BAB 3
ANALISIS DAN PERANCANGAN SISTEM

3.1 Analisis Sistem

Analisis sistem dapat didefinisikan sebagai penguraian dari sistem utama ke dalam sub-sub sistem dengan tujuan untuk mengidentifikasikan kebutuhan-kebutuhan yang diperlukan agar dapat dibangun sebuah aplikasi informasi mengenai bus DAMRI di kota Bandung.

3.1.1 Analisis Masalah

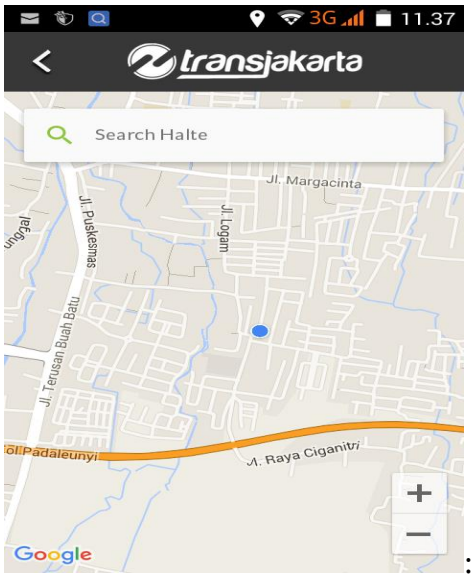
Berdasarkan penelitian yang dilakukan, permasalahan yang terjadi dapat dirumuskan sebagai berikut :

1. Masih banyaknya pengguna transportasi umum bus DAMRI yang belum mengetahui trayek-trayek bus DAMRI yang ada.
2. Belum adanya aplikasi media informasi trayek bus DAMRI yang dapat diakses melalu *smartphone* berbasis android.
3. Kemacetan yang tidak dapat diprediksi membuat pengguna dapat menunggu cukup lama di halte dan tidak mengetahui bus DAMRI telah lewat atau belum.

3.1.2 Analisis Aplikasi Sejenis

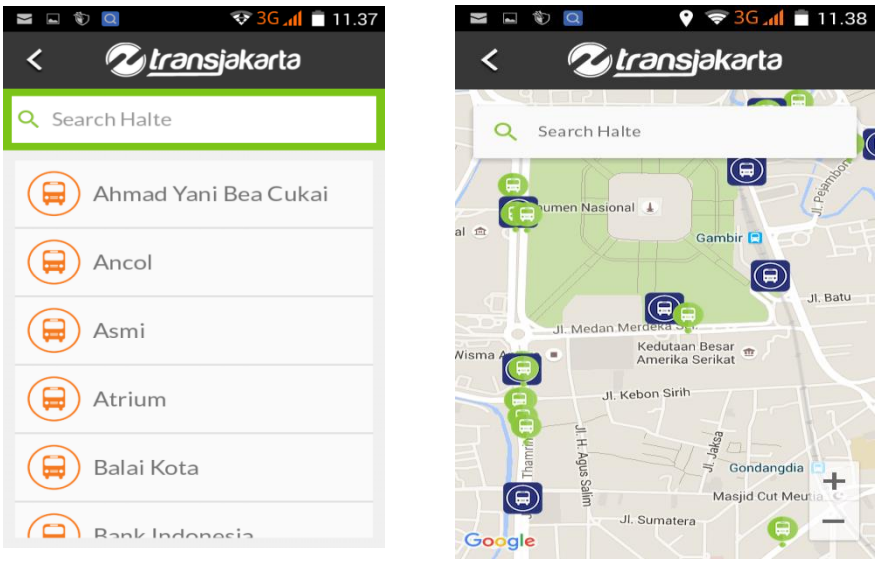
Analisis aplikasi sejenis dilakukan terhadap aplikasi GO-JEK dimana pada aplikasi tersebut tersebut terdapat salah satu menu yang menampilkan informasi lokasi keberadaan bus TransJakarta berdasarkan pencarian halte-halte yang ada di jakarta.

Tampilan awal dari menu Go Busway yang tedapat pada aplikasi GO-JEK tersebut adalah sebagai berikut pada Gambar 3.1



Gambar 3.1 Tampilan Awal Go Busway

Pada pertama kali aplikasi dibuka, aplikasi langsung diarahkan kepada posisi pengguna di map. Selanjutnya pengguna harus memilih salah satu halte yang tertera pada menu diatas map seperti Gambar 3.2

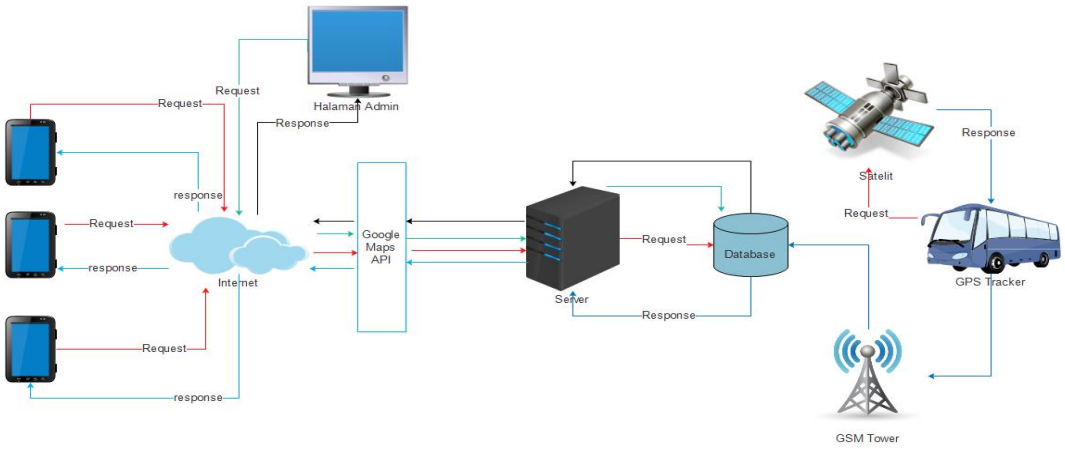


Gambar 3.2 Tampilan Aplikasi Go Busway

Setelah pengguna memilih salah satu halte yang ada, aplikasi akan mengarahkan pada halte yang dipilih dan menampilkan bus-bus yang ada baik yang melewati halte tersebut ataupun tidak.

3.1.3 Analisis Arsitektur Sistem

Analisis arsitektur sistem bertujuan untuk mengidentifikasi arsitektur yang akan dibangun.Berikut adalah arsitektur sistem secara keseluruhan pada Gambar.3.3

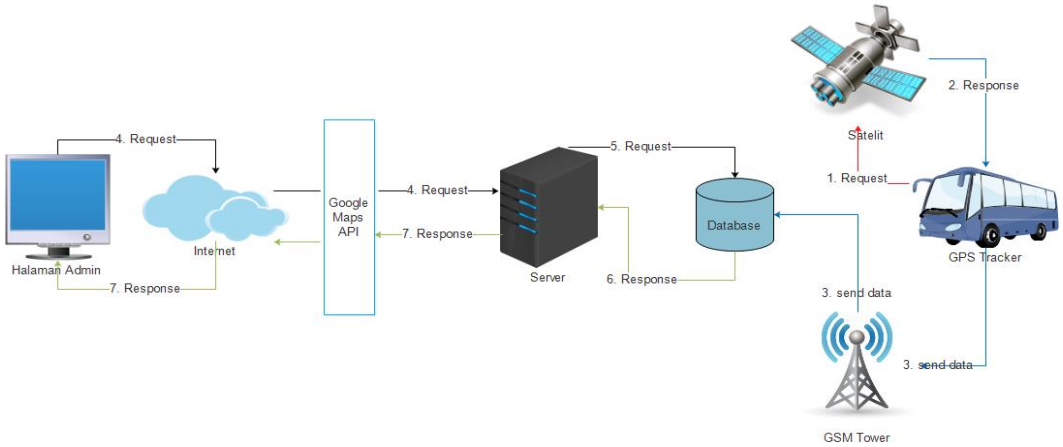


Gambar.3.3 Arsitektur Perangkat lunak keseluruhan

Berdasarkan gambaran arsitektur sistem diatas terbagi menjadi dua subsistem, yaitu web dan mobile.

a. Web

Dalam pembangun perangkat lunak ini, *Platform* web digunakan sebagai halaman untuk administrator bertugas mengolah data trayek dari bus DAMRI. Berikut adalah gambaran arsitektur perangkat lunak pada Gambar 3.4



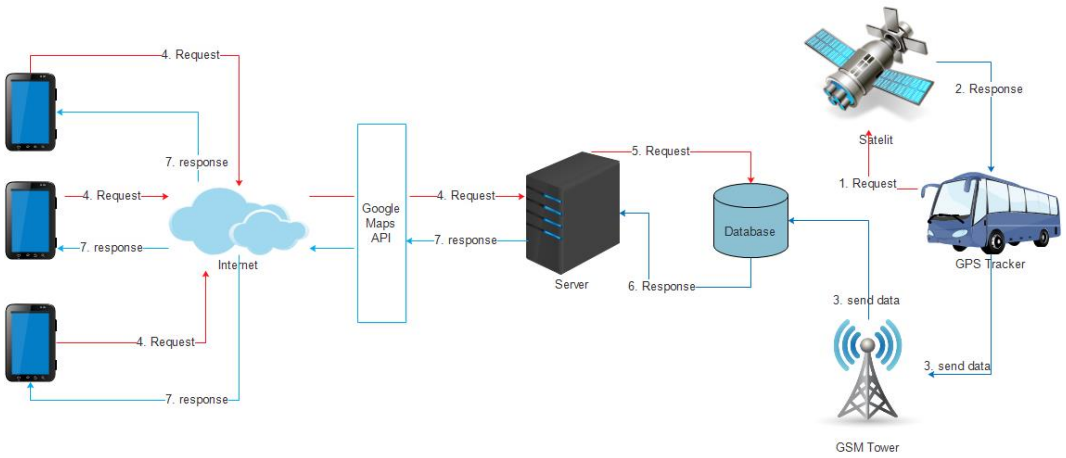
Gambar 3.4 Arsitektur Perangkat lunak pada web

Berikut deskripsi dari gambar 3.4 arsitektur perangkat lunak pada web :

1. *GPS Tracker* melakukan *request* data berupa lokasi keberadaan GPS.
2. Satelit menerima *request* data lalu mengirimkan koordinat lokasi GPS berada.
3. GPS *Tracker* mengirimkan koordinat lokasi GPS berada pada *database* melalui GSM Tower menggunakan internet.
4. Sistem web admin melakukan *request* data pada *server* melalu internet.
5. *Server* menerima *request* data dari admin dan mengambil data sesuai permintaan dari *database*.
6. Jika data ada, *database* akan memberikan respon dengan mengirimkan data yang diminta kembali ke *server*.
7. *Server* mengirimkan data yang diminta kepada admin melalui internet.

b. *Mobile*

Platform mobile digunakan sebagai halaman untuk *user* berinteraksi dengan sistem agar mendapatkan informasi mengenai trayek serta lokasi keberadaan dari bus DAMRI di kota Bandung. Berikut adalah gambaran arsitektur perangkat lunak pada Gambar 3.5



Gambar 3.5 Arsitektur Perangkat lunak pada *mobile*

Berikut deskripsi dari gambar 3.5 arsitektur perangkat lunak pada *mobile* :

1. *GPS Tracker* melakukan *request* data berupa lokasi keberadaan GPS.
2. Satelit menerima *request* data lalu mengirimkan koordinat lokasi GPS berada.

3. GPS *Tracker* mengirimkan koordinat lokasi GPS berada pada *database* melalui GSM Tower menggunakan internet.
4. Pengguna melakukan *request* data menggunakan perangkat *mobile* ke server melalui internet.
5. Server akan mengambil data pada *database* sesuai *request* dari pengguna.
6. Jika *request* pengguna berupa informasi umum bus DAMRI maka server akan langsung mengirimkan data pada *mobile* pengguna.
7. Jika *request* pengguna berupa lokasi keberadaan bus DAMRI dan DAMRI *near me* maka server akan menggunakan Google Maps API untuk menampilkannya dalam bentuk peta digital pada perangkat *mobile* pengguna.

3.1.4 Spesifikasi Kebutuhan Perangkat Lunak

Spesifikasi kebutuhan perangkat lunak terbagi kedalam dua kebutuhan yaitu kebutuhan fungsional dan non fungsional.Kebutuhan fungsional dapat dilihat pada Tabel 3.1 Spesifikasi Kebutuhan Fungsional

Tabel 3.1 Spesifikasi kebutuhan fungsional

Kode SKPL	Spesifikasi Kebutuhan Fungsional
SKPL-F-001	Subsistem mobile menyediakan fasilitas bagi pengguna untuk memilih salah satu trayek yang ada.
SKPL-F-002	Subsistem mobile menyediakan fasilitas bagi pengguna untuk mendapatkan informasi umum trayek-trayek dan bus DAMRI.
SKPL-F-003	Subsistem mobile menyediakan fasilitas bagi pengguna untuk mendapatkan informasi lokasi keberadaan bus DAMRI berada.
SKPL-F-004	Subsistem mobile menyediakan fasilitas bagi pengguna untuk mendapatkan informasi lokasi keberadaan bus DAMRI berada yang ada disekitar pengguna.
SKPL-F-005	Subsistem web menyediakan fasilitas bagi admin untuk login kedalam sistem.
SKPL-F-006	Subsistem web menyediakan fasilitas bagi admin untuk kelola datanya sendiri
SKPL-F-007	Subsistem web menyediakan fasilitas bagi admin untuk

	mengubah data admin
SKPL-F-008	Subsistem web menyediakan fasilitas bagi admin untuk kelola data trayek
SKPL-F-009	Subsistem web menyediakan fasilitas bagi admin untuk menambah data trayek
SKPL-F-010	Subsistem web menyediakan fasilitas bagi admin untuk mengubah data trayek
SKPL-F-011	Subsistem web menyediakan fasilitas bagi admin untuk menghapus data trayek
SKPL-F-012	Subsistem web menyediakan fasilitas bagi admin untuk kelola data bus
SKPL-F-013	Subsistem web menyediakan fasilitas bagi admin untuk menambah data bus
SKPL-F-014	Subsistem web menyediakan fasilitas bagi admin untuk mengubah data bus
SKPL-F-015	Subsistem web menyediakan fasilitas bagi admin untuk menghapus data bus
SKPL-F-016	Subsistem web menyediakan fasilitas bagi admin untuk menambah data GPS
SKPL-F-017	Subsistem web menyediakan fasilitas bagi admin untuk mengubah data GPS
SKPL-F-018	Subsistem web menyediakan fasilitas bagi admin untuk menghapus data GPS

Kebutuhan non-fungsional pada perangkat lunak yang dibangun dapat dilihat pada Tabel 3.2 Spesifikasi kebutuhan non fungsional

Tabel 3.2 Spesifikasi kebutuhan fungsional

Kode SKPL	Spesifikasi Kebutuhan Non Fungsional
SKPL-NF-001	Sistem yang dibangun memiliki 2 subsistem yaitu web dan *mobile*
SKPL-NF-002	Sistem yang dibangun minimal menggunakan android versi 4.2

	Jelly Bean
SKPL-NF-003	Sistem yang dibangun menggunakan GPS *Tracker* TR02
SKPL-NF-004	Sistem yang dibangun memanfaatkan Google Maps API
SKPL-NF-005	Sistem dibangun dengan spesifikasi *hardware* yang memenuhi standar minimum kebutuhan

3.1.5 Analisis Kebutuhan Non-Fungsional

Analisis Non-Fungsional merupakan analisis yang dibutuhkan untuk menentukan spesifikasi kebutuhan sistem. Spesifikasi ini juga meliputi elemen atau komponen-komponen apa saja yang dibutuhkan untuk sistem yang akan dibangun sampai dengan sistem tersebut diimplementasikan. Analisis kebutuhan ini juga menentukan spesifikasi masukan yang diperlukan dalam sistem, keluaran yang akan dihasilkan sistem dan proses yang dibutuhkan untuk mengolah masukan sehingga menghasilkan suatu keluaran yang diinginkan.

Pada analisis kebutuhan sistem non fungsional ini juga dijelaskan analisis mengenai perangkat keras (hardware) dan perangkat lunak (software) sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan.

3.1.5.1 Analisis Kebutuhan Perangkat Lunak

Analisis kebutuhan perangkat lunak (*software*) yang diperlukan untuk menjalankan Sistem dijelaskan pada Tabel 3.3 berikut :

Tabel 3.3 Analisis Perangkat Lunak

Perangkat lunak	Keadaan yang dibutuhkan
Operating System untuk web	*Win* 7 x32*bit*
Operating System untuk *mobile*	Android versi 4.2 Jelly Bean
Web *Server*	*XamppVersion* 3.1.0
Browser	*Mozila Firefox, Opera, Chrome*
Kode *Editor*	Notepad ++, Macromedia Dreamweaver
Integrated development environment (IDE)	Eclipse
Framework Andorid	*Android Software Development Kit*

3.1.5.2 Analisis Kebutuhan Perangkat Keras

Analisis kebutuhan perangkat keras (*Hardware*) yang diperlukan untuk menjalankan sistem dijelaskan pada Tabel 3.4 - Tabel 3.6 berikut :

Tabel 3.4 Analisis Kebutuhan Perangkat Keras *Mobile*

Smartphone		
No	**Spesifikasi**	
1	Sistem Operasi	Versi 4.2 (JB)
2	Display	Menyesuaikan
3	RAM	1 GB
4	Jaringan	Internet

Tabel 3.5 Analisis Kebutuhan Perangkat Keras Komputer

PC		
No	**Spesifikasi**	
1	Monitor	Monitor 14”
2	RAM	2 GB
3	Display	Menyesuaikan
4	Jaringan	Internet

Tabel 3.6 Analisis Kebutuhan Perangkat Keras GPS *Tracker*

GPS		
No	**Spesifikasi**	
1	Provider	GSM
2	Jaringan	3G/GPRS

3.1.5.3 Analisis Kebutuhan Pengguna

Kebutuhan pengguna dapat dilihat pada Tabel 3.7 - Tabel 3.9 berikut.

Tabel 3.7 Analisis Kebutuhan pengguna

Jenis SDM	Karakteristik
Admin	- Dapat mengoperasikan komputer dan *smartphone*

	- Memiliki pengalaman menggunakan internet - Dapat menggunakan browser seperti Mozila Firefox, Google Chrome dan Internet Explorer
User	- Dapat mengoperasikan *smartphone* - Memiliki pengalaman menggunakan internet

Spesifikasi pengguna yang terlibat dalam sistem yang akan dibangun dapat dideskripsikan sebagai berikut:

1. Admin

Tabel 3.8 Analisis Kebutuhan Pengguna Admin

Pengguna	Admin
Tanggung Jawab	- Melakukan Login - Melakukan pengubahan data admin - Melakukan pengolahan data bus - Melakukan pengolahan data trayek - Melakukan pengolahan data GPS
Hak Akses	Login, mengolah data bus, data trayek
Tingkat Keterampilan	- Dapat mengoperasikan komputer dan *smartphone* - Memiliki pengalaman menggunakan internet - Dapat menggunakan browser seperti Mozila Firefox, Google Chrome dan Internet Explorer
Pengalaman	- Pernah mengakses internet - Pernah melakukan pengolahan data berbasis web

2. User (Masyarakat Pengguna Bus DAMRI)

Tabel 3.9 Analisis Kebutuhan Pengguna User

Pengguna	*User*
Tanggung Jawab	- Melihat informasi umum mengenai trayek dan bus DAMRI

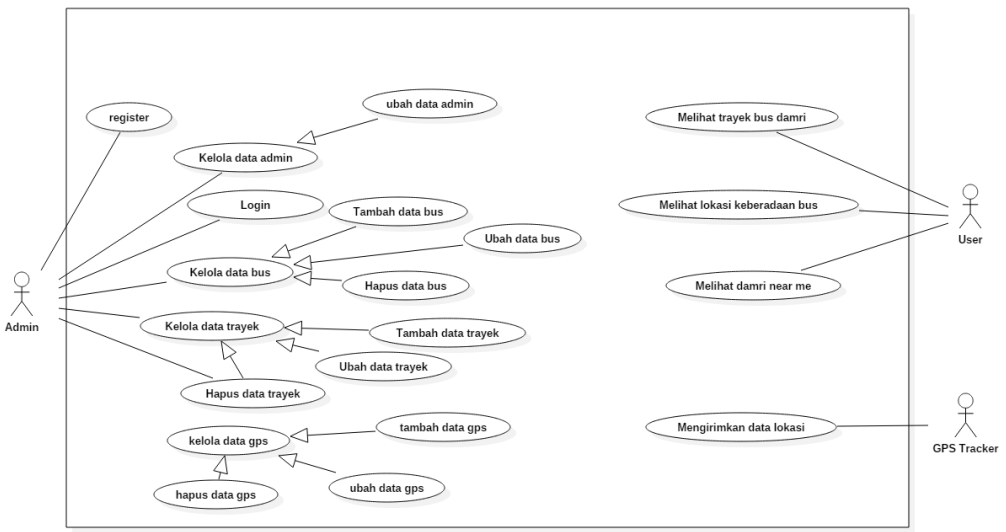
	- Melihat lokasi keberadaan bus DAMRI - Melihat lokasi keberadaan bus DAMRI yang berada disekitarnya
Hak Akses	Melihat informasi umum trayek dan bus DAMRI, melihat lokasi keberadaan bus DAMRI, melihat lokasi keberadaan bus DAMRI yang ada disekitar *user*
Tingkat Keterampilan	- Dapat mengoperasikan *smartphone* - Memiliki pengalaman menggunakan internet
Pengalaman	- Pernah mengakses internet - Pernah mengoperasikan *smartphone*

3.1.6 Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional mengambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan.

3.1.6.1 *Use Case*

Use case diagram menggambarkan proses pada sistem *risk management*. *Use case diagram* aplikasi ini digambarkan pada Gambar 3.6.



Gambar 3.6 *Use Case Diagram*

3.1.6.2 *Use Case Scenario*

Berikut penjelasan *use case scenario* pada Tabel 3.10 - Tabel 3.23

1. *Use Case Scenario login*

Tabel 3.10 *Use Case Scenario Login*

Identifikasi Masalah	
Nama *Use Case*	*Login*
Tujuan	Masuk ke dalam sistem sebagai *admin*
Deskripsi	Proses login ini sebagai autentifikasi kewenangan sebagai *admin* dalam sistem.
Aktor	*admin*
Skenario Utama	
Kondisi Awal	*Form Login* ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. *Input username* dan *password* pada *form login*	
	2. Pengecekan valid atau tidaknya *username* dan *password* 3. Tampilkan pesan *login* berhasil 4. Tampilkan halaman utama
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. *Input username* dan *password* pada *form login*	
	2. Pengecekan *valid* atau tidaknya *username* dan *password*

	3. Tampilkan pesan gagal *login* jika *username* dan *password* tidak cocok
4. *Input username* dan *password* pada *form login*	
	5. Pengecekan *login* telah valid 6. Tampilkan pesan *login* berhasil 7. Tampilkan halaman utama
Kondisi Akhir	*admin* dapat melakukan kegiatan sistem sesuai kewenangan

2. *Use Case Scenario* ubah data admin

Tabel 3.11 *Use Case Scenario* ubah data admin

Identifikasi Masalah	
Nama *Use Case*	*Edit* data admin
Tujuan	Mengubah *data* pada admin kedalam *database*
Deskripsi	Proses ubah *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses data admin
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data	

admin	
	2. Tampilkan halaman kelola data admin dan data admin yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan diubah pada tabel *data* admin	
	4. Tampilkan *data* identifikasi tujuan yang dipilih pada *form*
5. Ubah *data*	
	6. Validasi *data* baru 7. *Data* yang diubah tersimpan dalam *database* 8. Tampilkan pesan *data* dalam databasetelah diubah dan tampilkan *data* yang telah diubah
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data admin	
	2. Tampilkan halaman kelola data admin dan data admin yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan diubah pada tabel *data* admin	

	4. Tampilkan urutan *data* identifikasi tujuan yang dipilih pada *form*
5. Ubah *data*	
	6. Validasi *data* baru 7. Munculkan pesan data belum lengkap 8. Jika *inputan* sudah *valid, data* yang diubah tersimpan dalam *database* 9. Tampilkan pesan *data* dalam databasetelah diubah dan tampilkan *data* yang telah diubah
Kondisi Akhir	Data admin berhasil di *edit*

3. *Use Case Scenario* tambah data trayek

Tabel 3.12 *Use Case Scenario* tambah data trayek

Identifikasi Masalah	
Nama *Use Case*	Tambah data trayek
Tujuan	Menambah *data* trayek kedalam *database*
Deskripsi	Proses penambahan *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses tambah trayek
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan

Aksi Aktor	**Reaksi Sistem**
1. Pilih menu Kelola data trayek	
	2. Tampilkan halaman kelola data trayek dan data trayek yang tersimpan dalam bentuk tabel
3. Pilih menu tambah	
	4. Tampilkan *form* tambah data trayek
5. *Input data* pada *form*	
	6. Validasi Inputan pada *form* 7. Tampilkan pesan *data* baru tersimpan dalam *database* dan tampilkan *data* baru
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu Kelola data trayek	
	2. Tampilkan halaman kelola data rayek dan data trayek yang tersimpan dalam bentuk tabel
3. Pilih menu tambah	
	4. Tampilkan *form* tambah data trayek
5. *Input data* pada form	

	6. Validasi Inputan pada *form* 7. Munculkan pesan *data* belum lengkap 8. Jika inputan sudah *valid*, *data* disimpan dalam *database* 9. Tampilkan pesan *data* baru tersimpan dalam *database* dan tampilkan *data* baru
Kondisi Akhir	Penambahan *data* trayek berhasil ditambahkan kedalam *database*

4. *Use Case Scenario* ubah data trayek

Tabel 3.13 *Use Case Scenario* Ubah data trayek

Identifikasi Masalah	
Nama *Use Case*	*Edit* data trayek
Tujuan	Mengubah *data* trayek kedalam *database*
Deskripsi	Proses ubah *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses data trayek
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data trayek	
	2. Tampilkan halaman kelola data

	trayek dan data trayek yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan diubah pada tabel *data* trayek	
	4. Tampilkan *data* trayek yang dipilih pada *form*
5. Ubah *data*	
	6. Validasi *data* baru 7. *Data* yang diubah tersimpan dalam *database* 8. Tampilkan pesan *data* dalam database telah diubah dan tampilkan *data* yang telah diubah
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data trayek	
	2. Tampilkan halaman kelola data trayek dan data trayek yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan diubah pada tabel *data*	

trayek	
	4. Tampilkan urutan *data* trayek yang dipilih pada *form*
5. Ubah *data*	
	6. Validasi *data* baru 7. Munculkan pesan data belum lengkap 8. Jika *inputan* sudah *valid*, *data* yang diubah tersimpan dalam *database* 9. Tampilkan pesan *data* dalam database telah diubah dan tampilkan *data* yang telah diubah
Kondisi Akhir	Data trayek berhasil di *edit*

5. *Use Case Scenario* hapus data trayek

Tabel 3.14 *Use Case Scenario* hapus data trayek

Identifikasi Masalah	
Nama *Use Case*	Hapus data trayek
Tujuan	Menghapus *data* trayek dalam *database*
Deskripsi	Proses penghapusan *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses hapus data trayek
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan

Aksi Aktor	Reaksi Sistem
1. Pilih menu kelola data trayek	
	2. Tampilkan halaman kelola data trayek dan data trayek yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan dihapus pada tabel *data* trayek	
	4. Tampilkan pilih pesan data akan dihapus “ya” atau “tidak” 5. Jika pilih “ya”, data terhapus dari *database*. Jika pilih “tidak”, hapus data dibatalkan
Kondisi Akhir	Data trayek berhasil di hapus.

6. *Use Case Scenario* tambah data bus

Tabel 3.15 *Use Case Scenario* tambah data bus

Identifikasi Masalah	
Nama *Use Case*	Tambah data bus
Tujuan	Menambah *data* bus kedalam *database*
Deskripsi	Proses penambahan *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses tambah bus

Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utamaditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu Kelola data bus	
	2. Tampilkan halaman kelola data bus dan data bus yang tersimpan dalam bentuk tabel
3. Pilih menu tambah	
	4. Tampilkan *form* tambah data bus
5. *Input data* pada *form*	
	6. Validasi Inputan pada *form* 7. Tampilkan pesan *data* baru tersimpan dalam *database* dan tampilkan *data* baru
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu Kelola data bus	
	2. Tampilkan halaman kelola data bus dan data bus yang tersimpan dalam bentuk tabel
3. Pilih menu tambah	

	4. Tampilkan *form* tambah data bus
5. *Input data* pada form	
	6. Validasi Inputan pada *form* 7. Munculkan pesan *data* belum lengkap 8. Jika inputan sudah *valid, data* disimpan dalam *database* 9. Tampilkan pesan *data* baru tersimpan dalam *database* dan tampilkan *data* baru
Kondisi Akhir	Penambahan *data* bus berhasil ditambahkan kedalam *database*

7. *Use Case Scenario* ubah data bus

Tabel 3.16 *Use Case Scenario* Ubah data bus

Identifikasi Masalah	
Nama *Use Case*	*Edit* data bus
Tujuan	Mengubah *data* bus kedalam *database*
Deskripsi	Proses ubah *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses data bus
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data bus	
	2. Tampilkan halaman kelola data bus

	dan data bus yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan diubah pada tabel *data* bus	
	4. Tampilkan *data* bus yang dipilih pada *form*
5. Ubah *data*	
	6. Validasi *data* baru 7. *Data* yang diubah tersimpan dalam *database* 8. Tampilkan pesan *data* dalam databasetelah diubah dan tampilkan *data* yang telah diubah
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data bus	
	2. Tampilkan halaman kelola data bus dan data bus yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan diubah pada tabel *data* bus	
	4. Tampilkan urutan *data* bus yang dipilih pada *form*

5. Ubah *data*	
	6. Validasi *data* baru 7. Munculkan pesan data belum lengkap 8. Jika *inputan* sudah *valid*, *data* yang diubah tersimpan dalam *database* 9. Tampilkan pesan *data* dalam database telah diubah dan tampilkan *data* yang telah diubah
Kondisi Akhir	Data bus berhasil di *edit*

8. *Use Case Scenario* hapus data bus

Tabel 3.17 *Use Case Scenario* hapus data bus

Identifikasi Masalah	
Nama *Use Case*	Hapus data bus
Tujuan	Menghapus *data* bus dalam *database*
Deskripsi	Proses penghapusan *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses hapus data bus
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**

1. Pilih menu kelola data bus	
	2. Tampilkan halaman kelola data bus dan data bus yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan dihapus pada tabel *data* bus	
	4. Tampilkan pilih pesan data akan dihapus “ya” atau “tidak” 5. Jika pilih “ya”, data terhapus dari *database*. Jika pilih “tidak”, hapus data dibatalkan
Kondisi Akhir	Data bus berhasil di hapus.

9. *Use Case Scenario* tambah data gps

Tabel 3.18 *Use Case Scenario* tambah data GPS

Identifikasi Masalah	
Nama *Use Case*	Tambah data gps
Tujuan	Menambah *data* gps kedalam *database*
Deskripsi	Proses penambahan *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses tambah gps
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan

Aksi Aktor	**Reaksi Sistem**
1. Pilih menu Kelola data gps	
	2. Tampilkan halaman kelola data gps dan data gps yang tersimpan dalam bentuk tabel
3. Pilih menu tambah	
	4. Tampilkan *form* tambah data gps
5. *Input data* pada *form*	
	6. Validasi Inputan pada *form* 7. Tampilkan pesan *data* baru tersimpan dalam *database* dan tampilkan *data* baru
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu Kelola data gps	
	2. Tampilkan halaman kelola data gps dan data gps yang tersimpan dalam bentuk tabel
3. Pilih menu tambah	
	4. Tampilkan *form* tambah data gps
5. *Input data* pada form	
	6. Validasi Inputan pada *form* 7. Munculkan pesan *data* belum

	lengkap 8. Jika inputan sudah *valid*, *data* disimpan dalam *database* 9. Tampilkan pesan *data* baru tersimpan dalam *database* dan tampilkan *data* baru
Kondisi Akhir	Penambahan *data* gps berhasil ditambahkan kedalam *database*

10. *Use Case Scenario* ubah data gps

Tabel 3.19 *Use Case Scenario* ubah data GPS

Identifikasi Masalah	
Nama *Use Case*	*Edit* data gps
Tujuan	Mengubah *data* gps kedalam *database*
Deskripsi	Proses ubah *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses data gps
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data gps	
	2. Tampilkan halaman kelola data gps dan data gps yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan	

diubah pada tabel *data* gps	
	4. Tampilkan *data* bus yang dipilih pada *form*
5. Ubah *data*	
	6. Validasi *data* baru 7. *Data* yang diubah tersimpan dalam *database* 8. Tampilkan pesan *data* dalam databasetelah diubah dan tampilkan *data* yang telah diubah
Skenario Alternatif (Jika Gagal)	
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data gps	
	2. Tampilkan halaman kelola data gps dan data bus yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan diubah pada tabel *data* gps	
	4. Tampilkan urutan *data* gps yang dipilih pada *form*
5. Ubah *data*	
	6. Validasi *data* baru 7. Munculkan pesan data belum lengkap

	8. Jika *inputan* sudah *valid*, *data* yang diubah tersimpan dalam *database* 9. Tampilkan pesan *data* dalam database telah diubah dan tampilkan *data* yang telah diubah
Kondisi Akhir	Data gps berhasil di *edit*

11. *Use Case Scenario* hapus data gps

Tabel 3.20 *Use Case Scenario* hapus data GPS

Identifikasi Masalah	
Nama *Use Case*	Hapus data gps
Tujuan	Menghapus *data* gps dalam *database*
Deskripsi	Proses penghapusan *data* ini salah satu pekerjaan dari *admin* yang mempunyai wewenang dalam melakukan proses hapus data bus
Aktor	*admin*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu kelola data gps	
	2. Tampilkan halaman kelola data gps dan data gps yang tersimpan dalam bentuk tabel
3. Pilih kolom yang akan dihapus pada tabel *data*	

gps	
	4. Tampilkan pilih pesan data akan dihapus “ya” atau “tidak” 5. Jika pilih “ya”, data terhapus dari *database*. Jika pilih “tidak”, hapus data dibatalkan
Kondisi Akhir	Data gps berhasil di hapus.

12. *Use Case Scenario* melihat informasi umum trayek & bus DAMRI

Tabel 3.21 *Use Case Scenario* informasi umum trayek & bus DAMRI

Identifikasi Masalah	
Nama *Use Case*	Melihat trayek bus DAMRI
Tujuan	*User* melihat trayek-trayek yang ada pada bus DAMRI di kota Bandung
Deskripsi	User dapat melihat informasi dari setiap trayek-trayek bus DAMRI di kota Bandung yang ada.
Aktor	*user*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu melihat trayek bus DAMRI	
	2. Tampilkan halaman pilihan dari trayek-trayek yang ada
3. Pilih salah satu trayek	

untuk menampilkan informasi umum trayek dan bus.	
	4. Tampilkan informasi umum trayek dan rute dari bus DAMRI yang dipilih.
Kondisi Akhir	Informasi umum dari trayek dan bus DAMRI dapat dilihat oleh *user*

13. *Use Case Scenario* melihat lokasi keberadaan bus DAMRI

Tabel 3.22 *Use Case Scenario* melihat lokasi keberadaan bus DAMRI

Identifikasi Masalah	
Nama *Use Case*	Melihat lokasi DAMRI
Tujuan	*User* melihat lokasi keberadaan bus DAMRI
Deskripsi	User memilih trayek untuk melihat lokasi keberadaan bus DAMRI
Aktor	*user*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu melihat lokasi DAMRI	
	2. Tampilkan halaman pilihan dari trayek-trayek yang ada
3. Pilih salah satu trayek untuk menampilkan lokasi keberadaan bus DAMRI	

	4. Tampilkan lokasi keberadaan bus DAMRI dalam bentuk map.
Kondisi Akhir	Lokasi keberadaan bus DAMRI dapat dilihat oleh *user*

14. *Use Case Scenario* DAMRI *Near Me*

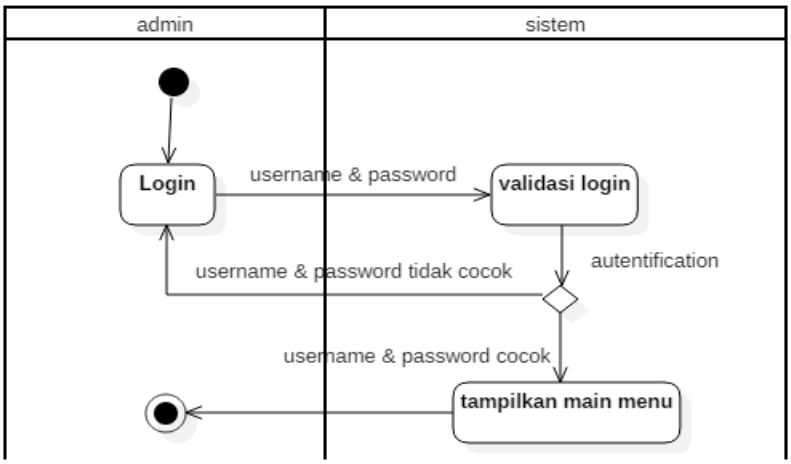
Tabel 3.23 *Use Case Scenario* DAMRI *Near Me*

Identifikasi Masalah	
Nama *Use Case*	Melihat DAMRI *near me*
Tujuan	*User* melihat bus terdekat yang ada sekitarnya
Deskripsi	*User* dapat mengetahui bus DAMRI mana saja yang ada disekitarnya.
Aktor	*user*
Skenario Utama	
Kondisi Awal	Menu utama ditampilkan
Aksi Aktor	**Reaksi Sistem**
1. Pilih menu melihat DAMRI *near me*	
	2. Tampilkan bus-bus yang terdekat yang ada disekitaran *user*
Kondisi Akhir	*User* dapat mengetahui bus-bus DAMRI terdekatnya.

3.1.6.3 *Activity Diagram*

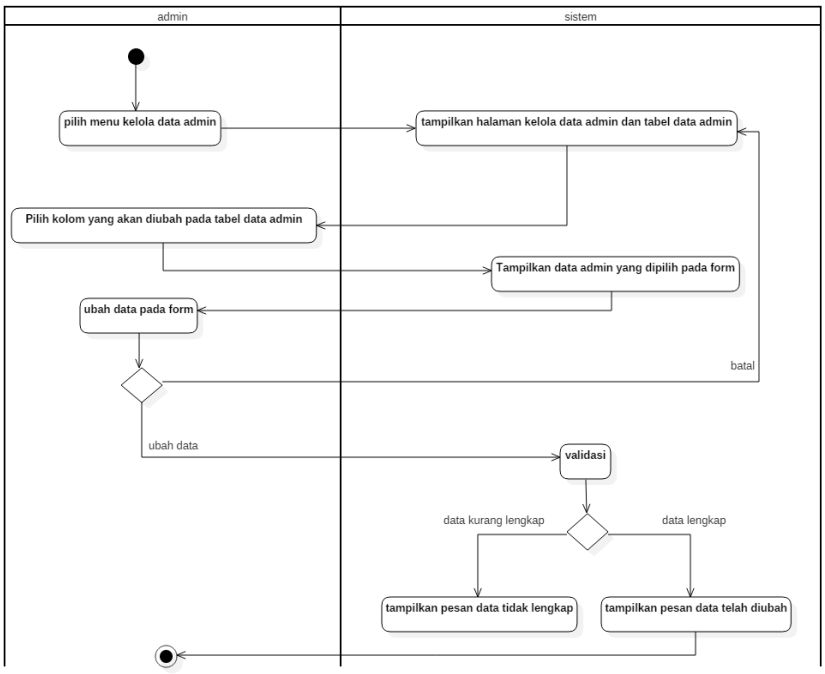
Berikut *Activity Diagram* dalam membangun aplikasi ini pada Gambar 3.10 - Gambar 3.21

1. *Activity Diagram login*



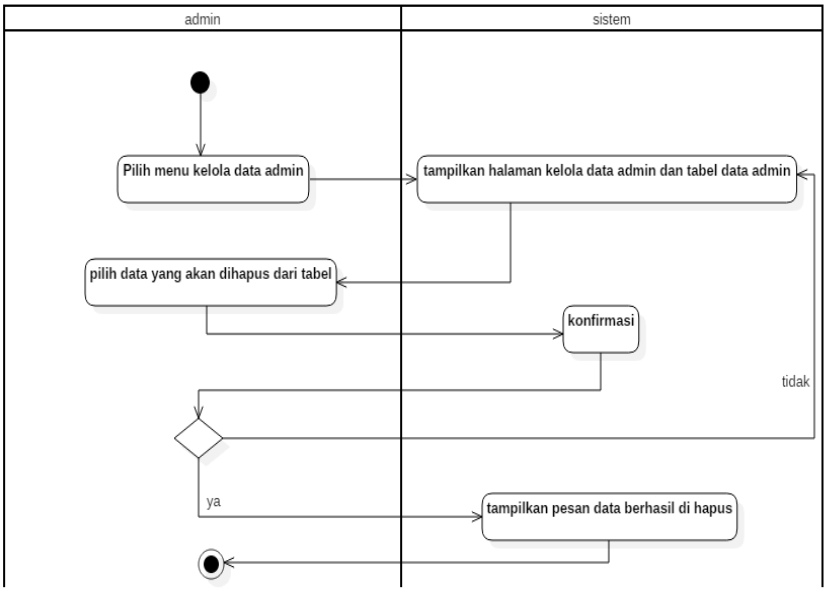
Gambar 3.7 *Activity Diagram login*

2. *Activity Diagram* ubah data admin



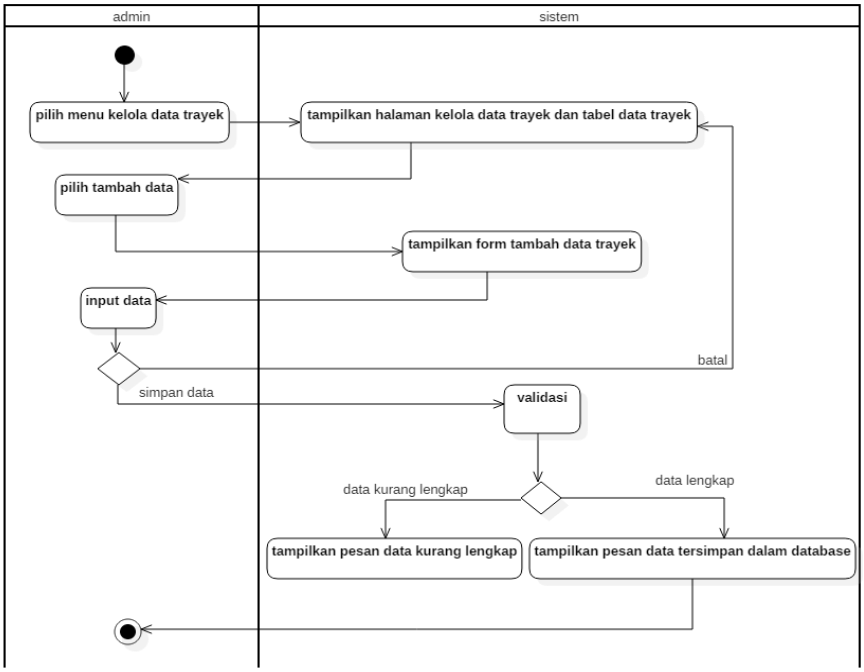
Gambar 3.8 *Activity Diagram* Ubah Data Admin

3. *Activity Diagram* hapus data admin



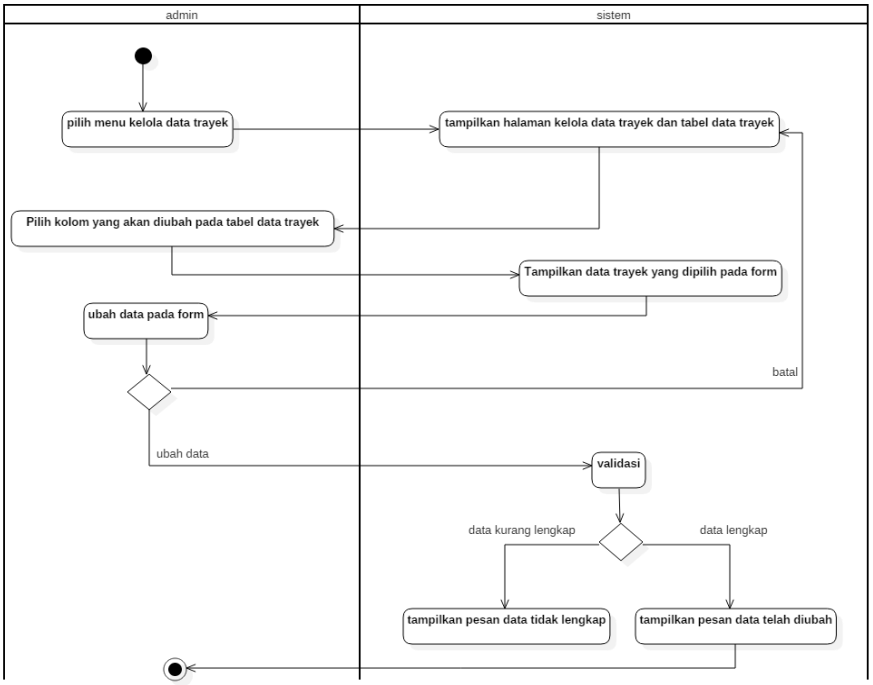
Gambar 3.9 *Activity Diagram* Hapus Data Admin

4. *Activity Diagram* tambah data trayek



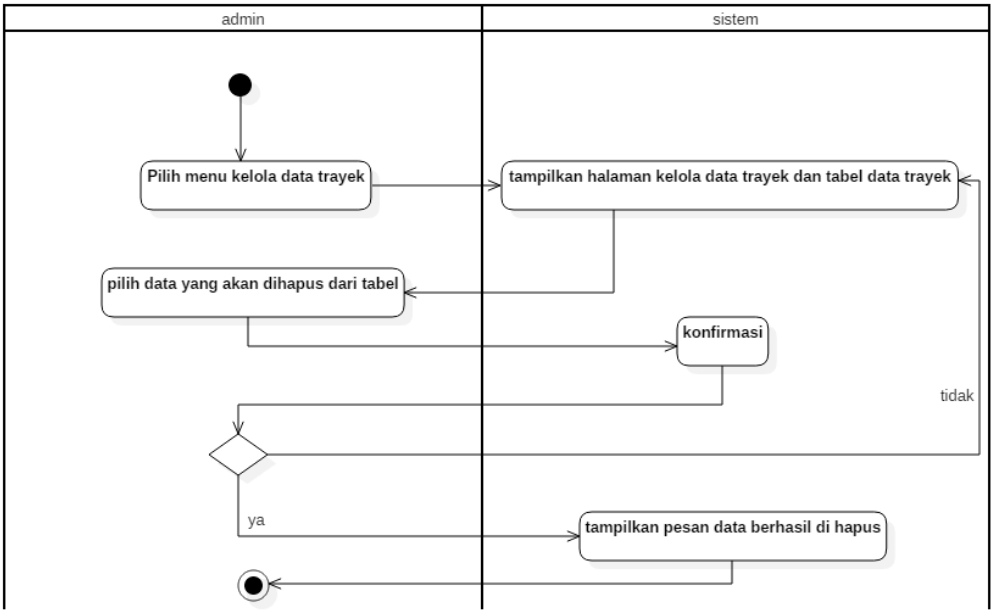
Gambar 3.10 *Activity Diagram* Tambah Data Trayek

5. *Activity Diagram* ubah data trayek



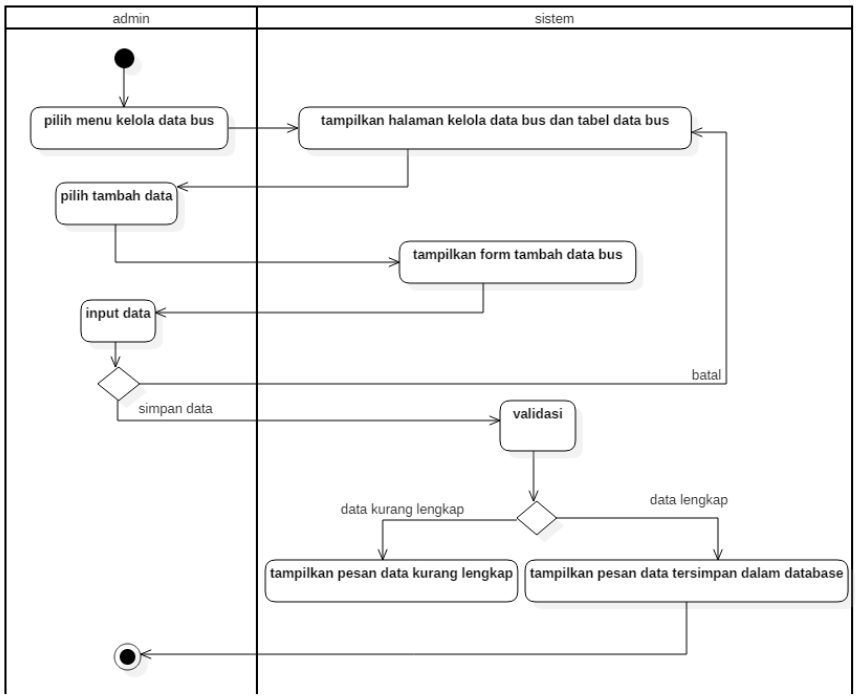
Gambar 3.11 *Activity Diagram* Ubah Data Trayek

6. *Activity Diagram* hapus data trayek



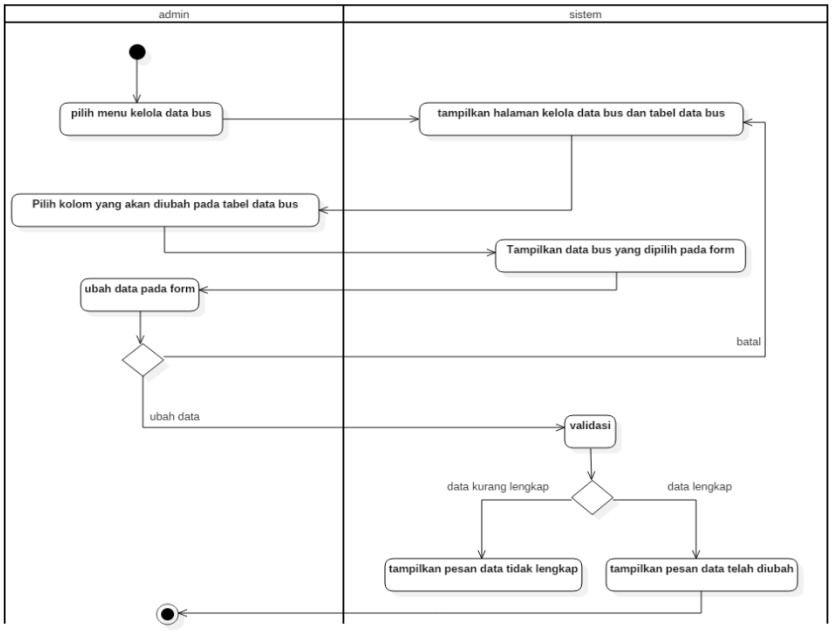
Gambar 3.12 *Activity Diagram* Hapus Data Trayek

7. *Activity Diagram* tambah data bus



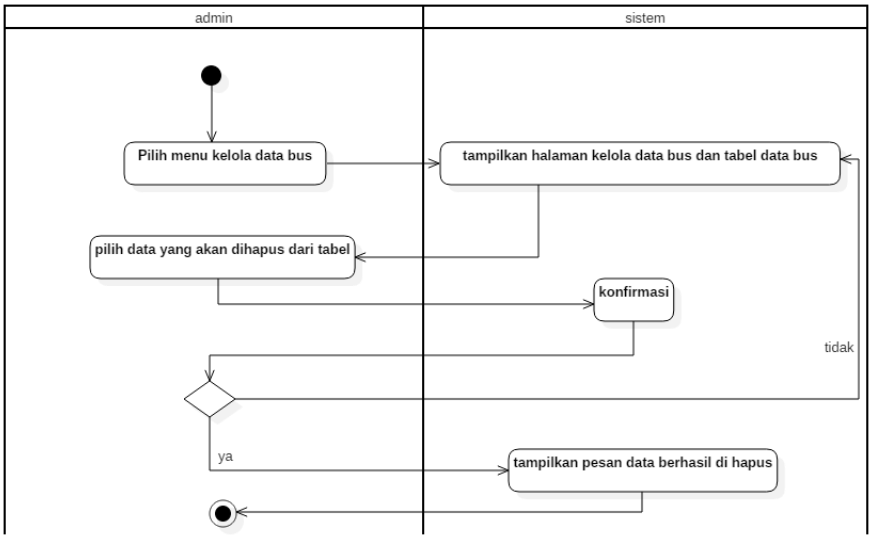
Gambar 3.13 *Activity Diagram* tambah data bus

8. *Activity Diagram* ubah data bus



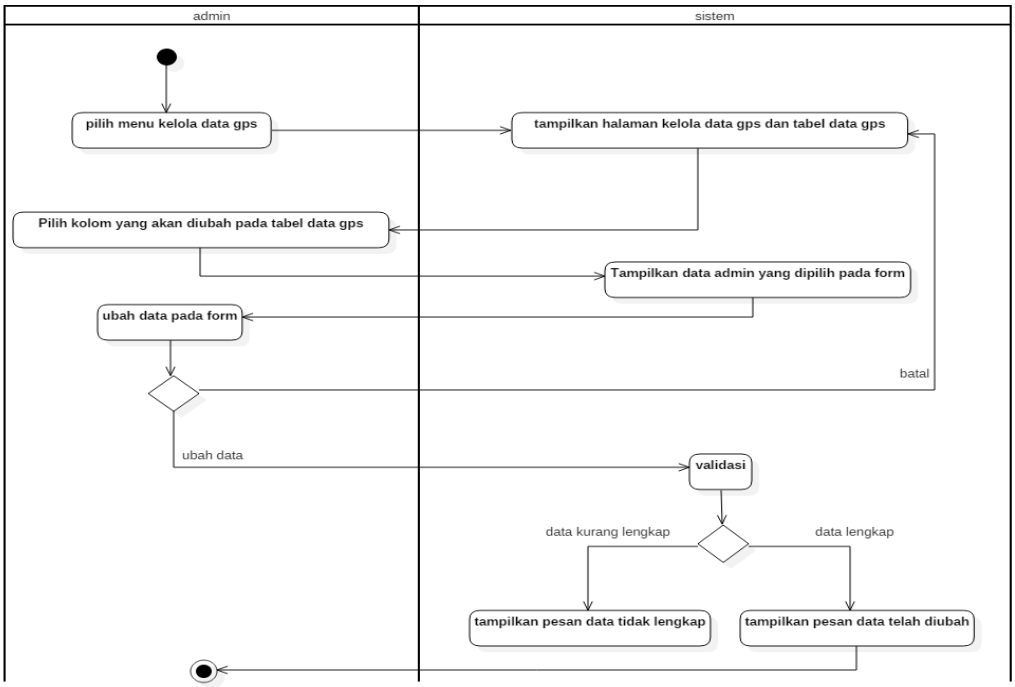
Gambar 3.14 *Activity Diagram* Ubah Data Bus

9. *Activity Diagram* hapus data bus



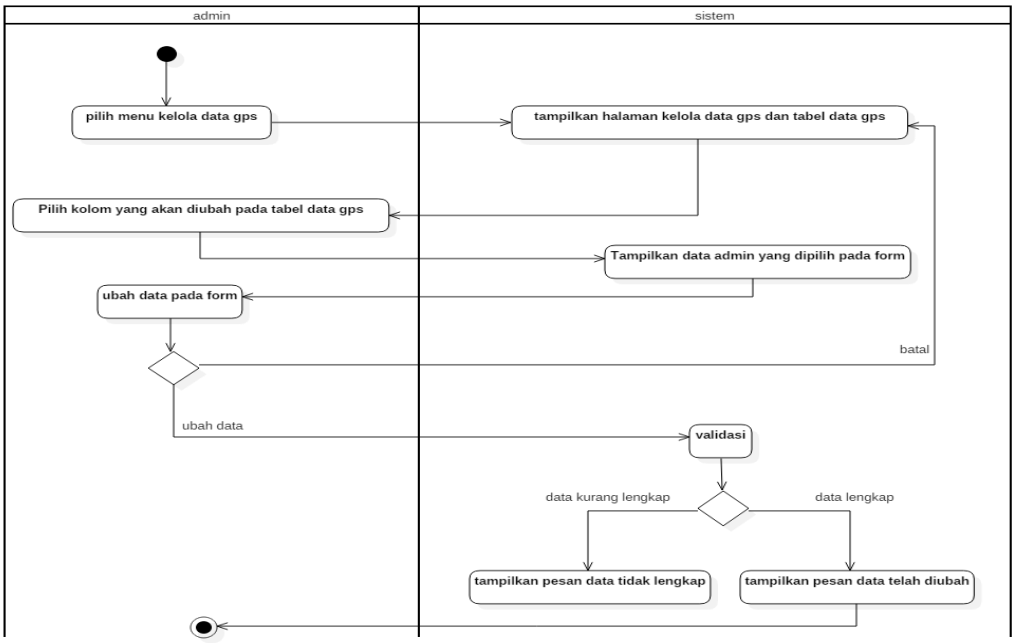
Gambar 3.15 *Activity Diagram* Hapus Data Bus

10. *Activity Diagram* tambah data gps



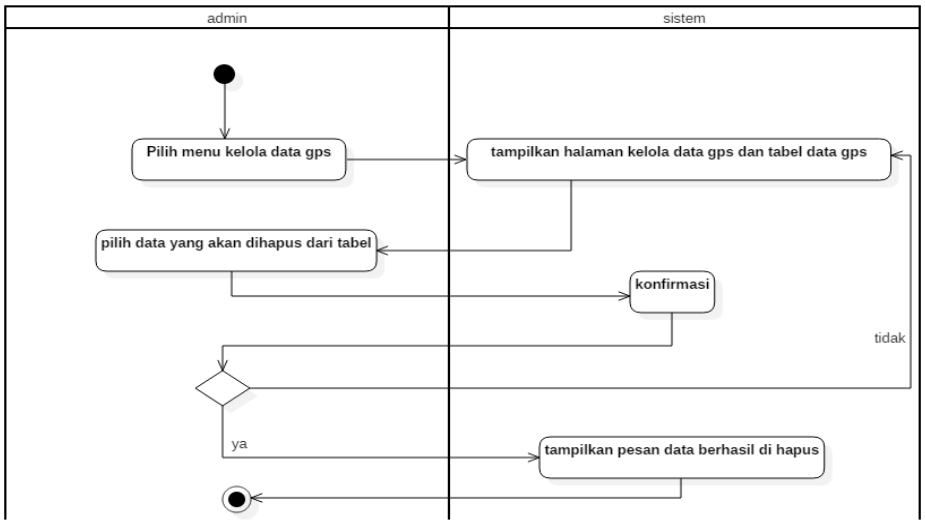
Gambar 3.16 *Activity Diagram* Tambah Data GPS

11. *Activity Diagram* ubah data gps



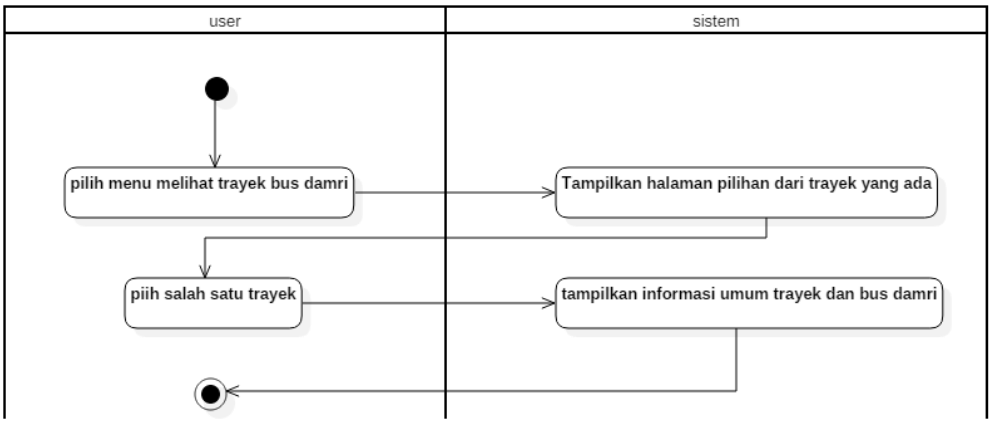
Gambar 3.17 *Activity Diagram* Ubah Data GPS

12. *Activity Diagram* hapus data gps



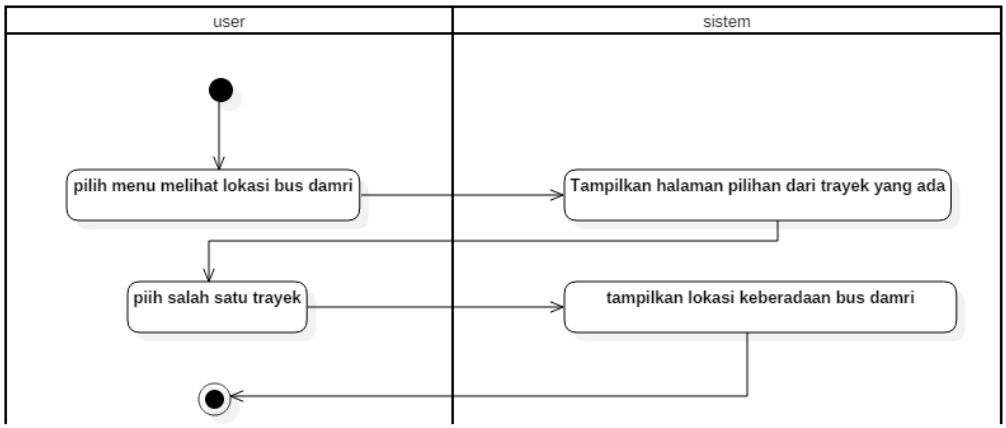
Gambar 3.18 *Activity Diagram* Hapus Data GPS

13. *Activity Diagram* melihat informasi umum trayek & bus DAMRI



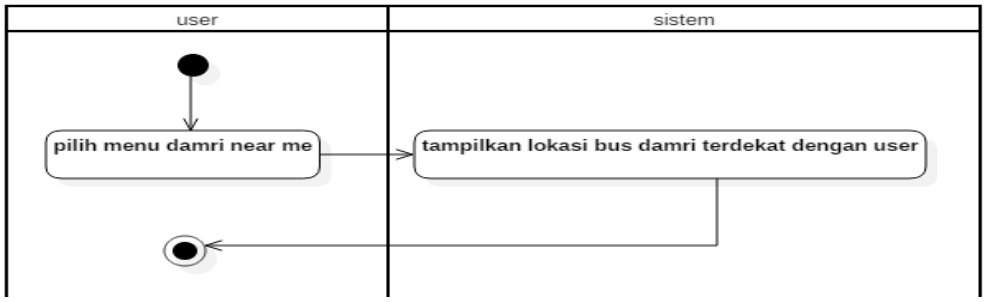
Gambar 3.19 *Activity Diagram* informasi umum trayek & bus DAMRI

14. *Activity Diagram* melihat lokasi keberadaan bus DAMRI



Gambar 3.20 *Activity Diagram* Melihat Lokasi Keberadaan Bus DAMRI

15. *Activity Diagram* DAMRI *Near Me*



Gambar 3.21 *Activity Diagram* DAMRI *Near Me*

3.1.6.4 *Class Diagram*

Class diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (atribut atau *property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut. *Class diagram* akan dijelaskan pada Tabel 3.24 - Tabel 3.25 dan Gambar 3.22 - Gambar 3.23 berikut

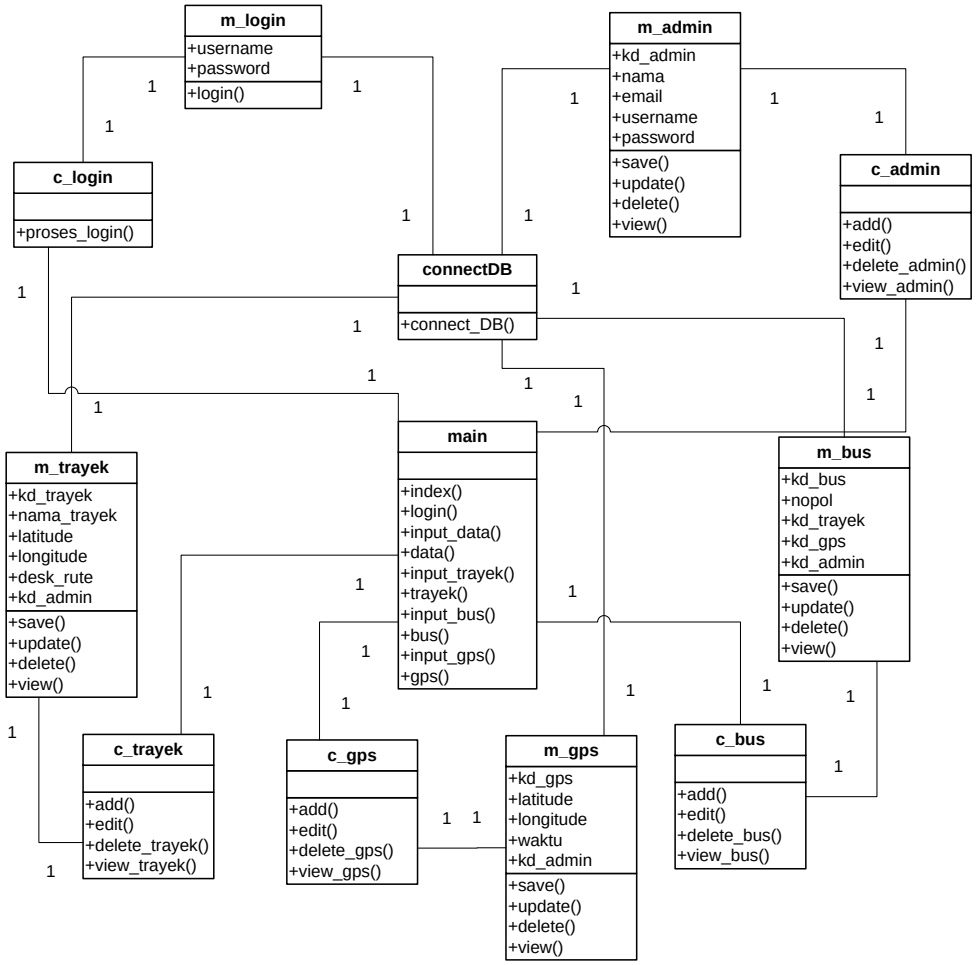
Tabel 3.24 Jenis *class* pada *class diagram* untuk admin

Jenis	Nama Class
Model	m_login
	m_admin
	m_trayek
	m_bus
	m_gps
View	main
control	c_login
	c_admin
	c_trayek
	c_bus
	c_gps

Tabel 3.25 Jenis *Class* Pada *class diagram* untuk *user*

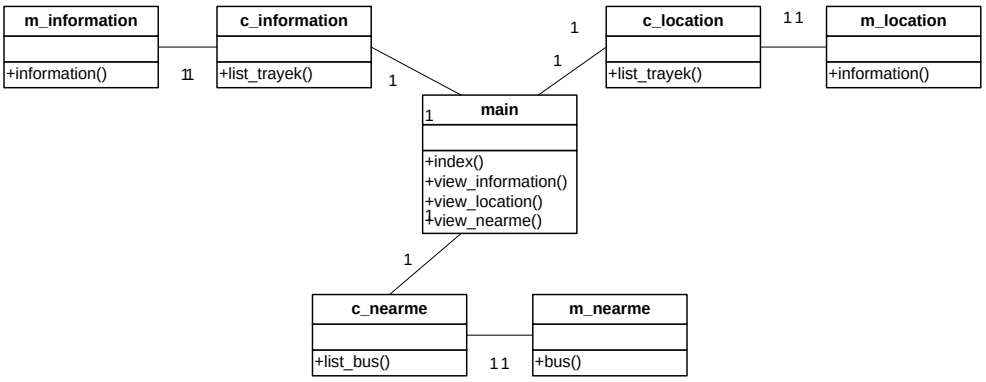
Jenis	Nama Class
Model	m_information
	m_location
	M_nearme
View	App_main
control	c_information
	c_location
	c_nearme

1. *Class Diagram* untuk dan Admin



Gambar 3.22 *class diagram* untuk admin

2. *Class Diagram* Untuk *User*

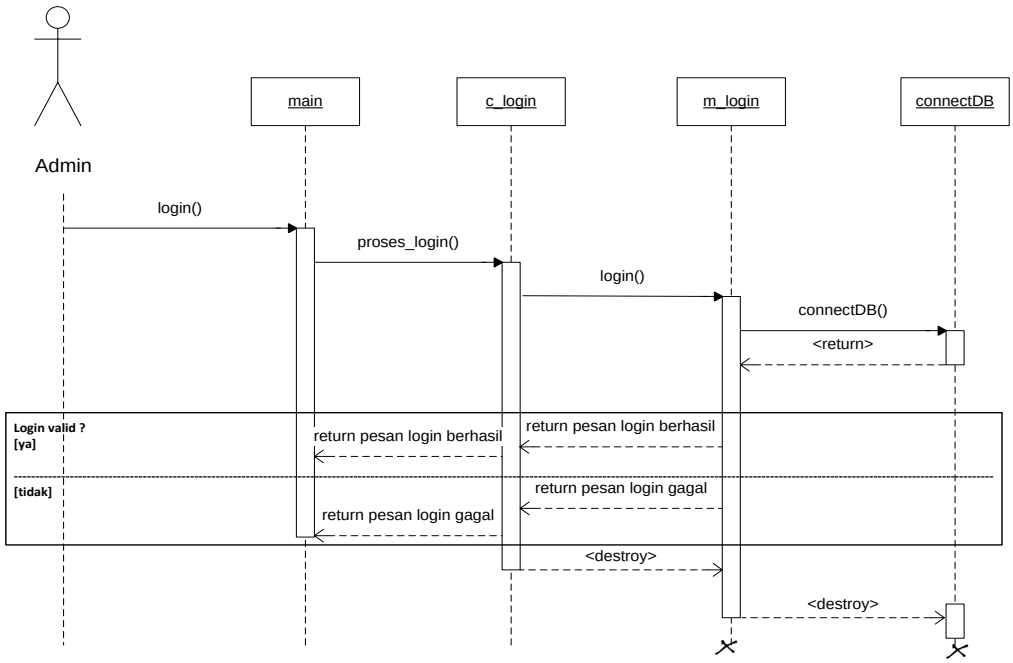


Gambar 3.23 *class diagram* untuk *user*

3.1.6.5 *Sequence Diagram*

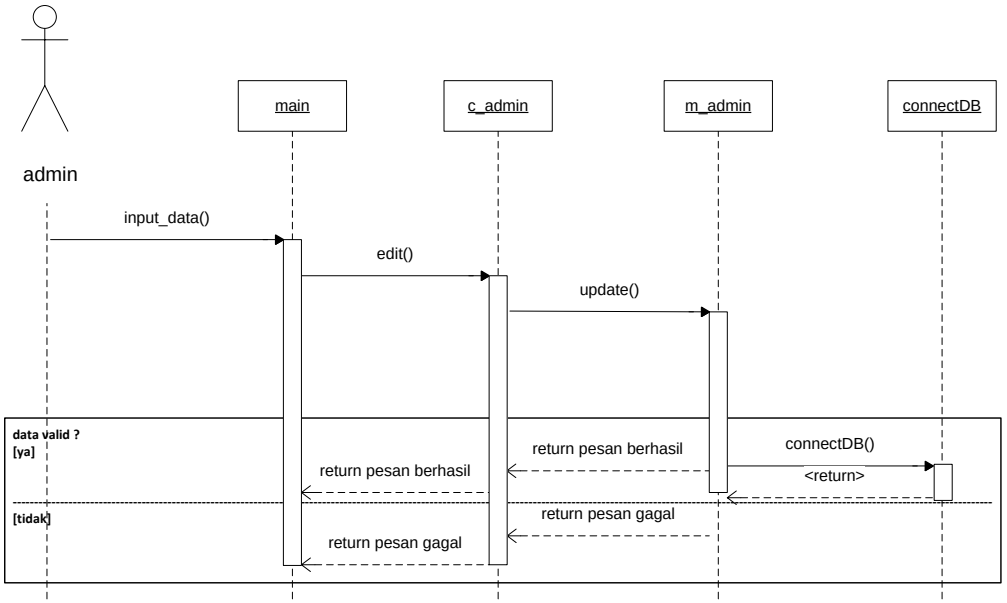
Berikut adalah *Sequence diagram* dari analisis pembangunan aplikasi ini. Dijelaskan pada Gambar 3.24 - Gambar 3.37

1. *Sequence diagram login*



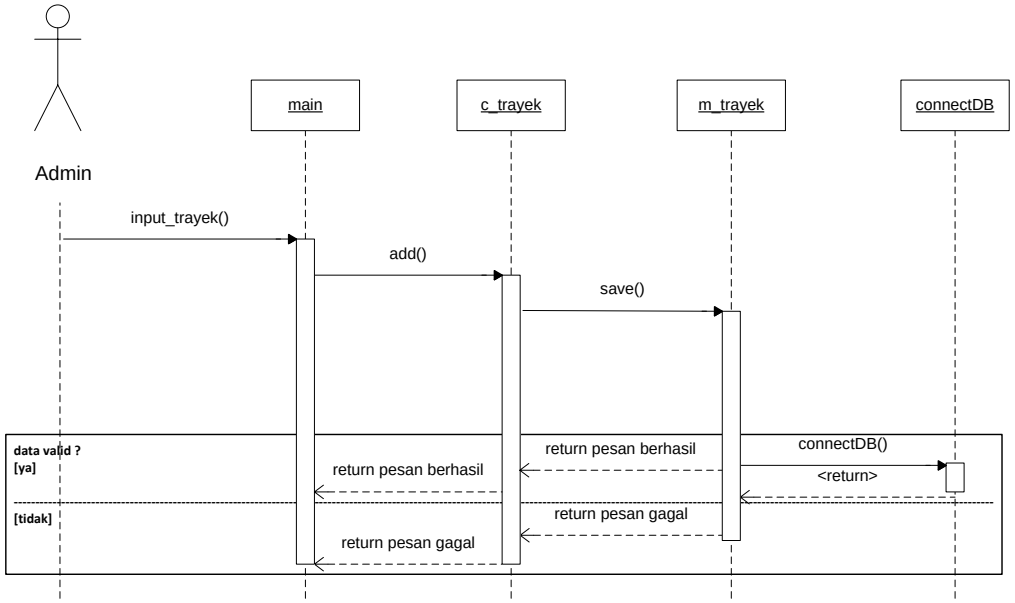
Gambar 3.24 *Sequence diagram login*

2. *Sequence diagram* ubah data admin



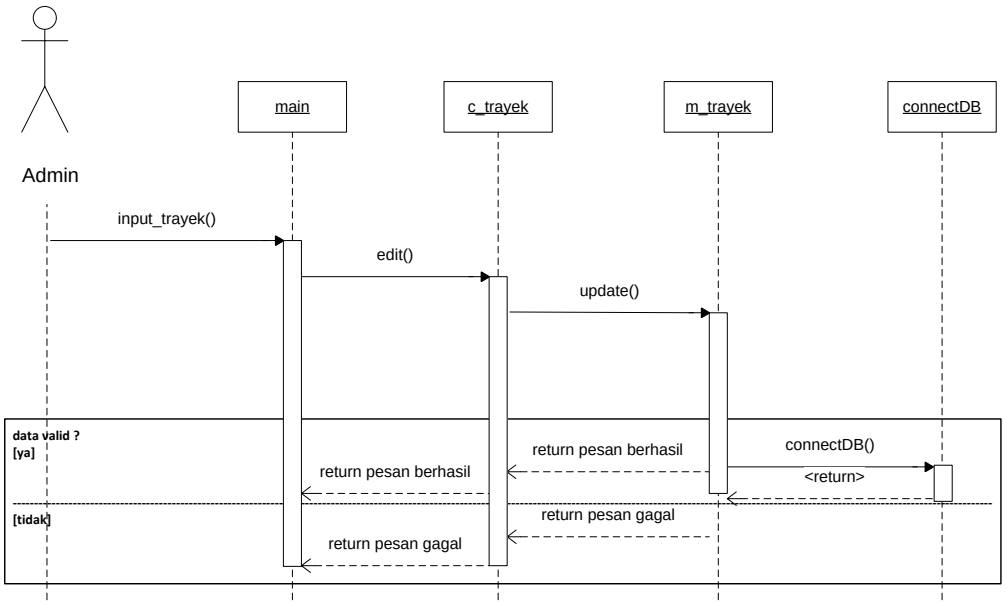
Gambar 3.25 *Sequence diagram* ubah data admin

3. *Sequence diagram* tambah data trayek



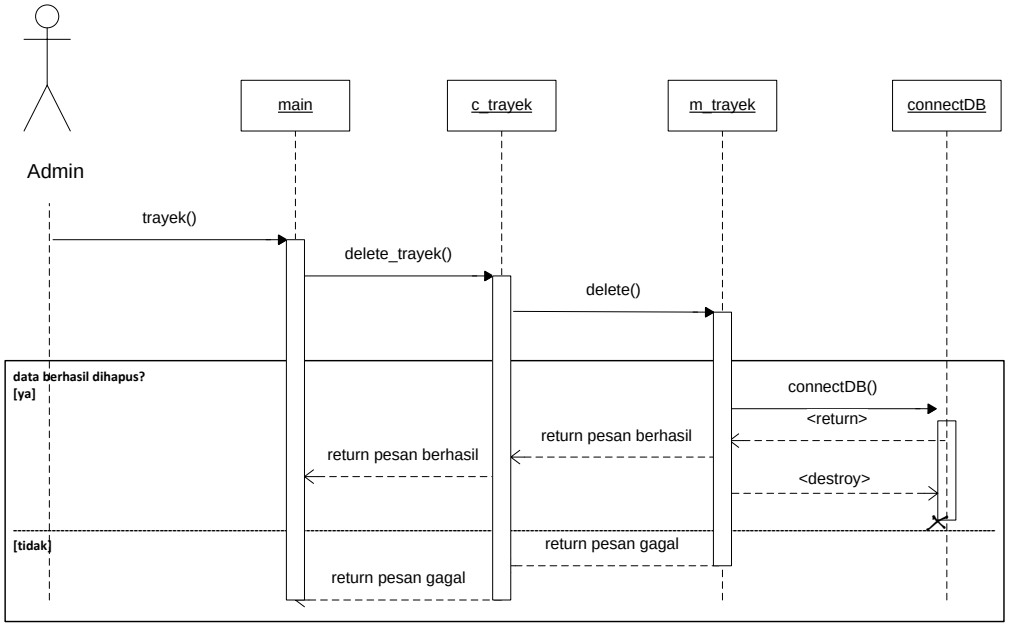
Gambar 3.26 *Sequence diagram* tambah data trayek

4. *Sequence diagram* ubah data trayek



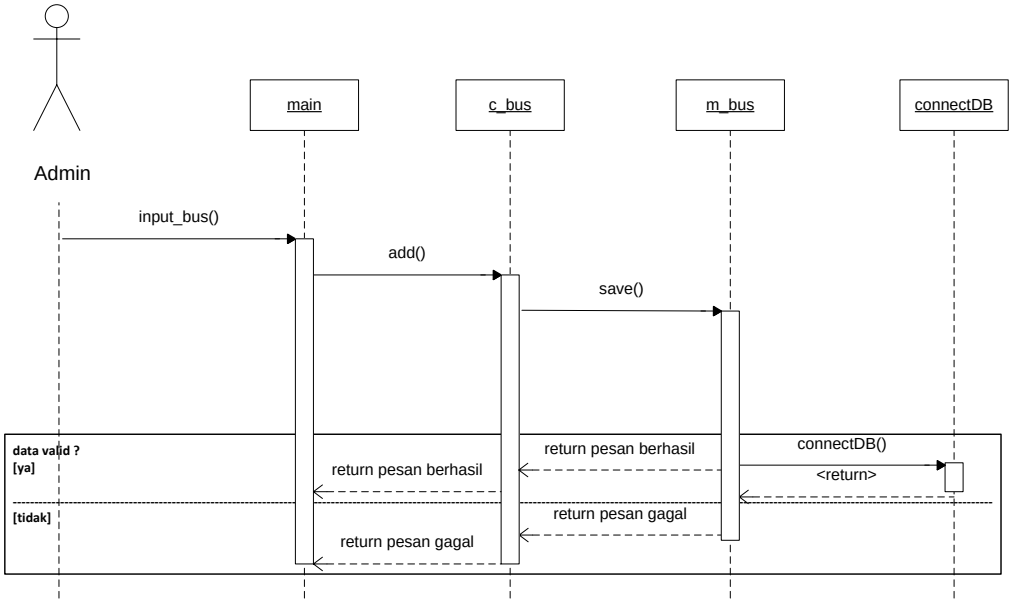
Gambar *3.27 Sequence diagram* ubah data trayek

5. *Sequence diagram* hapus data trayek



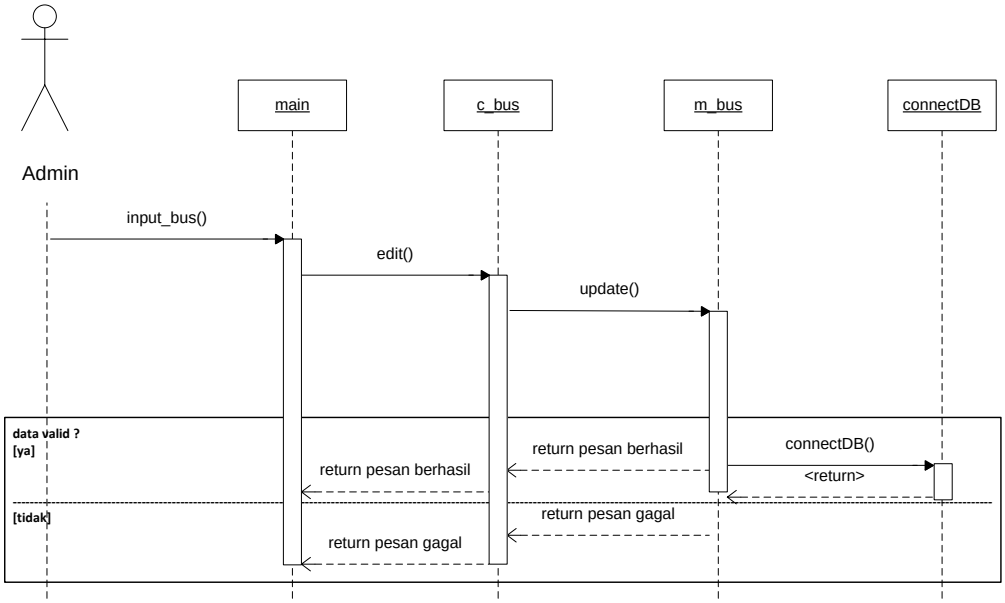
Gambar *3.28 Sequence diagram* hapus data trayek

6. *Sequence diagram* tambah data bus



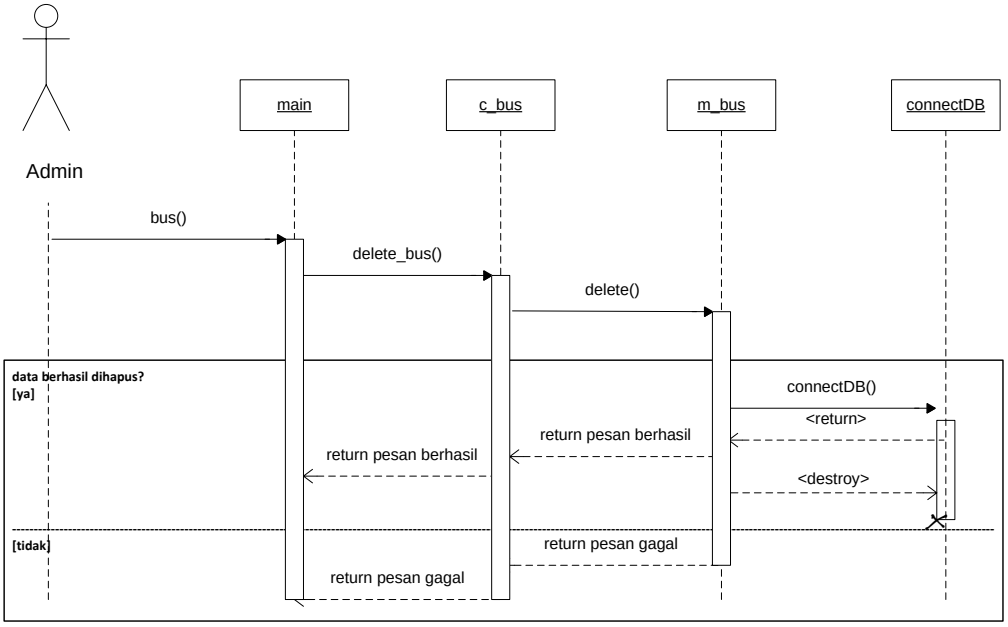
Gambar *3.29 Sequence diagram* tambah data bus

7. *Sequence diagram* ubah data bus



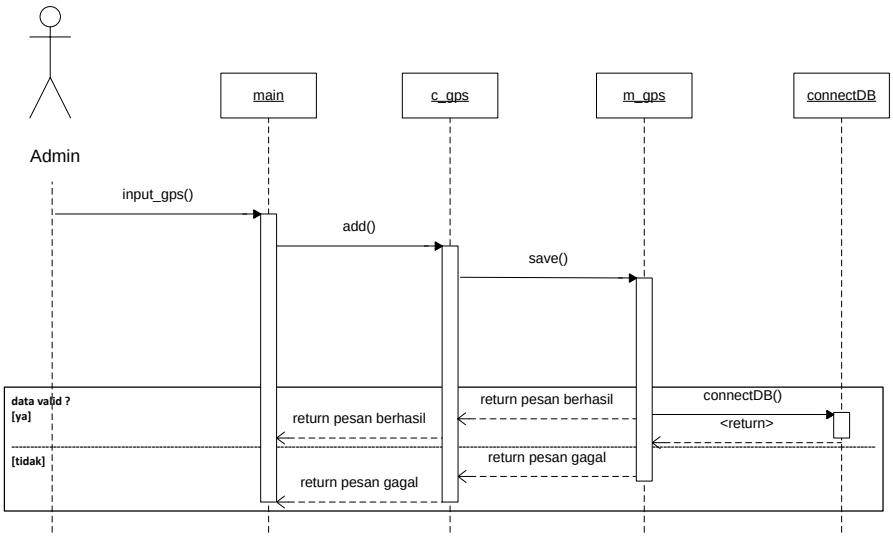
Gambar *3.30 Sequence diagram* ubah data bus

8. *Sequence diagram* hapus data bus



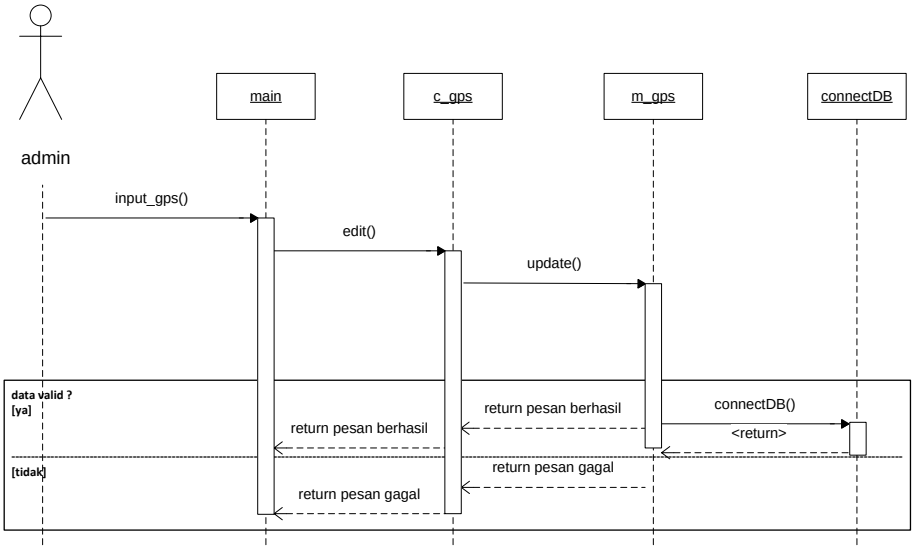
Gambar *3.31 Sequence diagram* hapus data bus

9. *Sequence diagram* tambah data gps



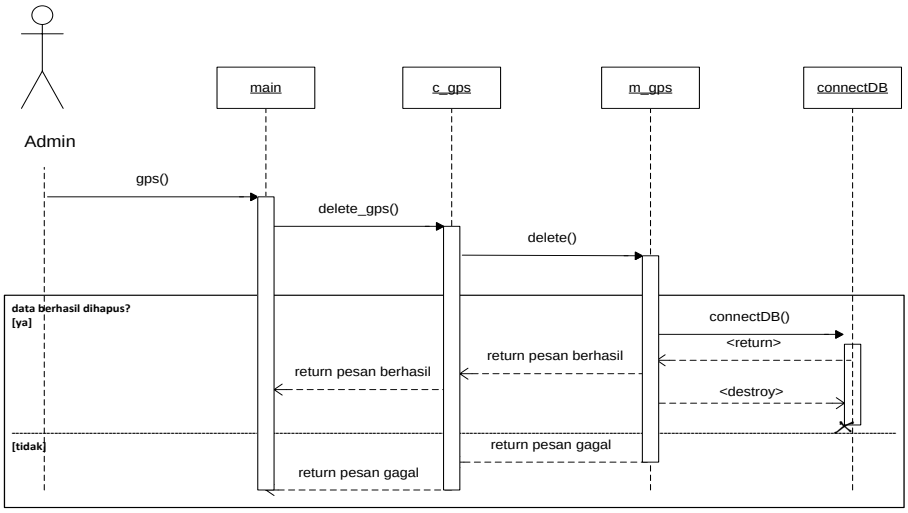
Gambar 3.32 *Sequence diagram* tambah data GPS

10. *Sequence diagram* ubah data gps



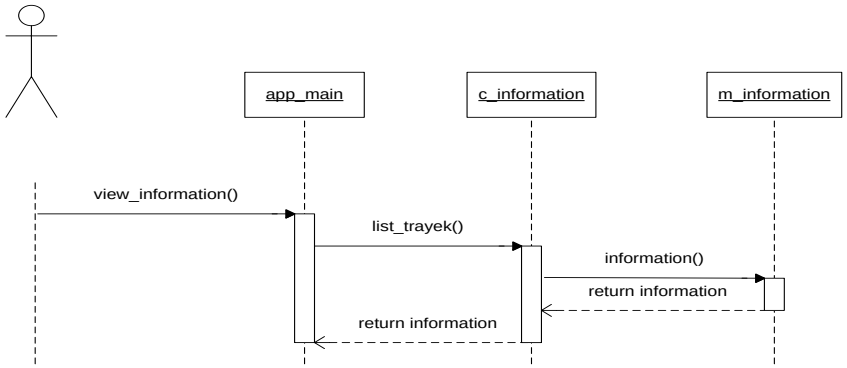
Gambar 3.33 *Sequence diagram* ubah data GPS

11. *Sequence diagram* hapus data gps



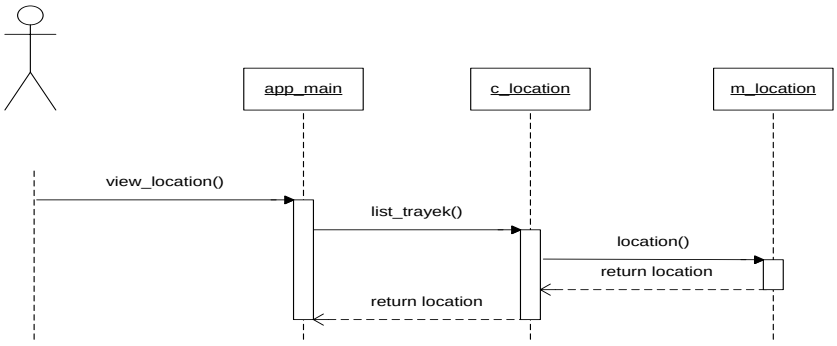
Gambar 3.34 *Sequence diagram* hapus data GPS

12. *Sequence diagram* melihat informasi umum



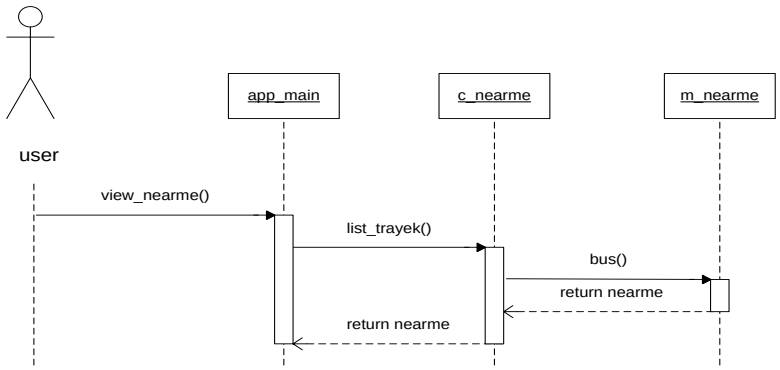
Gambar *3.35 Sequence diagram* melihat informasi umum bus DAMRI

13. *Sequence diagram* melihat lokasi keberadaan bus



Gambar 3.36 *Sequence diagram* melihat lokasi keberadaan bus DAMRI

14. *Sequence diagram* melihat DAMRI *Near Me*



Gambar 3.37 *Sequence diagram* melihat DAMRI *Near Me*

3.1.7 Perancangan Sistem

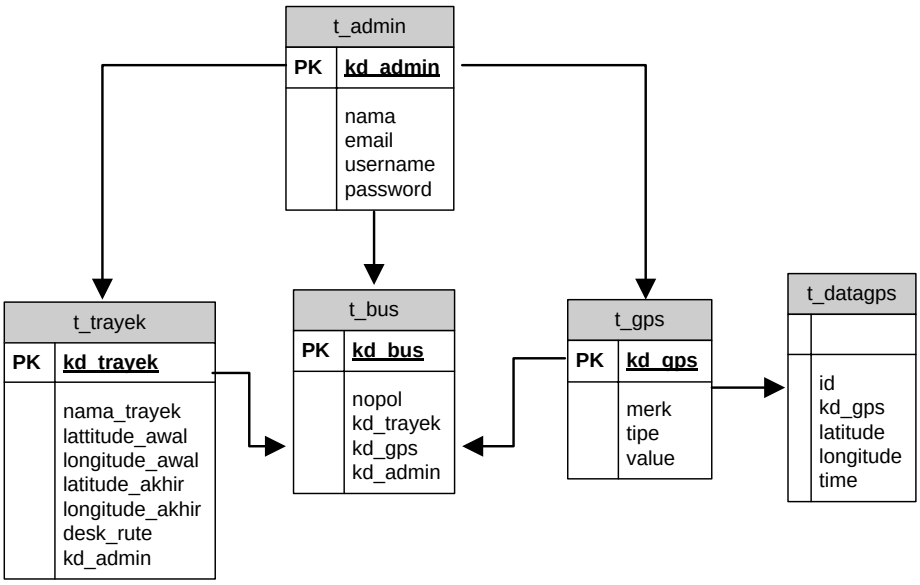
Perancangan adalah suatu bagian dari metodologi pengembangan pembangunan satuan perangkat lunak yang dilakukan setelah tahapan analisis untuk memberikan gambaran secara terperinci.

3.1.7.1 Perancangan Basis Data

Perancangan basis data merupakan tahapan untuk memetakan model konseptual ke model basis data yang akan dipakai. Perancangan basis data terbagi menjadi dua yaitu diagram relasi dan perancangan struktur tabel.

1. Diagram Relasi

Diagram relasi merupakan rangkaian hubungan antara dua tabel atau lebih pada sistem basis data. Berikut adalah perancangan data dari perangkat lunak yang akan dibangun. Dijelaskan pada Gambar 3.38



Gambar *3.38* Diagram Relasi

2. Struktur Tabel

Struktur tabel menggambarkan detail tabel yang berisi field, tipe data, dan keterangan lainnya. Berikut adalah struktur tabel dari perangkat lunak yang dibangun. Dijelaskan pada Tabel 3.26 - Tabel 3.30

a. Struktur Tabel t_admin

Tabel 3.26 struktur tabel t_admin

Nama Field	Tipe Data	Panjang	Keterangan
kd_admin	Int	10	Pri*mary Key*
Nama	Varchar	50	
Email	Varchar	50	
Username	Varchar	20	
password	MD5	32	

b. Struktur Tabel t_trayek

Tabel 3.27 struktur tabel t_trayek

Nama Field	Tipe Data	Panjang	Keterangan
kd_trayek	Int	10	*Primary key*
Nama_trayek	Varchar	60	
Latitude_awal	Varchar	60	
Longitude_awal	Varchar	60	
Latitude_akhir	varchar	60	
Longitude_akhir	varchar	60	
Desk_rute	text		
Kd_admin	Int	10	*Foreign Key*

c. Struktur Tabel t_bus

Tabel 3.28 struktur tabel t_bus

Nama Field	Tipe Data	Panjang	Keterangan
kd_bus	int	10	*Primary key*
nopol	varchar	11	
Kd_trayek	Int	10	*Foreign Key*
Kd_gps	Int	10	*Foreign Key*
Kd_admin	Int	10	*Foreign Key*

d. Struktur Tabel t_gps

Tabel 3.29 struktur tabel t_gps

Nama Field	Tipe Data	Panjang	Keterangan
Kd_gps	Int	10	*Primary Key*
merk	Varchar	60	
tipe	varchar	60	
value	Varchar	1	

e. Struktur Tabel t_datagps

Tabel 3.30 struktur tabel t_gps

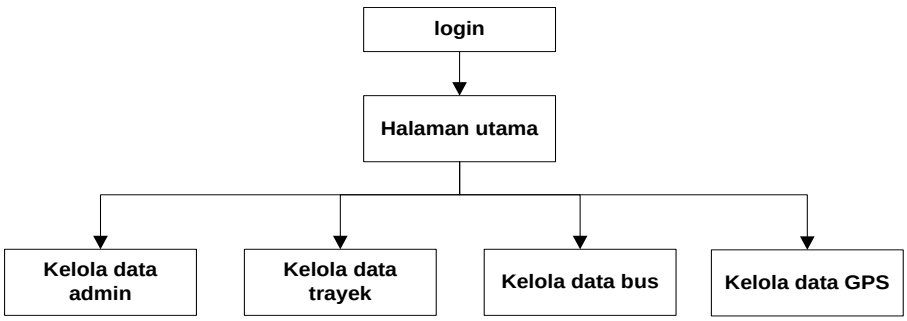
Nama Field	Tipe Data	Panjang	Keterangan
Kd_gps	Int	10	*Primary Key*
Lattitude	Varchar	60	
longitude	varchar	60	
time	datetime		

3.1.7.2 Perancangan Struktur Menu

Perancangan struktur menu dibagi menjadi 3 sesuai dengan penggunanya.

1. Perancangan struktur menu untuk Admin

Berikut adalah perancangan struktu menu untuk Admin pada Gambar 3.39



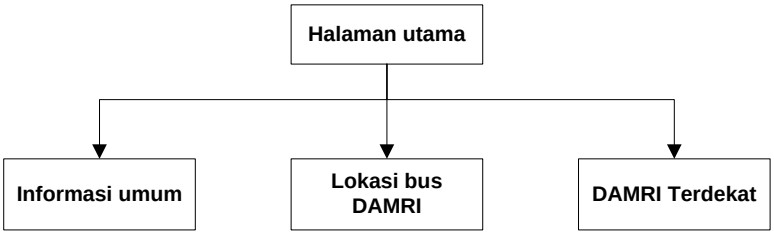
Gambar *3.39* Perancangan Struktur Menu Admin

Keterangan dari setiap menu yang ada untuk admin

a. Login : digunakan admin untuk masuk kedalam sistem web agar dapat mengolah data yang tersedia.
b. Halaman utama : setelah admin berhasil login, admin akan langsung masuk ke halaman utama dimana terdapat menu yang tersedia disebelah kiri untuk menuju pengolahan data
c. Kelola data admin : pada menu ini, admin disediakan informasi mengenai admin itu sendiri dan dapat melakukan pengubahan datanya sendiri.
d. Kelola data trayek : digunakan oleh admin untuk menambah, mengubah dan menghapus data trayek.
e. Kelola data bus : digunakan oleh admin untuk menambah, mengubah dan menghapus data bus.
f. Kelola data GPS : digunakan oleh admin untuk menambah, mengubah dan menghapus data GPS.

2. Perancangan struktur menu untuk *User*

Berikut adalah perancangan struktu menu untuk *User* padaGambar 3.40



Gambar *3.40* Perancangan Struktur Menu *User*

Keterangan dari setiap menu yang ada untuk *user*

a. Halaman utama : tampilan awal ketika aplikasi dibuka oleh *user.*
b. Informasi umum : pada menu ini disediakan informasi umum mengenai trayek yang ada dimana terdapat rute yang dilalui, jumlah bus yang ada pada trayek tersebut dan adanya peta yang menggambarkan rute yang dilalui oleh bus.
c. Lokasi bus DAMRI : digunakan oleh *user* untuk melihat lokasi keberadaan bus DAMRI berada setelah memilih salah satu trayek yang ada.

d. DAMRI Terdekat : pada menu ini sistem akan langsung memfokuskan ke lokasi keberadaan pengguna pada map. Selain itu, akan dimunculkan seluruh marker bus yang menggunakan GPS pada map. Karena sistem langsung memfokuskan ke lokasi pengguna, maka *marker* yang terlihat hanya yang ada disekitar pengguna. Jika *user* ingin melihat seluruh bus yang ada, *user* hanya tinggal men-*zoom out* pada map.

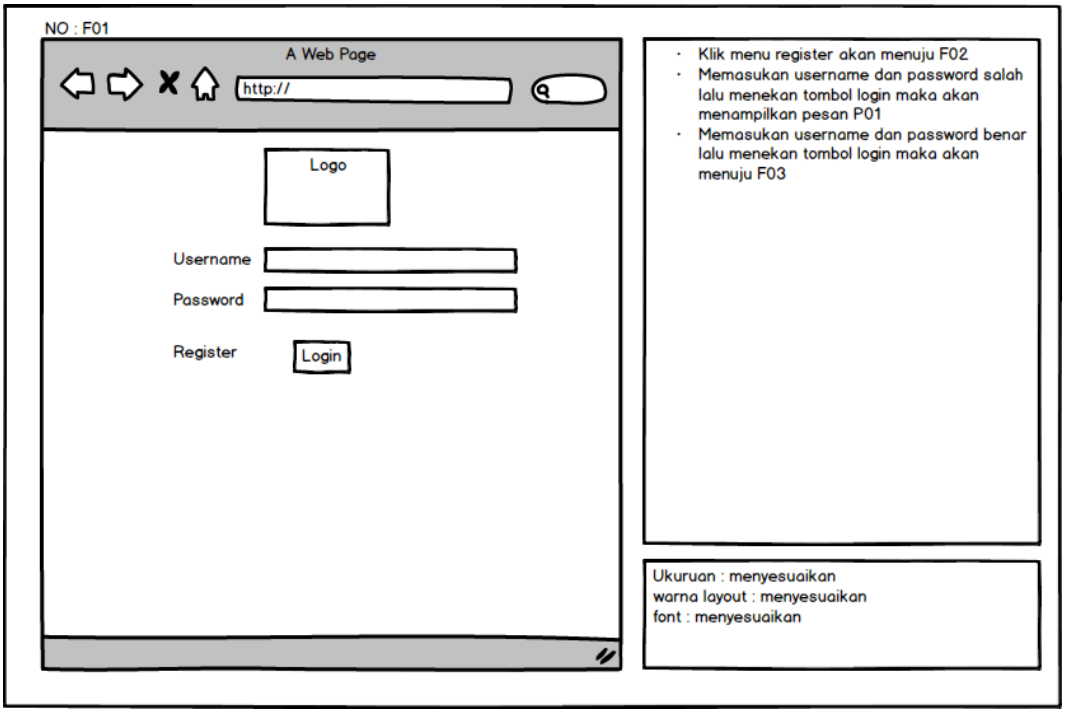
3.1.7.3 Perancangan Antarmuka

Perancangan antarmuka mendeskripsikan rencana tampilan yang akan digunakan pada aplikasi yang akan dibangun. Perancangan antarmuka yang ada terdiri dari Form dan pesan. Dijelaskan pada Gambar 3.41 - Gambar 3.63

1. Perancangan *Form*

a. Perancangan antarmuka *login*

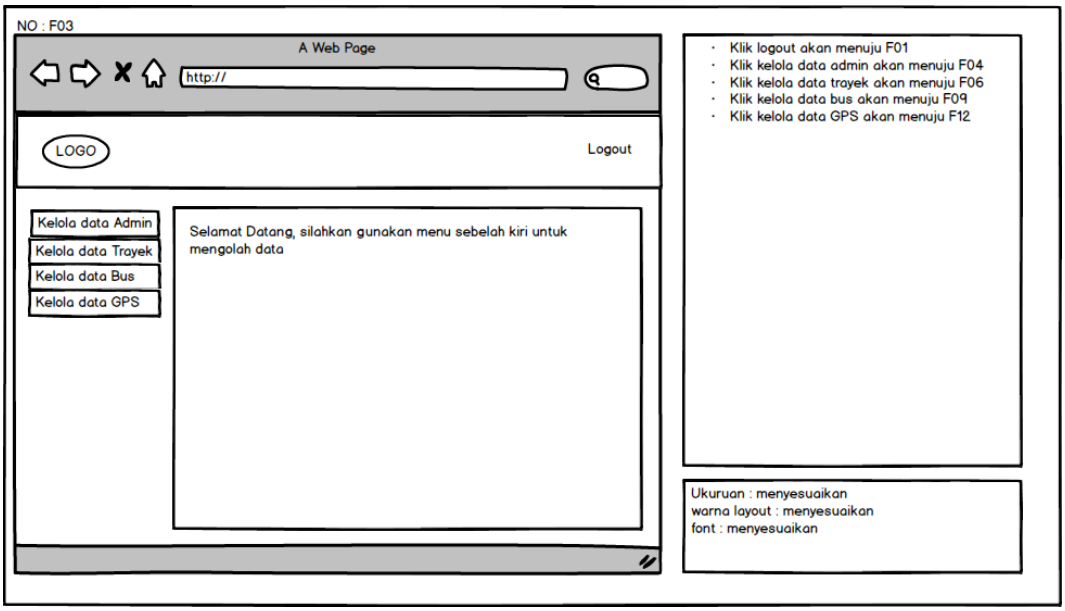
Halaman ini digunakan untuk masuk ke halaman pengolahan data oleh superadmin atau admin.



Gambar *3.41* Perancangan antarmuka *login*

b. Perancangan antarmuka halaman utama admin

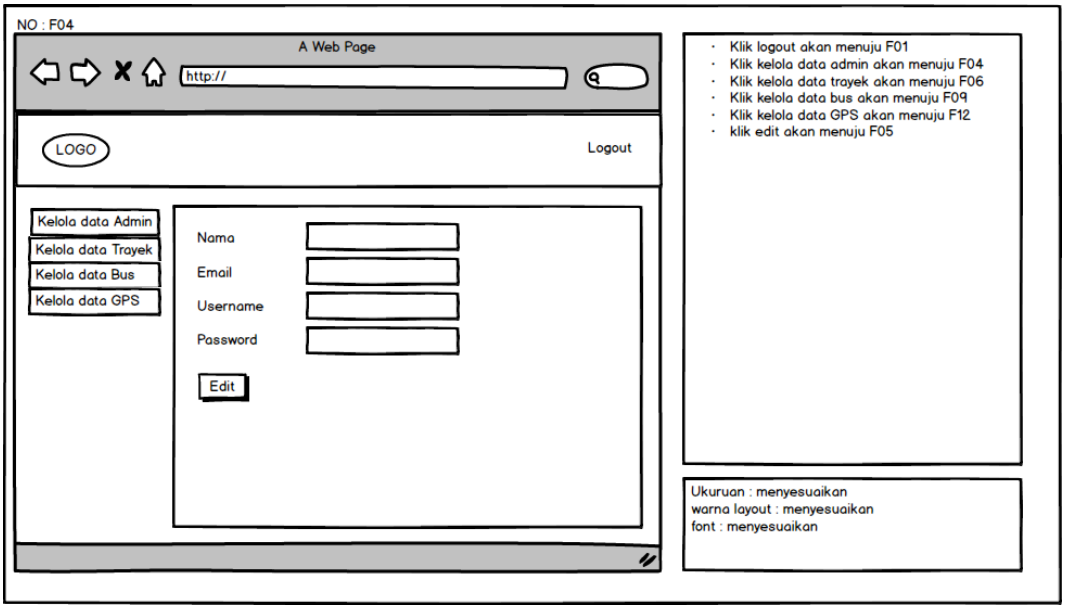
Setelah *login* akan masuk ke halaman utama dimana terdapat menu pengolahan data admin.



Gambar *3.42* Perancangan antarmuka halaman utama admin

c. Perancangan antarmuka kelola data admin

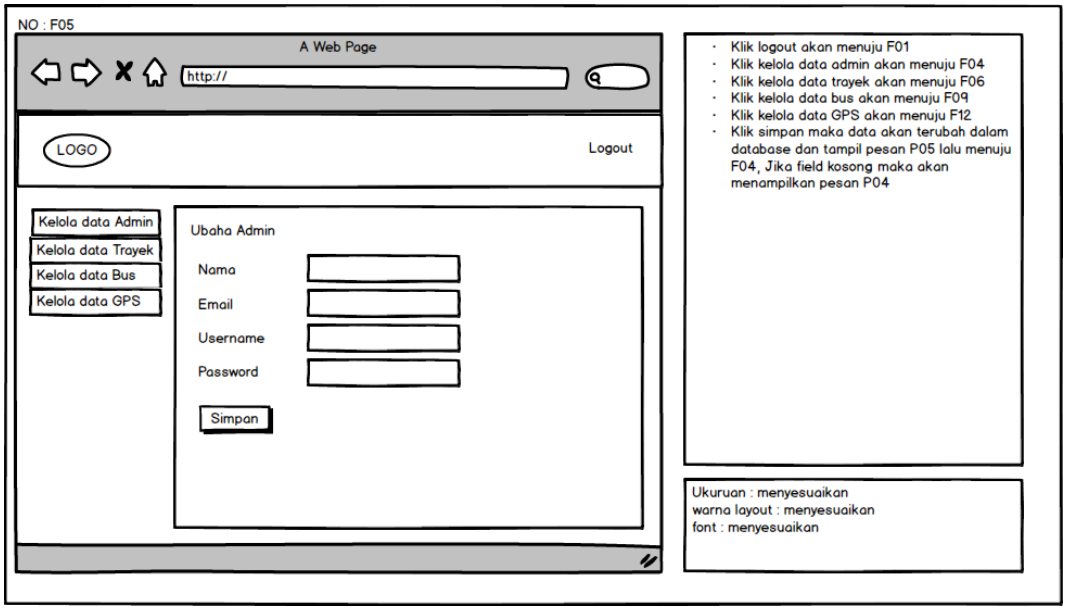
Halaman ini digunakan untuk mengelola data admin



Gambar *3.43* Perancangan antarmuka kelola data admin

d. Perancangan antarmuka ubah data admin

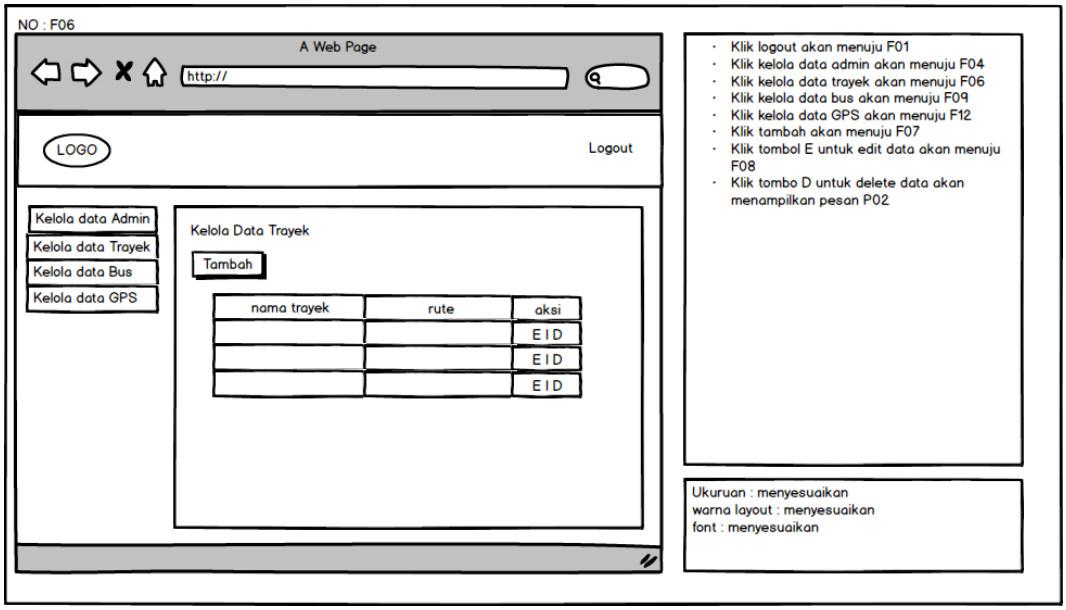
Halaman ini digunakan untuk mengubah data admin



Gambar *3.44* Perancangan antarmuka ubah data admin

e. Perancangan antarmuka kelola data trayek

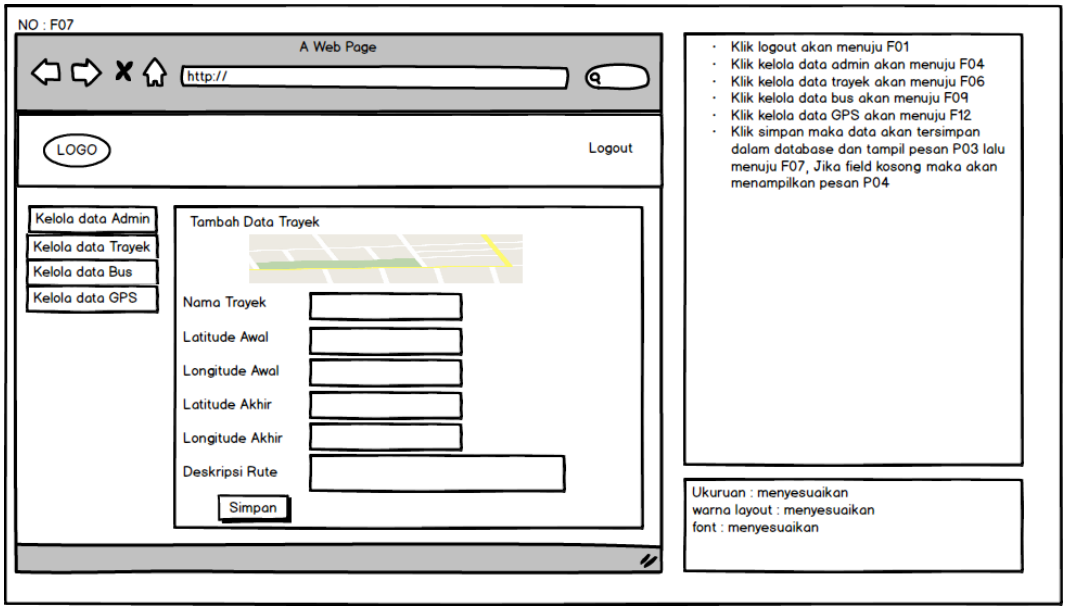
Halaman ini digunakan untuk mengelola data trayek



Gambar *3.45* Perancangan antarmuka kelola data trayek

f. Perancangan antarmuka tambah data trayek

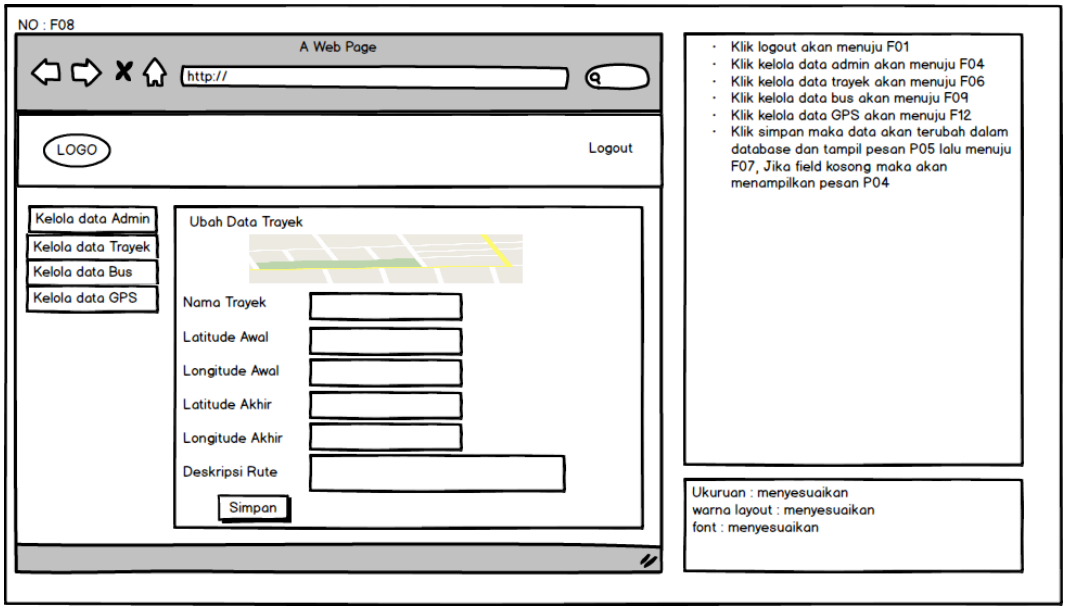
Halaman ini digunakan untuk menambah data trayek



Gambar 3.46 Perancangan antarmuka tambah data trayek

g. Perancangan antarmuka ubah data trayek

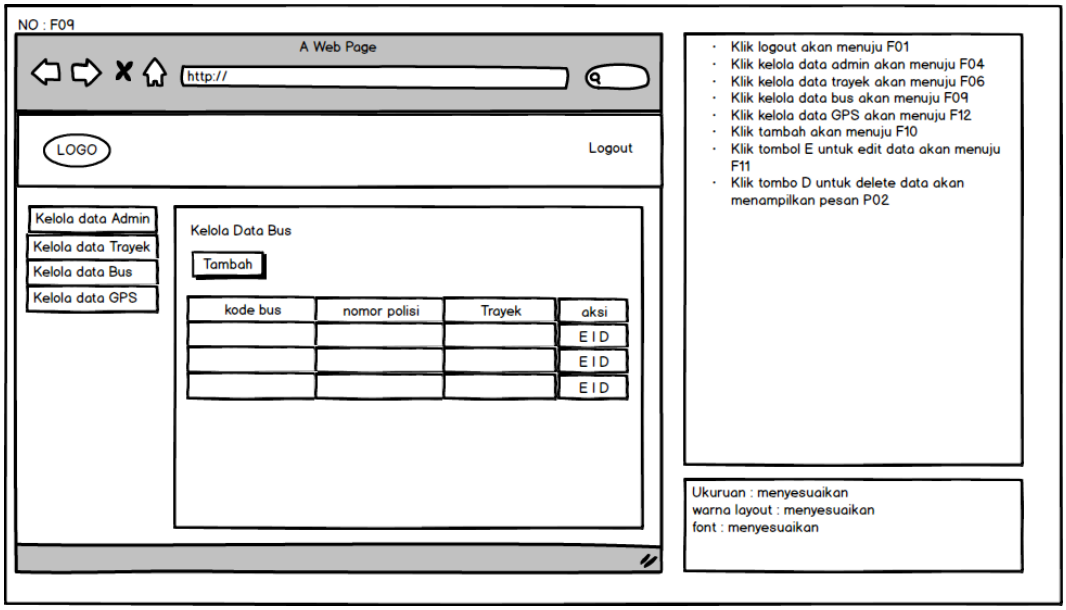
Halaman ini digunakan untuk mengubah data trayek



Gambar 3.47 Perancangan antarmuka ubah data trayek

h. Perancangan antarmuka kelola data bus

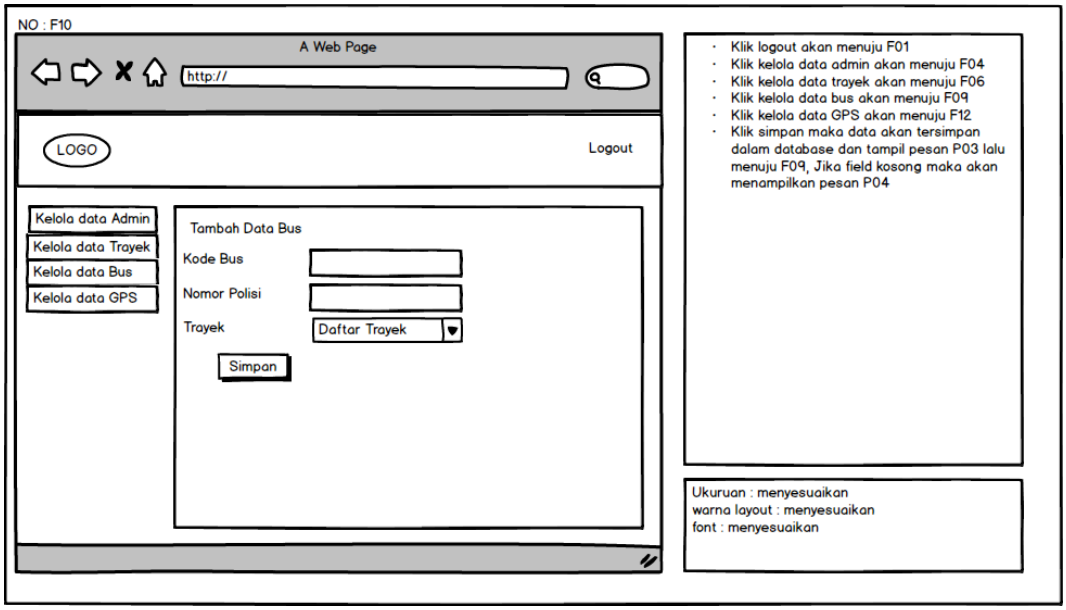
Halaman ini digunakan untuk mengelola data bus



Gambar *3.48* Perancangan antarmuka kelola data bus

i. Perancangan antarmuka tambah data bus

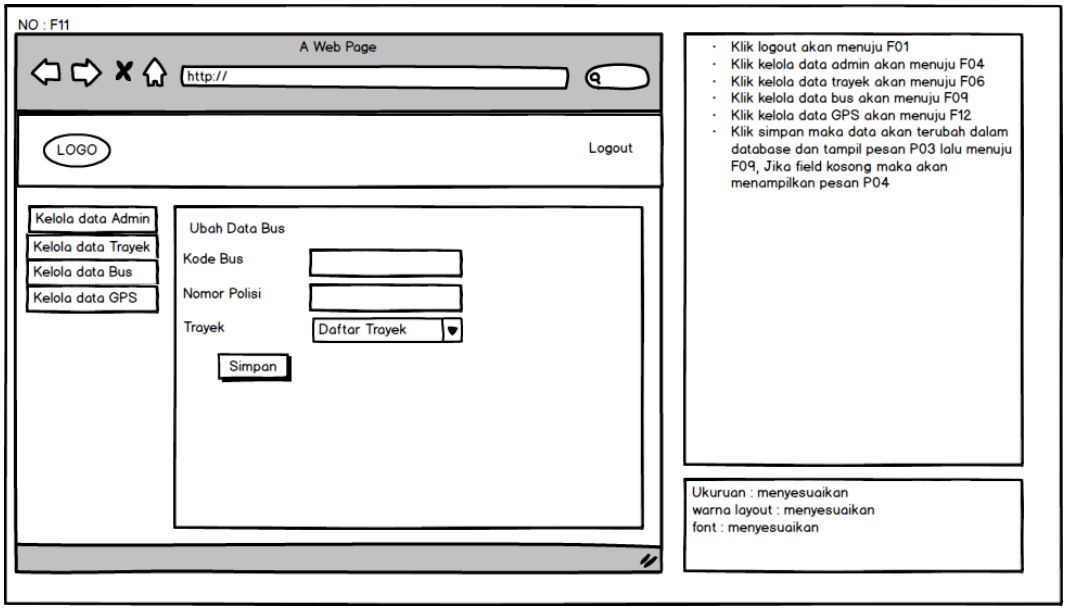
Halaman ini digunakan untuk menambah data bus



Gambar *3.49* Perancangan antarmuka tambah data bus

j. Perancangan antarmuka ubah data bus

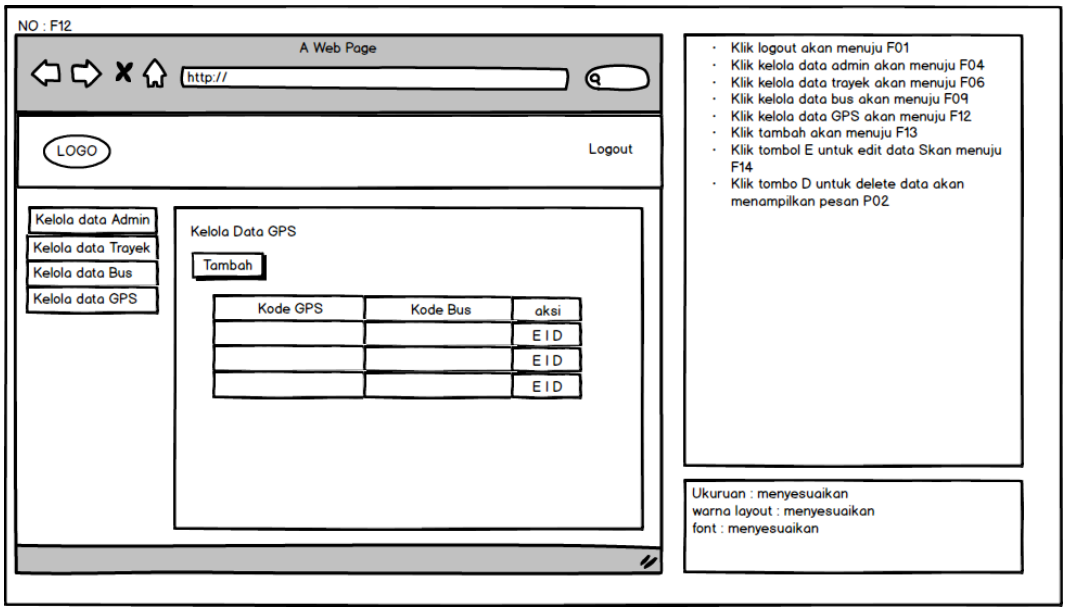
Halaman ini digunakan untuk mengubah data bus



Gambar *3.50* Perancangan antarmuka ubah data bus

k. Perancangan antarmuka kelola data gps

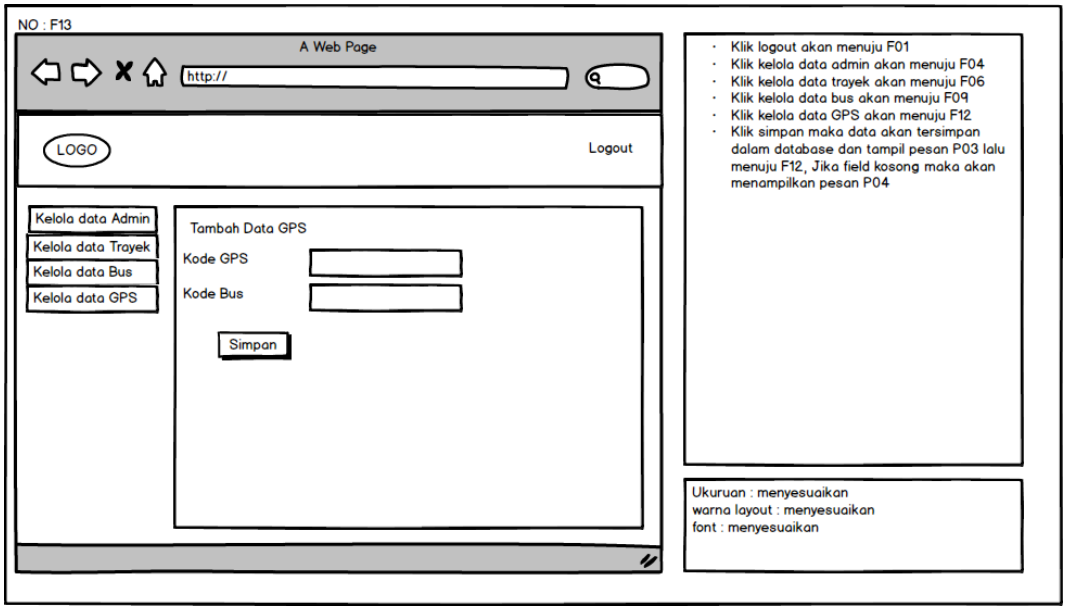
Halaman ini digunakan untuk mengelola data gps



Gambar *3.51* Perancangan antarmuka kelola data GPS

l. Perancangan antarmuka tambah data gps

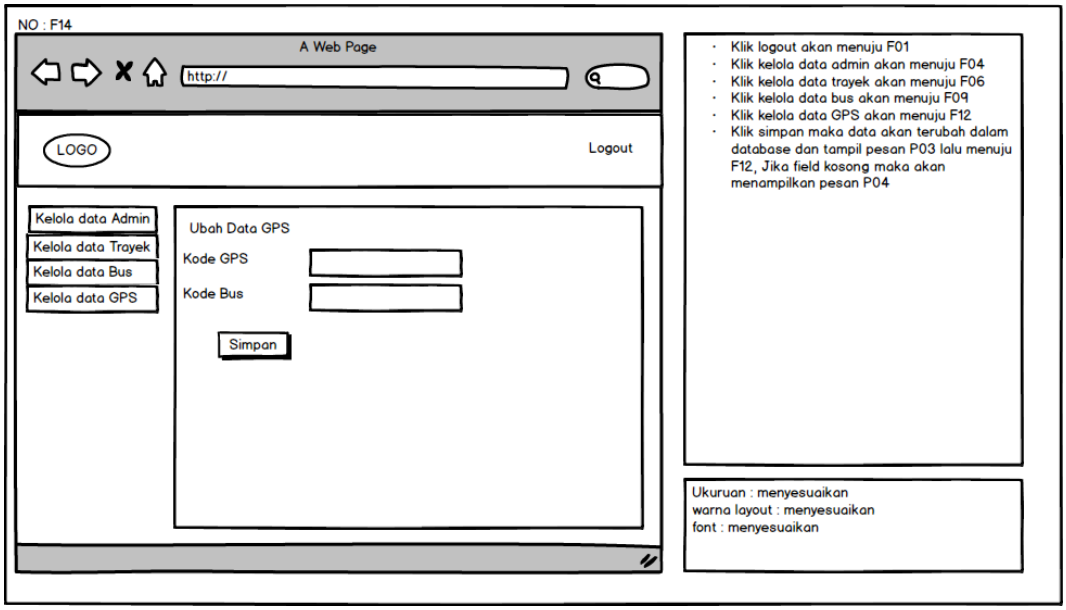
Halaman ini digunakan untuk menambah data gps



Gambar *3.52* Perancangan antarmuka tambah data GPS

m. Perancangan antarmuka ubah data gps

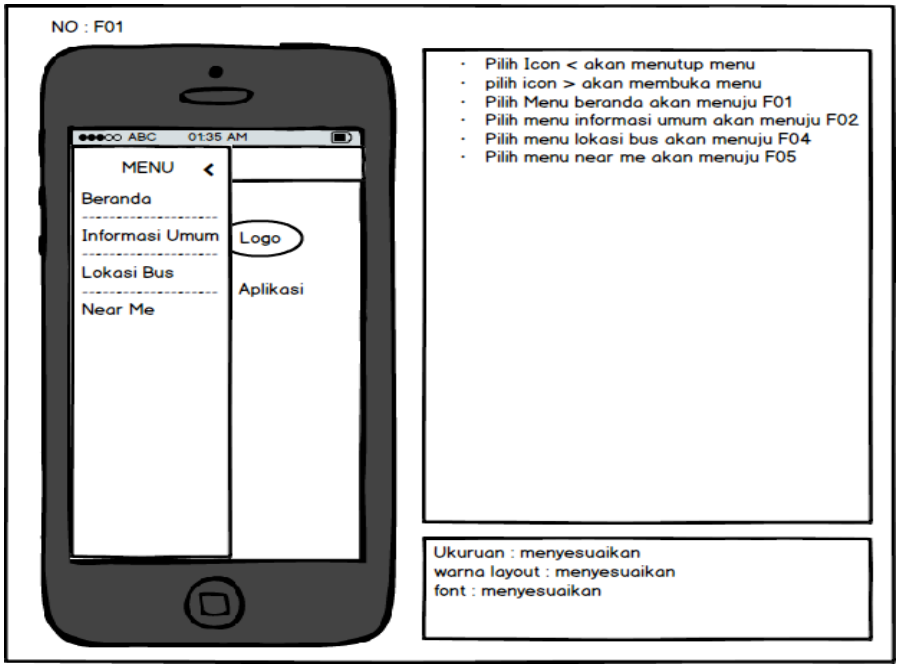
Halaman ini digunakan untuk mengubah data gps



Gambar 3.53 Perancangan antarmuka ubah data GPS

n. Perancangan antarmuka halaman utama *user*

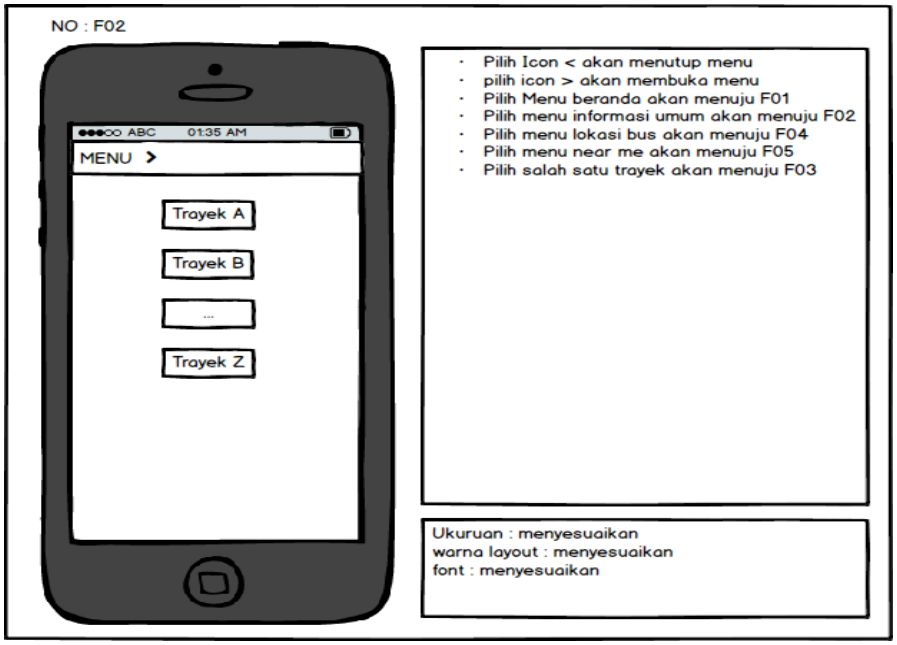
Form ini digunakan sebagai halaman utama *user* menggunakan aplikasi



Gambar 3.54 Perancangan antarmuka halaman utama *user*

o. Perancangan antarmuka trayek bus DAMRI

Form ini digunakan *user* untuk memilih trayek yang ada.



Gambar 3.55 Perancangan antarmuka trayek bus DAMRI

p. Perancangan antarmuka informasi umum trayek dan bus DAMRI

Form ini digunakan *user* untuk melihat informasi umum trayek dan bus DAMRI terpilih.



Gambar *3.56* Perancangan antarmuka informasi umum trayek & bus DAMRI

q. Perancangan antarmuka lokasi keberadaan bus DAMRI

Form ini digunakan *user* untuk mengetahui lokasi keberadaan bus DAMRI.



Gambar *3.57* Perancangan antarmuka lokasi keberadaan bus DAMRI

r. Perancangan antarmuka DAMRI *near me*

Form ini digunakan *user* untuk mengetahui lokasi keberadaan bus DAMRI yang ada disekitar *user.*

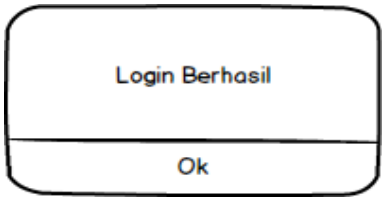


Gambar *3.58* Perancangan antarmuka DAMRI *Near Me*

2. Perancangan Pesan

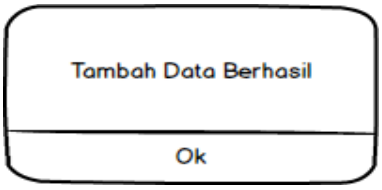
Berikut perancangan pesan dalam membangun aplikasi ini.

a. Perancangan Pesan P01



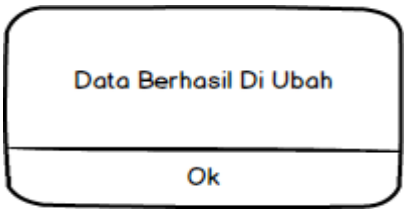
Gambar *3.59* Perancangan pesan P01

b. Perancangan Pesan P02



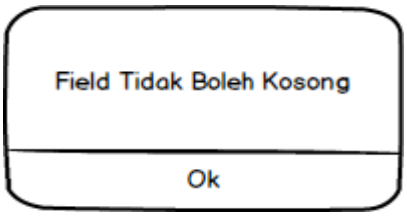
Gambar 3.60 Perancangan pesan P02

c. Perancangan Pesan P03



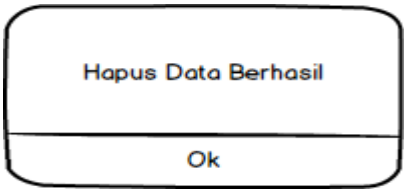
Gambar *3.61* Perancangan pesan P03

d. Perancangan Pesan P04



Gambar *3.62* Perancangan pesan P04

e. Perancangan Pesan P05

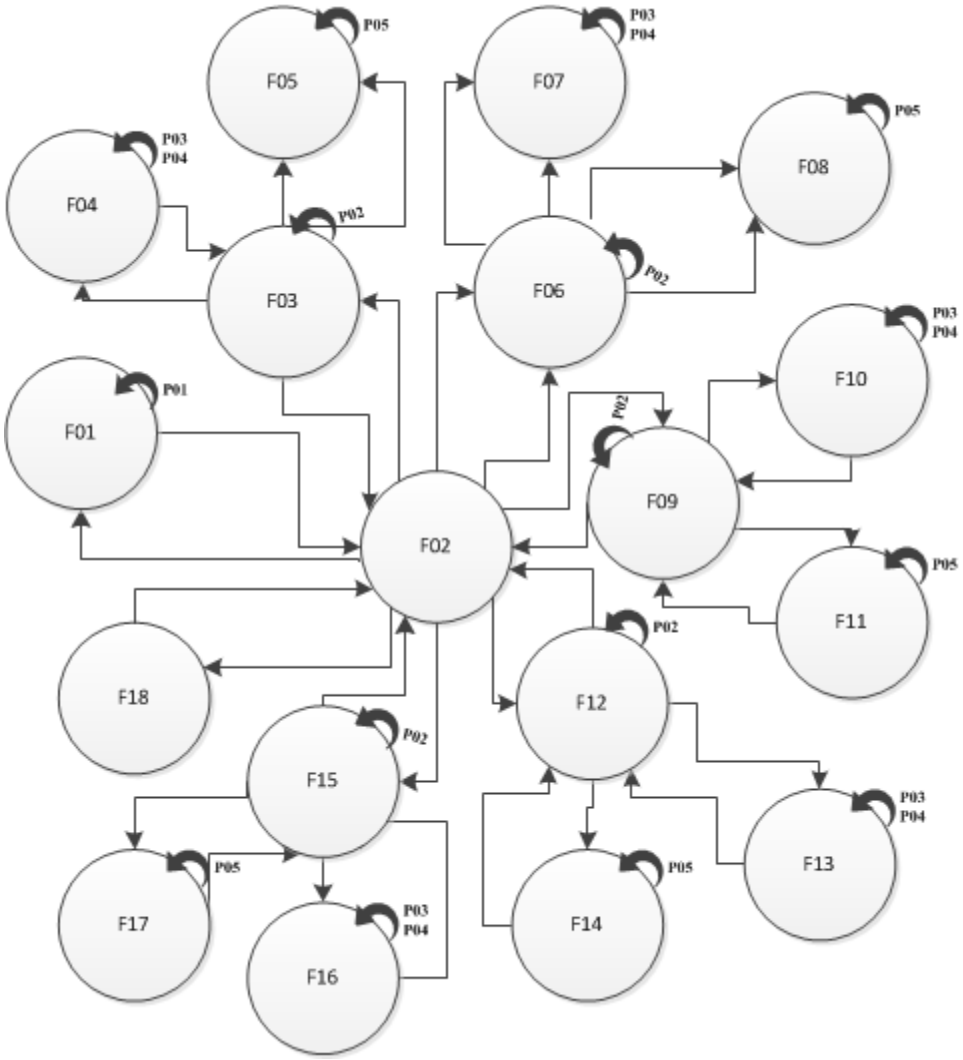


Gambar *3.63* Perancangan pesan P05

3.1.7.4 Jaringan Semantik

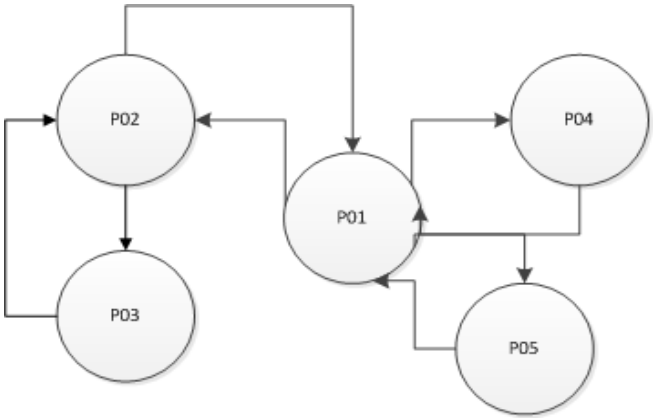
Jaringan semantik menggambarkan keterhubungan navigasi menu dari satu form ke form lainnya. Berikut jaringan semantik dalam membangun aplikasi ini. Dijelaskan pada Gambar 3.64 - Gambar 3.65

1. Jaringan Semantik Admin



Gambar *3.64* Jaringan Semantik Admin

2. Jaringan Semantik *User*



Gambar *3.65* Jaringan Semantik *User*