnya diharapkan sistem ini dapat berjalan di *platform mobile*.
2. Untuk pengembangan selanjutnya diharapkan sistem ini dapat diintegrasikan ke sistem-sistem yang ada pada dinas kota Mataram.
3. Untuk pengembangan selanjutnya diharapkan sistem ini dapat dikembangkan dalam cakupan wilayah yang lebih luas.

DAFTAR PUSTAKA

- [1] Eko P. Julianto, “*Sistem Informasi Geografis Pencarian Kuliner Terdekat di Kota Mataram*”, 2017.
- [2] A. Wahyudinur, “*Pencarian Jalur Terpendek Pengiriman Barang Menggunakan Algoritma A* Studi Kasus Kantor Pos Besar Medan*”, 2016.
- [3] Cahyo P. Adi, “*Implementasi Algoritma Greedy Pada Layanan Taksi Wisata Berbasis Web*”, 2014.
- [4] Cahyany A. Ayu, “*Sistem Informasi Geografis (SIG) Pemetaan Rute Terpendek UMKM Kota Medan Berbasis Web Menggunakan Algoritma Bellman-Ford*”, 2016.
- [5] Setiadi. Teguh, “*Pencarian Rute Terpendek Menuju Pom Bensin Pada Kota Semarang Menggunakan Algoritma Dijkstra Berbasis Sistem Informasi Geografis (GIS)*”, 2017.
- [6] Febrinus N. Jimmy, “*Pemetaan Informasi Sarana Kesehatan Masyarakat Serta Penyajian Rute Terdekat Menuju Lokasi Sarana Pelayanan Kesehatan Pada Wilayah Kota Berbasis Online*”, 2017.
- [7] Primadasa. Yogi, “*Pencarian Rute Terpendek Menggunakan Algoritma Dijkstra Pada SIG Berbasis Web Untuk Distribusi Minuman (Studi Kasus PT. Coca-Cola Kota Padang)*”, 2015.
- [8] Fajri. Albes., Purwandari. Endina P., dan Coastera. Funny F, “*Implementasi Sistem Informasi Geografis Untuk Pencarian Indekos Dengan Jarak Terpendek Menggunakan Metode Algoritma Dijkstra*”, 2018.
- [9] Setiawan. Kiki., Supriyadin., Santoso. Imam., dan Buana. Roy, “*Menghitung Rute Terpendek Menggunakan Algoritma A* Dengan Fungsi Euclidean Distance*”, 2018.
- [10] Fahmi. Bahtiar, “*Penentuan Lokasi Wisata Menggunakan Algoritma Floyd-Marshall di Kabupaten Banjarnegara Berbasis Web-GIS*”, 2014.
- [11] R. Munir. , *Diktat Kuliah IF2251 Strategi Algoritmik*, Bandung, 2005.

- [12] K. C. Laudon, *Management Informaton System*. New York, United States of America: Prentise Hall, 2005.
- [13] I. W. E. Swastikayana, *Sistem Informasi Geografis Berbasis Web untuk Pemetaan Pariwisata Kabupaten Gianyar, Fakultas Teknologi Industri, Universitas Pembangunan Nasional*. Yogyakarta: Veteran, 2011.
- [14] Anon, *Sistem Informasi Geografis*. Jakarta, 2003.
- [15] A. Shodiq, *Tutorial Dasar Pemrograman Google Maps API*. Bogor, Indonesia, 2009.
- [16] B. Wang, S., dan Yi, *Location Based Service for Mobiles Technologies and Standars, In Proceedings of the IEEE International Conference on Communication (ICC)*. Beijing, China, 2008.
- [17] S. Crosier, *ArcGIS Geocoding in ArcGIS, ESRI*. Amerika Serikat, 2004.
- [18] Hernawati Kuswari, *Aplikasi Perhitungan Jarak Antara Dua Waypoint Pada Google Maps*, 2012.
- [19] I. Simarmata, J., dan Paryudi, *Basis Data*. Yogyakarta: Andi, 2010.
- [20] D. Widodo, Prabowo. P, *Pemodelan Sistem Berorientasi Obyek dengan UML*. Yogyakarta: Graha Ilmu, 2011.
- [21] D. Puspita, *Perancangan Sistem Informasi Pengelolaan Data Tamu dan Pengunjung di Museum Geologi, Universitas Komputer Indonesia Bandung*. Bandung, 2012.
- [22] L. Hakim, *Membangun Web Berbasis PHP dengan Framework CodeIgniter*. Yogyakarta: Lokomedia, 2010.
- [23] Basuki, *Elektronik Data Prosesing*. Yogyakarta, 2014.
- [24] H. Winarto, *Pemanfaatan XAMPP Sebagai Web, Database, dan FTP Server di Sekolah*. Indonesia: Sekolah Menengah Kejuruan 1 Pacitan, 2009.
- [25] I. Sommerville, *Software Engineering (Rekayasa Perangkat Lunak)*. Jakarta: Erlangga.