

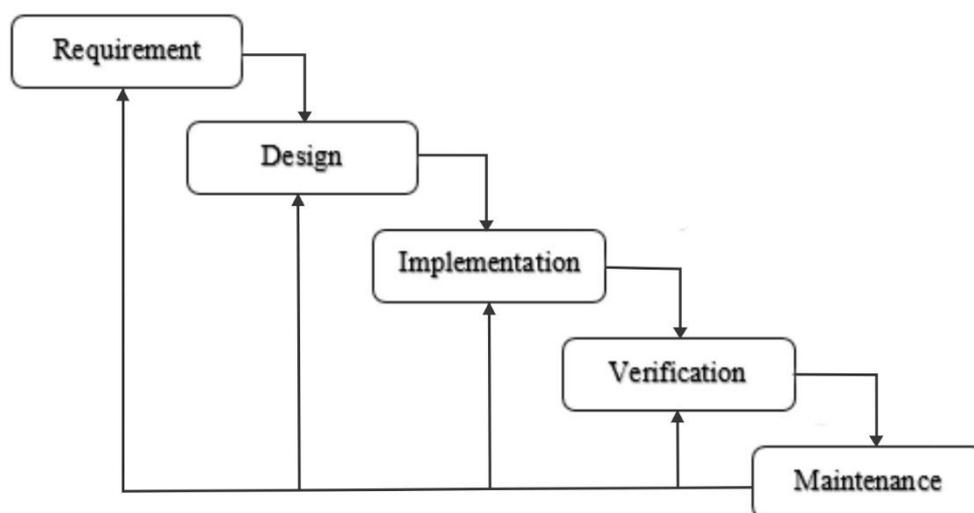
BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Penelitian

Penelitian pembuatan sistem informasi *showroom* Lampung Jaya Motor berbasis web menggunakan metode *Software Development Life Cycle* (SDLC) model *Waterfall*. Metode SDLC dengan model *Waterfall* dipilih dengan alasan untuk memudahkan peneliti dalam menentukan pencapaian suatu sistem.

Dalam model *Waterfall*, setiap tahapan harus dilakukan secara berurutan dan tidak dapat lanjut ke tahap berikutnya tanpa melakukan tahapan sebelumnya yang artinya bahwa peneliti tidak dapat lanjut ke tahap kedua ataupun seterusnya jika belum menyelesaikan proses pada tahap pertama. Hal ini juga dijadikan sebuah pertimbangan oleh peneliti dalam membuat sistem informasi *showroom* Lampung Jaya Motor ini, karena sistem ini hanya dikerjakan oleh satu orang dan untuk mempermudah pembuatan sistem maka peneliti harus mengembangkan sistem dengan tahap yang berurutan juga.

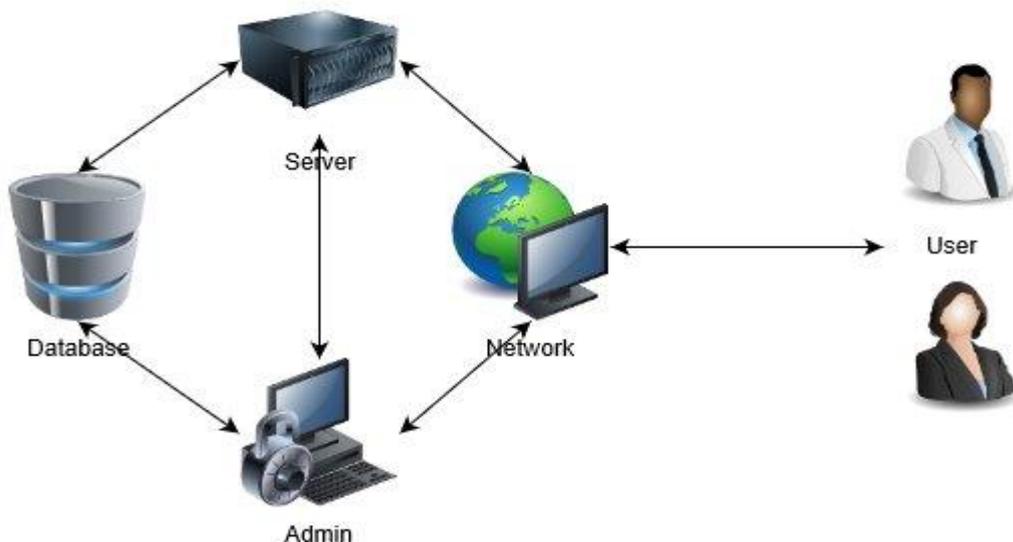


Gambar 3.1 Metode Waterfall SDLC

Tahapan dalam metode SDLC menggunakan model *Waterfall* pada pembangunan aplikasi ini adalah :

1. *Requirements System* – merupakan tahap perencanaan sistem yang menyangkut kebutuhan pengguna, kelayakan baik secara teknik maupun secara teknologi.
2. *Design System* – merupakan proses perancangan sistem mulai dari pembagian persyaratan dalam sistem perangkat keras maupun perangkat lunak.
3. *Implementation System* – merupakan proses perancangan perangkat lunak direalisasikan dengan program atau unit program.
4. *Verification System* – merupakan proses pengujian unit program atau program individual yang diintegrasikan dan diuji sebagai sistem yang lengkap untuk menjamin bahwa kebutuhan sistem telah terpenuhi.
5. *Operation and Maintenance* – merupakan proses pengoperasian sistem pada lingkungannya dan melakukan pemeliharaan sistem. Tahap ini merupakan tahap siklus hidup yang lama.

3.2 Arsitektur Aplikasi



Gambar 3.2 Arsitektur *Website Showroom Lampung Jaya Motor*

Pada **Gambar 3.2** dalam penggunaan *database* yang diterapkan pada sistem adalah MySQL. Untuk bahasa pemrograman yang digunakan adalah PHP dengan menggunakan *framework CodeIgniter*. Pada saat pengguna akan mengakses *website*, maka digunakanlah *internet* dan web *browser* sebagai perantara untuk menampilkan atau memuat antar muka yang diambil serta dibutuhkan dari *database*. Melalui tampilan yang sudah dimuat, pengguna akan mendapatkan informasi yang dibutuhkan.

3.3 Teknik Pengumpulan Data

Dalam penelitian ini, dilakukan analisis kebutuhan yang mencakup kebutuhan perangkat keras, perangkat lunak dan *requirement system* pada aplikasi. Sebelum membuat *website*, terlebih dahulu dilakukan pengumpulan kebutuhan dengan metode wawancara kepada pemilik *Showroom* Lampung Jaya Motor. Wawancara yang dilakukan akan menghasilkan data terkait bagaimana fitur-fitur yang *website* yang diharapkan.

3.4 Analisis Kebutuhan

Analisis kebutuhan didapat dari hasil wawancara dan observasi. Berikut adalah analisis kebutuhan *website* yang dibuat agar dapat digunakan oleh admin dan *user*.

- a. Kebutuhan Admin
 1. Admin dapat login pada sistem.
 2. Admin dapat mengelola (*create, read, update, delete*) data pada sistem.
 3. Admin dapat *logout* pada sistem.
- b. Kebutuhan *User*
 1. *User* dapat melihat daftar kendaraan yang dijual.
 2. *User* dapat melihat harga kendaraan yang dijual.
 3. *User* dapat melihat foto kendaraan yang dijual.
 4. *User* dapat melihat info tentang *detail* kendaraan yang dijual.

5. *User* dapat melihat info tentang penjual.
6. *User* dapat mengirim order kepada penjual.

3.5 Peralatan Pendukung

Dalam penelitian menggunakan beberapa peralatan pendukung sebagai perangkat untuk mendukung berjalannya perancangan dan implementasi *website*, antara lain :

3.5.1 Alat

Alat yang digunakan berupa perangkat keras (*hardware*) dan perangkat lunak (*software*) :

1. Perangkat Keras (*hardware*)
 - Personal Computer (PC)/Laptop :
 - a. Intel® Core™ i5-5200U CPU
 - b. 4.00 GB RAM
 - c. 64-bit *Operating System*
2. Perangkat Lunak (*software*)
 - a. Web Server : XAMPP
 - b. Text Editor : Sublime Text 3
 - c. Database Server : MySql

3.5.2 Bahan

Bahan yang digunakan dalam penelitian ini adalah data-data kendaraan yang diberikan oleh pengelola *showroom* Lampung Jaya Motor. Data-data yang diberikan berupa gambar, video dan detail kendaraan.

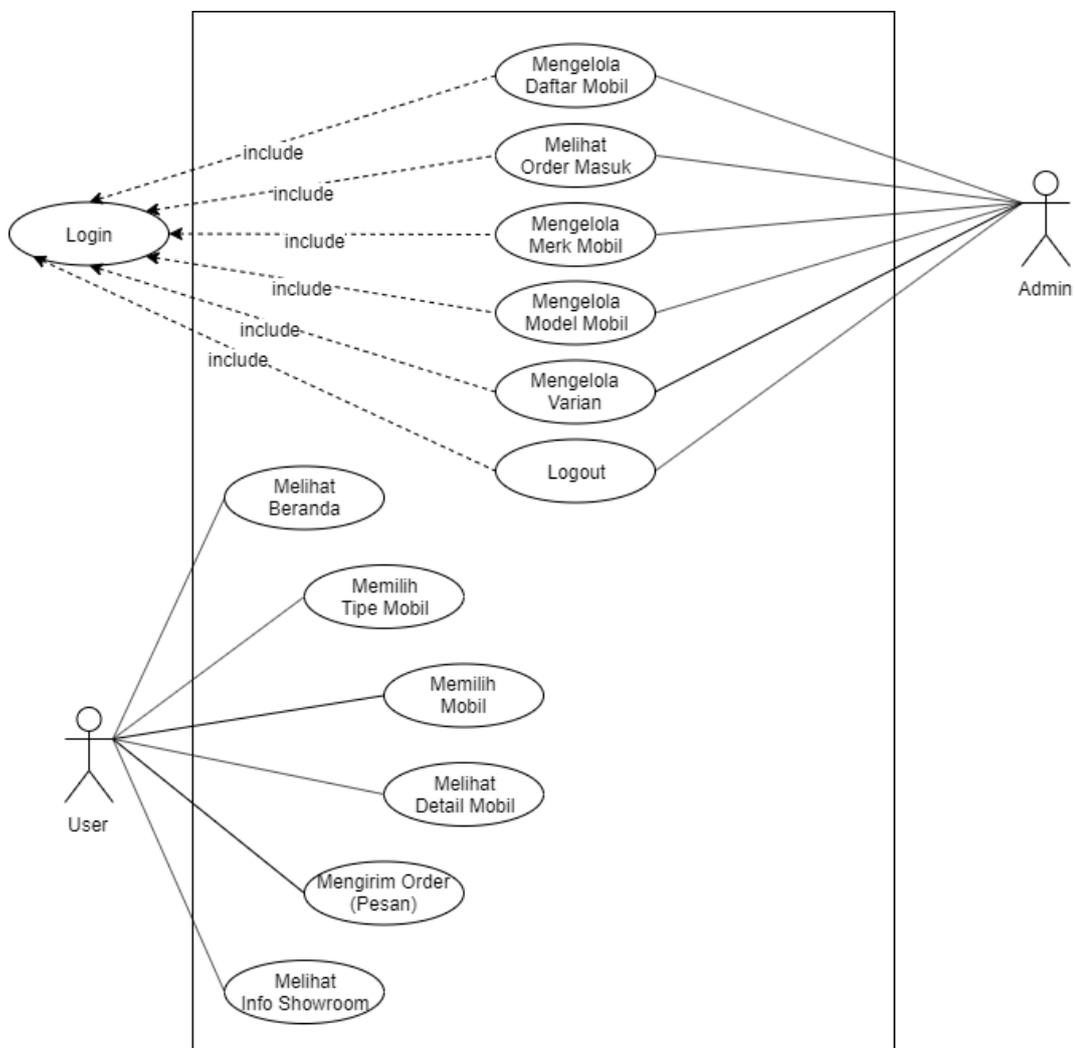
3.6 Desain Sistem

Dalam perancangan *logic* aplikasi ini menggunakan metode SDLC maka hal yang harus dilakukan berikutnya adalah membuat rancangan atau desain sistem. Dalam pengembangan aplikasi ini beberapa rancangan yang dibuat adalah *use case*

diagram, *entity relationship* diagram, *activity* diagram, *class* diagram dan desain *user interface*.

3.6.1 Use case diagram

Berdasarkan analisis kebutuhan pada sub bab 3.4, dibuatlah *use case* diagram untuk membantu perancangan logic pada aplikasi web ini. Pada aplikasi web ini terdapat dua *actor* yaitu admin dan *user*. Gambaran *use case* diagram dapat dilihat pada Gambar 3.3.



Gambar 3.3 Use Case Diagram Admin dan User

Penjelasan tentang Use Case Diagram Admin pada Gambar 3.3 :

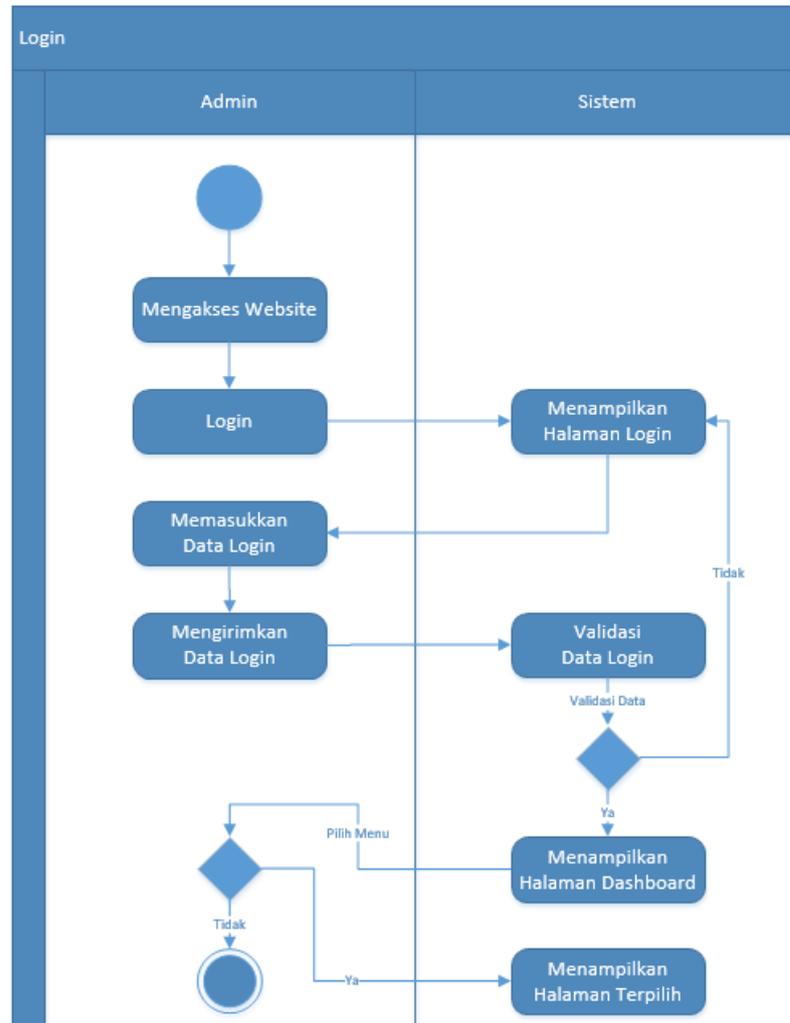
- a. *Login* : memungkinkan admin dapat menggunakan aplikasi web.
- b. Melihat *Dashboard* : memungkinkan admin dapat melihat *dashboard*.
- c. Mengelola Daftar Mobil : memungkinkan admin dapat mengelola daftar mobil.
- d. Melihat Pesan Masuk : memungkinkan admin dapat melihat pesan masuk.
- e. Mengelola *Merk* Mobil : memungkinkan admin dapat mengelola *merk* mobil.
- f. Mengelola *Model* Mobil : memungkinkan admin dapat mengelola *model* mobil.
- g. Mengelola Varian Mobil : memungkinkan admin dapat mengelola varian mobil.

Penjelasan tentang *Use Case Diagram User* pada Gambar 3.3 :

- a. Melihat Halaman Beranda : Memungkinkan *user* melihat halaman beranda.
- b. Memilih Tipe Mobil : Memungkinkan *user* memilih tipe mobil.
- c. Memilih Mobil : Memungkinkan *user* memilih mobil.
- d. Melihat *Detail* Mobil : Memungkinkan *user* melihat *detail* mobil.
- e. Melakukan Pemesanan : Memungkinkan *user* melakukan pemesanan.
- f. Mengirim *Feedback* : Memungkinkan *user* mengirim *feedback*.
- g. Melihat Tentang Penjual : Memungkinkan *user* melihat tentang penjual.

3.6.2 Activity Diagram

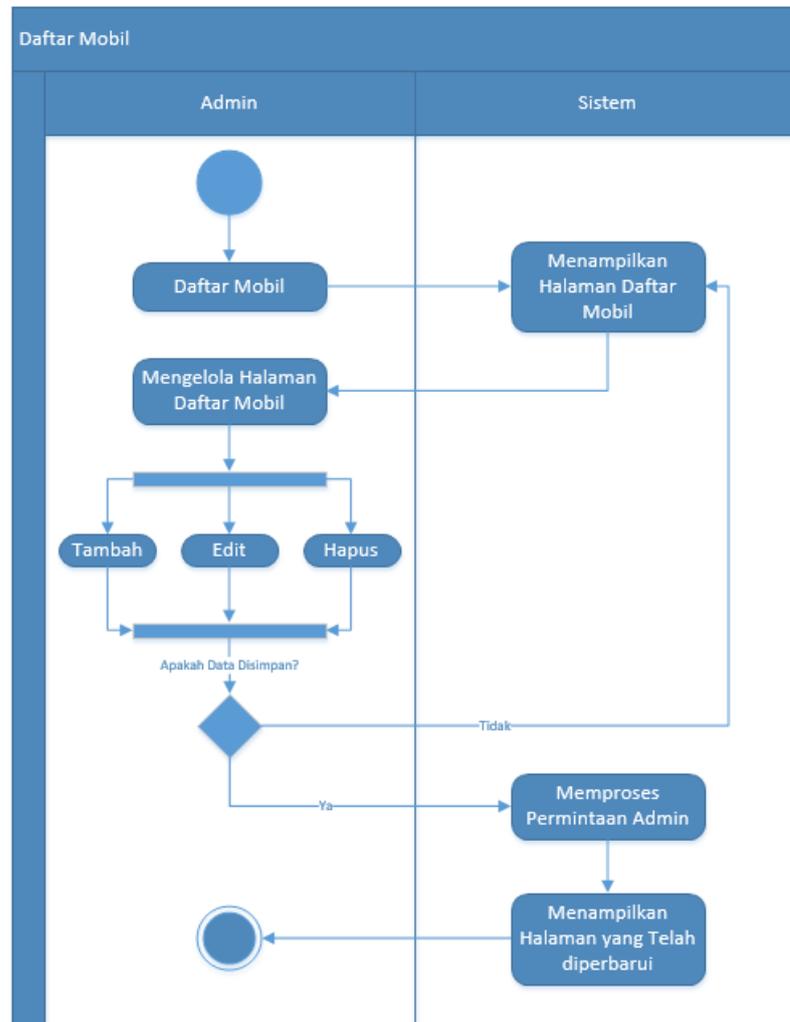
Activity Diagram menjelaskan aktivitas - aktivitas yang dilakukan secara sistematis, yang sedang direncanakan dan bagaimana masing-masing aliran berawal, keputusan yang mungkin terjadi, dan bagaimana aktivitas tersebut berakhir.

a. *Activity diagram Login*

Gambar 3.4 Activity Diagram Login

Gambar 3.4 menunjukkan Activity Diagram untuk *login*. Aktor utama dalam *activity* diagram ini adalah *administrator* sistem yang selanjutnya akan disebut sebagai *admin*. Setelah *admin* mengakses *website* dan memilih halaman *login*, maka sistem akan menampilkan halaman *login*, kemudian *admin* memasukkan data *login* dan submit data. Setelah itu sistem akan melakukan validasi data. Jika data yang dimasukkan *admin* *valid* maka sistem akan menampilkan halaman *dashboard*, dan *admin* dapat mengakses halaman *admin*. Jika data *login* tidak *valid* maka sistem akan kembali menampilkan halaman *login*.

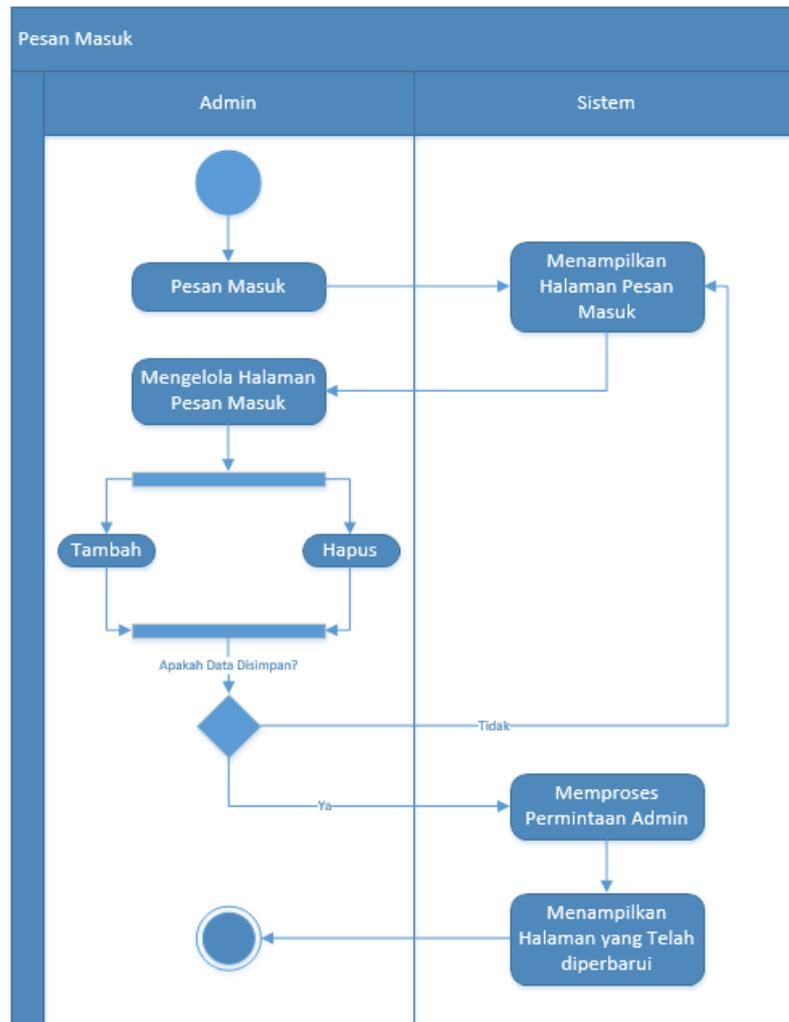
b. *Activity Diagram* Daftar Mobil



Gambar 3.5 *Activity Diagram* Daftar Mobil

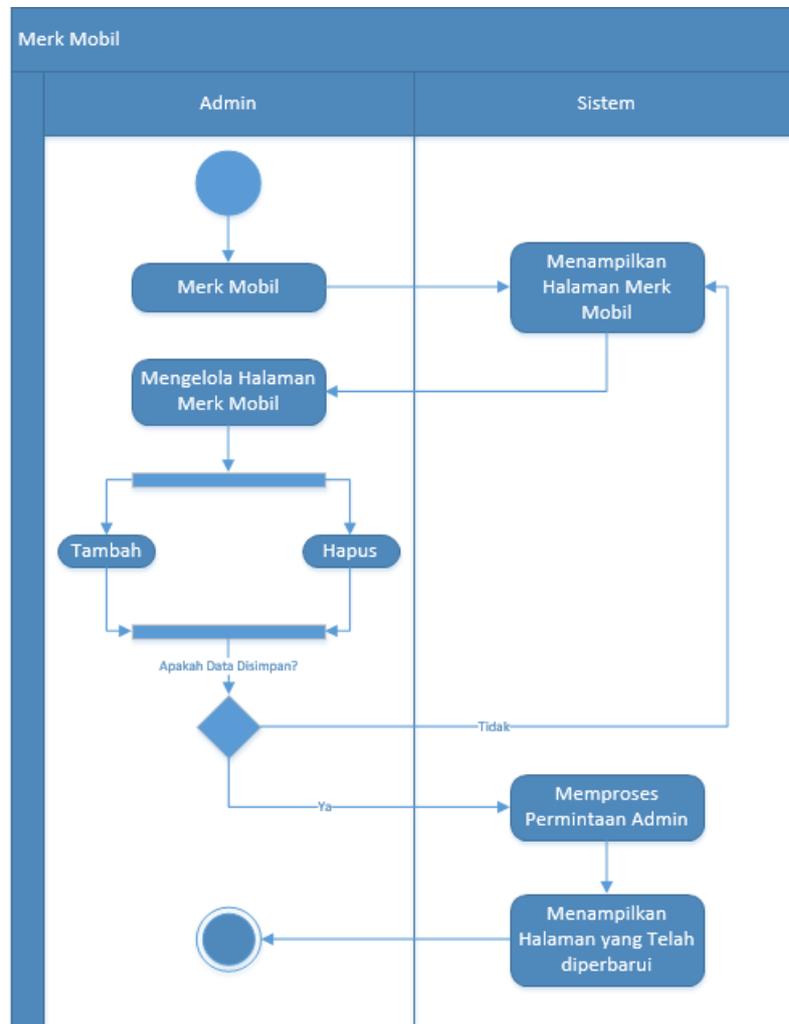
Gambar 3.5 menunjukkan *activity diagram* untuk daftar mobil. Aktor utama dalam *activity diagram* ini adalah *administrator* sistem yang selanjutnya akan disebut sebagai admin. Setelah admin *login* dan masuk ke halaman daftar mobil, admin dapat melihat halaman data mobil dan admin juga dapat mengelola data mobil seperti tambah data, *edit* data dan menghapus data mobil sesuai dengan kebutuhan admin. Jika proses pengelolaan data berlanjut maka akan ada 2 pilihan yaitu membatalkan proses dan sistem mengembalikan ke halaman daftar mobil atau menyimpan data dilanjutkan sistem menampilkan halaman daftar mobil yang telah diperbarui.

c. *Activity Diagram* Pesan Masuk



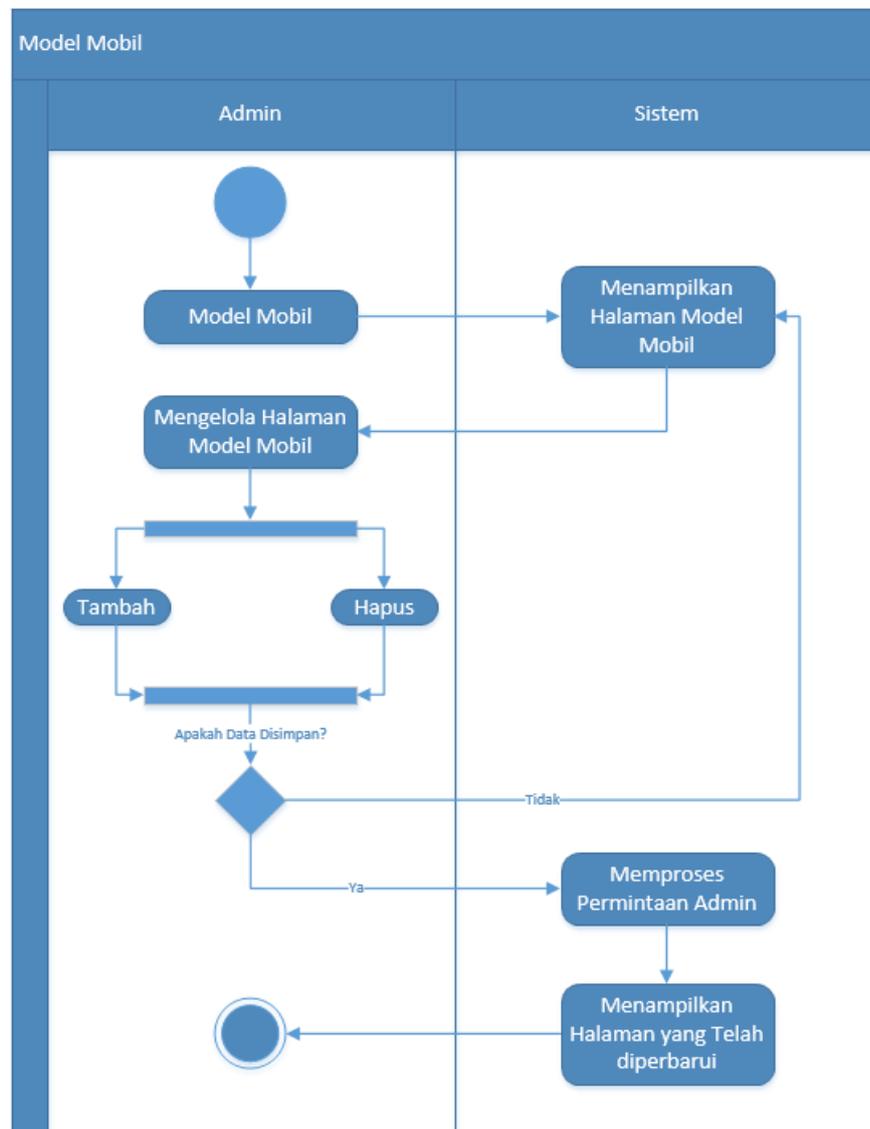
Gambar 3.6 *Activity Diagram* Pesan Masuk

Gambar 3.6 menunjukkan *activity diagram* untuk pesan masuk. Aktor utama dalam *activity diagram* ini adalah *administrator* sistem yang selanjutnya akan disebut sebagai admin. Setelah admin *login* dan masuk ke halaman pesan masuk, admin dapat melihat halaman pesan masuk dan admin juga dapat mengelola pesan masuk seperti tambah data, dan menghapus pesan masuk sesuai dengan kebutuhan admin. Jika proses pengelolaan data berlanjut maka akan ada 2 pilihan yaitu membatalkan proses dan sistem mengembalikan ke halaman pesan masuk atau menyimpan data dilanjutkan sistem menampilkan halaman pesan masuk yang telah diperbarui.

d. *Activity Diagram Merk Mobil*

Gambar 3.7 *Activity Diagram Merk Mobil*

Gambar 3.7 menunjukkan *activity diagram* untuk *merk mobil*. Aktor utama dalam *activity diagram* ini adalah *administrator* sistem yang selanjutnya akan disebut sebagai *admin*. Setelah *admin login* dan masuk ke halaman *merk mobil*, *admin* dapat melihat halaman pesan masuk dan *admin* juga dapat mengelola *merk mobil* seperti *tambah data*, dan *menghapus pesan masuk* sesuai dengan kebutuhan *admin*. Jika proses pengelolaan data berlanjut maka akan ada 2 pilihan yaitu *membatalkan proses* dan *sistem mengembalikan ke halaman merk mobil* atau *menyimpan data* dilanjutkan *sistem menampilkan halaman merk mobil yang telah diperbarui*.

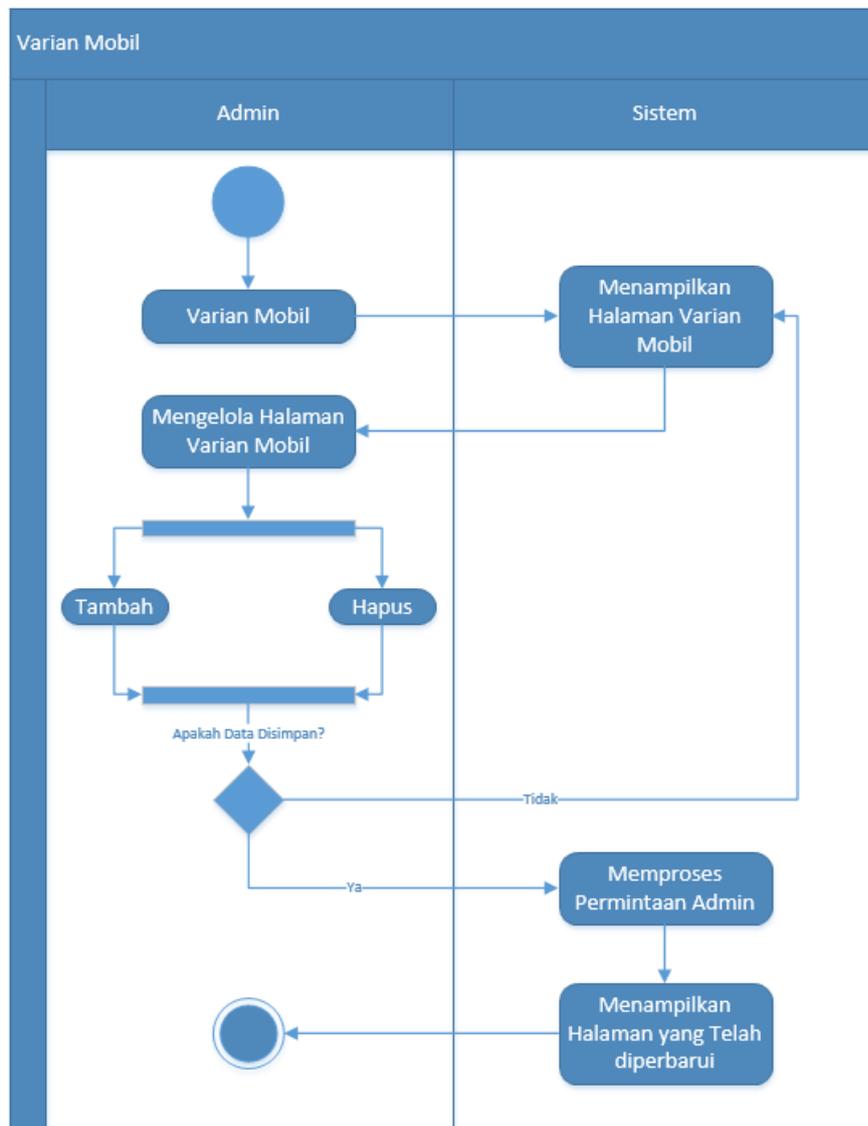
e. *Activity Diagram Model Mobil*

Gambar 3.8 *Activity Diagram Model Mobil*

Gambar 3.8 menunjukkan *activity diagram* untuk *model mobil*. Aktor utama dalam *activity diagram* ini adalah *administrator* sistem yang selanjutnya akan disebut sebagai admin. Setelah admin *login* dan masuk ke halaman pesan masuk, admin dapat melihat halaman *model mobil* dan admin juga dapat mengelola *model mobil* seperti tambah data, dan menghapus pesan masuk sesuai dengan kebutuhan admin. Jika proses pengelolaan data berlanjut maka akan ada 2 pilihan yaitu membatalkan proses dan sistem

mengembalikan ke halaman *model* mobil atau menyimpan data dilanjutkan sistem menampilkan halaman *model* mobil yang telah diperbarui.

f. *Activity Diagram* Varian Mobil



Gambar 3.9 *Activity Diagram* Varian Mobil

Gambar 3.9 menunjukkan *activity diagram* untuk varian mobil. Aktor utama dalam *activity diagram* ini adalah *administrator* sistem yang selanjutnya akan disebut sebagai admin. Setelah admin *login* dan masuk ke halaman varian mobil, admin dapat melihat halaman varian mobil dan admin juga dapat mengelola varian mobil seperti tambah data, dan menghapus pesan

yang nantinya akan menjadi rancangan tabel pada *database Showroom Lampung Jaya Motor*. Hubungan antar *entity* adalah :

1. *Entitas Relationship* Diagram aplikasi *Showroom Lampung Jaya Motor* memiliki 3 buah entitas yaitu:
 - a. Admin
 - b. User
 - c. Mobil
2. Entitas admin merupakan entitas yang kuat karena entitas admin tidak bergantung pada entitas lain.
3. Entitas user dengan entitas mobil menggunakan relasi *one-to-many*.

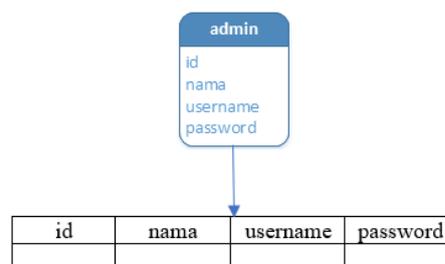
3.6.4 Konversi ERD ke Bentuk Tabel

Konversi ERD ke bentuk tabel adalah proses memasukkan entitas menjadi nama tabel dan attributes menjadi nama kolom beserta relasinya ke dalam setiap tabel yang saling berhubungan.

1. Konversi Entitas Kuat

Konversi entitas kuat (*strong entity*) adalah entitas yang tidak memiliki ketergantungan terhadap entitas lain dan entitas ini tercantum jelas dalam ERD. Pada Gambar 3.10 terdapat entitas kuat yaitu entitas admin dan entitas pesan.

Entitas Kuat Admin



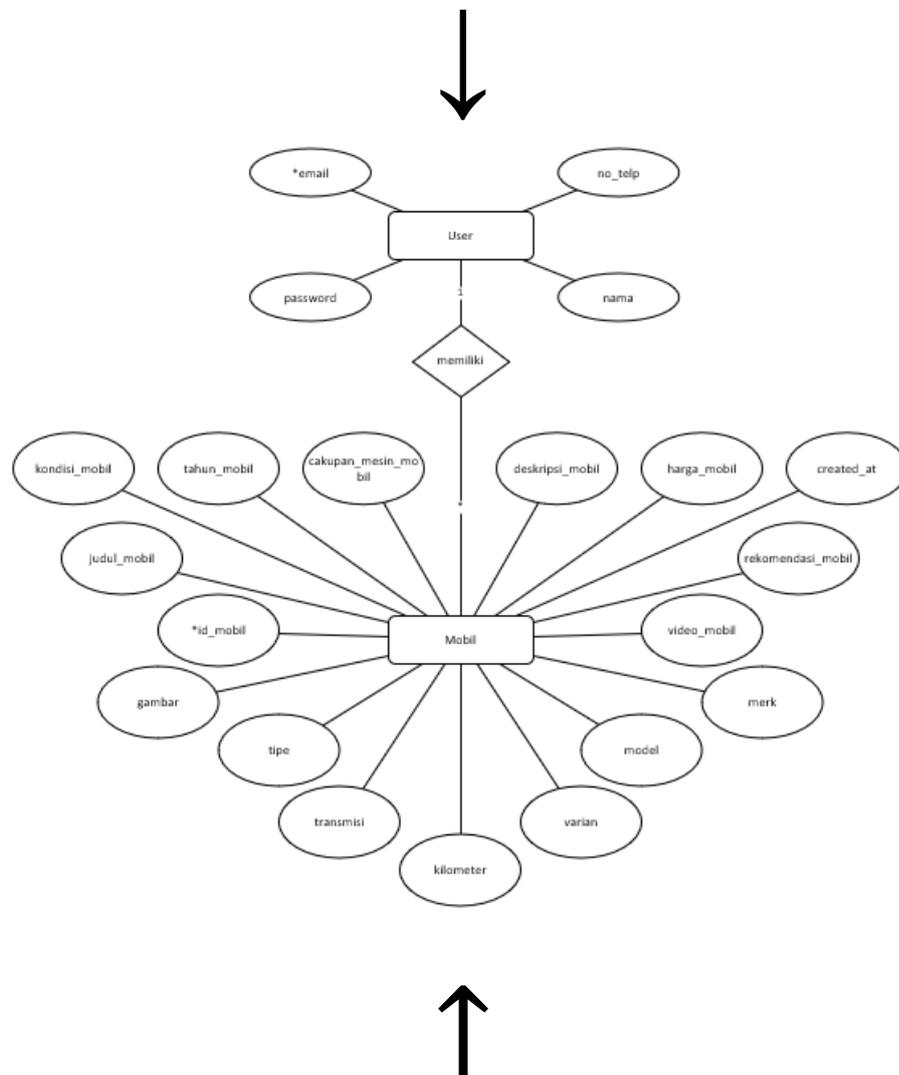
Gambar 3.11 Entitas Kuat Admin

2. Konversi Entitas yang Memiliki Relasi

Konversi entitas yang memiliki relasi *one to many* (1-M) adalah atribut kunci pada entitas yang memiliki kardinal satu (*one*), akan menjadi *foreign key* pada entitas yang memiliki kardinal banyak (*many*). Pada ERD yang ditunjukkan pada Gambar 3.9 ditunjukkan terdapat relasi *one to many* yaitu antara entitas user dengan entitas mobil. Dari entitas tersebut akan direpresentasikan kedalam tabel, dapat dilihat pada gambar berikut.

Entitas User dengan Entitas Mobil

Email	no_telp	nama	Password



id_mobil	gambar	tipe	transmisi	kilometer	varian	model	merk	video_mobil	rekomendasi_mobil	created_at	harga_mobil	deskripsi_mobil	cakupan_mesin	tahun_mobil	kondisi_mobil	judul_mobil

Gambar 3.12 Entitas User dengan Entitas Mobil

Entitas Mobil memiliki relasi *many to one* dengan entitas Tipe yang artinya satu tipe dapat memiliki lebih dari satu mobil maka kolom *id* yang ada pada tabel tipe berfungsi sebagai *foreign key* pada tabel mobil.

3. Normalisasi Tabel Mobil

Setelah tabel Mobil dibuat maka tahap selanjutnya tabel tersebut diperiksa validasinya dan diperbaiki (dinormalisasikan). Tujuan dilakukan normalisasi adalah

- menghilangkan rangkap data sehingga berdampak kepada pemborosan memori.
- untuk menghindari anomali (ketidak konsistenan data).

Desain awal tabel mobil

id	judul	gambar	kondisi	id_merk	merk	id_model	model	id_varian	varian	id_tipe	tipe	tahun	cakupan mesin	id transmisi	transmisi	id km	km	deskripsi	harga	created at	rekomendasi	video
1	A	1a,1b	A	1	A	1	A	1	A	1	A	A	A	1	A	1	A	A	A	A	A	A
2	B	2a,2b	B	2	B	2	B	2	B	2	B	B	B	2	B	2	B	B	B	B	B	B

Gambar 3.13 Desain Awal Tabel Mobil

Berikut adalah bentuk tidak normal dari tabel mobil

id	judul	gambar	kondisi	id_merk	merk	id_model	model	id_varian	varian	id_tipe	tipe	tahun	cakupan mesin	id transmisi	transmisi	id km	km	deskripsi	harga	created at	rekomendasi	video
1	A	1a,1b	A	1	A	1	A	1	A	1	A	A	A	1	A	1	A	A	A	A	A	A
2	B	2a,2b	B	2	B	2	B	2	B	2	B	B	B	2	B	2	B	B	B	B	B	B

Gambar 3.14 Tabel Mobil Tidak Normal

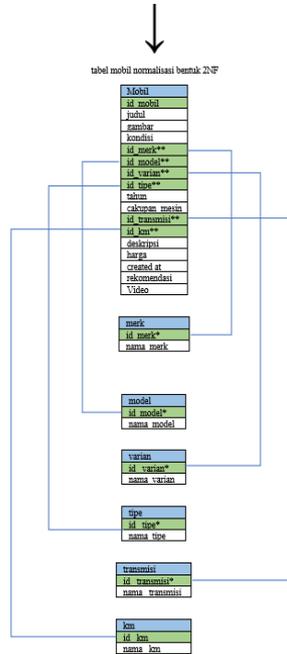
Pada gambar 3.14 kolom gambar ditunjukkan masih mempunyai data ganda maka perlu dinormalisasikan menjadi 1NF ditunjukkan pada Gambar 3.15.

id	judul	gambar	kondisi	id_merk	merk	id_model	model	id_varian	varian	id_tipe	tipe	tahun	cakupan mesin	id transmisi	transmisi	id km	km	deskripsi	harga	created at	rekomendasi	video
1	A	1a	A	1	A	1	A	1	A	1	A	A	A	1	A	1	A	A	A	A	A	A
2	A	1b	B	2	B	2	B	2	B	2	B	B	B	2	B	2	B	B	B	B	B	B
3	B	2a	A	1	A	1	A	1	A	1	A	A	A	1	A		A	A	A	A	A	A
4	B	2b	B	2	B	2	B	2	B	2	B	B	B	2	B		B	B	B	B	B	B

Gambar 3.15 Tabel Mobil Bentuk 1NF

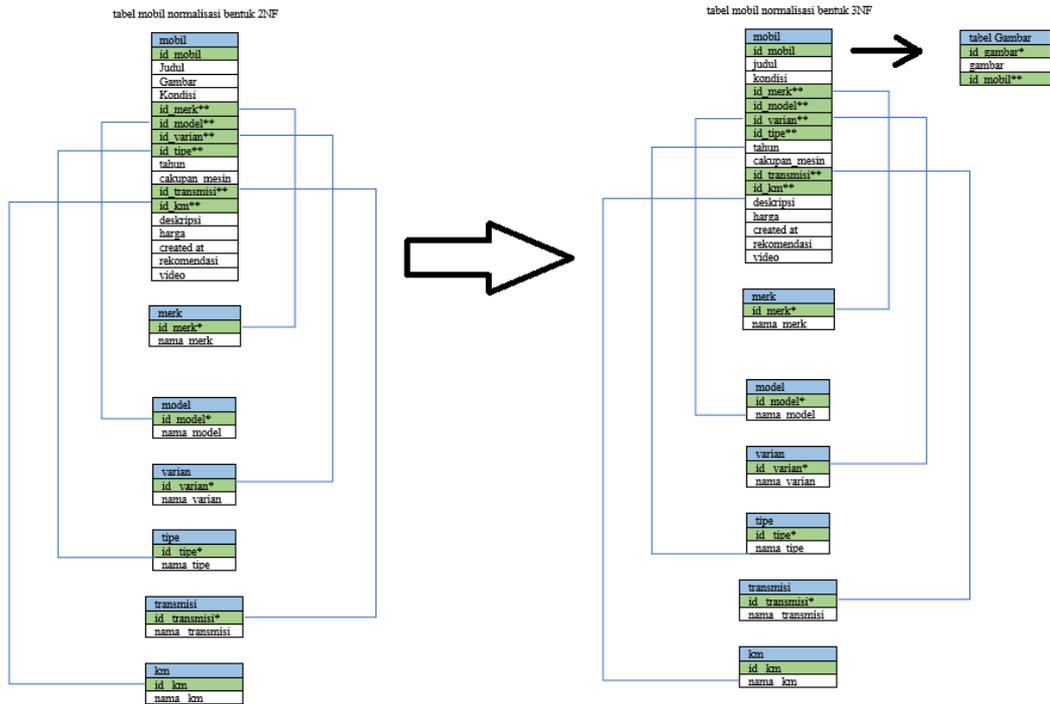
Gambar 3.15 ditunjukkan terdapat ketergantungan fungsional antara kolom id_merk dengan merk, id_model dengan model, id_varian dengan varian, id-tipe dengan tipe, id_transmisi dengan transmisi dan id_km dengan km sehingga dapat tabel mobil data diubah menjadi 2nf, dengan cara memisahkan kolom yang memiliki ketergantungan fungsional dapat dilihat pada Gambar 3.16.

id	judul	gambar	kondisi	id_merk	merk	id_model	model	id_varian	varian	id_tipe	tipe	tahun	cakupan_mesin	id_transmisi	transmisi	id_km	km	deskripsi	harga	created_at	rekomendasi	video	
1	A	1a	A	1	A	1	A	1	A	1	A	A	A	1	A	1	A	A	A	A	A	A	A
2	A	1b	B	2	B	2	B	2	B	2	B	B	B	2	B	2	B	B	B	B	B	B	B
3	B	2a	A	1	A	1	A	1	A	1	A	A	A	1	A	1	A	A	A	A	A	A	A
4	B	2b	B	2	B	2	B	2	B	2	B	B	B	2	B	2	B	B	B	B	B	B	B



Gambar 3.16 Tabel Mobil bentuk 2NF

Gambar 3.15 ditunjukkan masih terdapat ketergantungan transitif antara kolom id_mobil, judul dengan gambar, dan terdapat anomaly pada kolom gambar mempunyai 4 data yang berbeda sedangkan judul hanya ada 2 data yang berbeda, maka perlu dilakukan normalisasi 3NF. ditunjukkan pada Gambar 3.17.

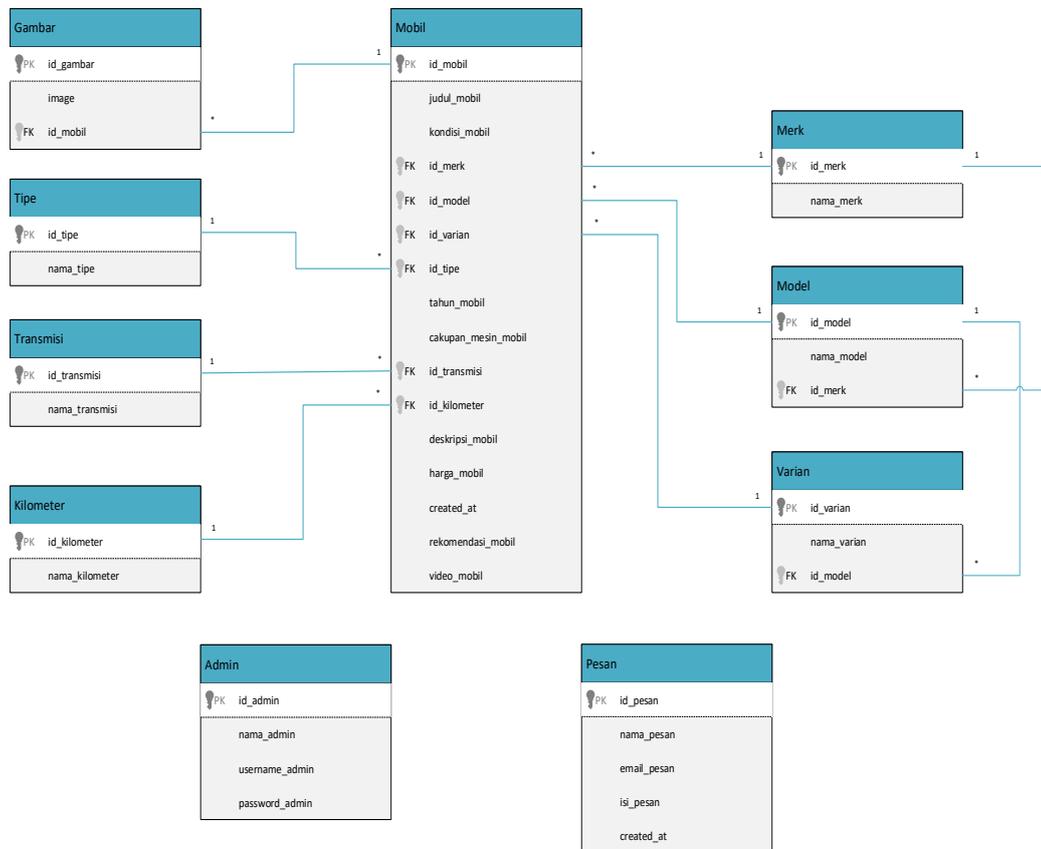


Gambar 3.17 Tabel Mobil bentuk 3NF

Gambar 3.17 ditunjukkan setelah melakukan proses normalisasi maka terbentuk beberapa tabel, tabel - tabel tersebut berupa tabel merk, tabel model, tabel varian, tabel tipe, tabel transmisi, tabel km, dan tabel gambar.

3.6.5 Relasi Antar Tabel

Setelah dilakukan konversi ERD ke bentuk tabel maka akan didapatkan relasi antar tabel yang dapat dilihat pada gambar 3.20.



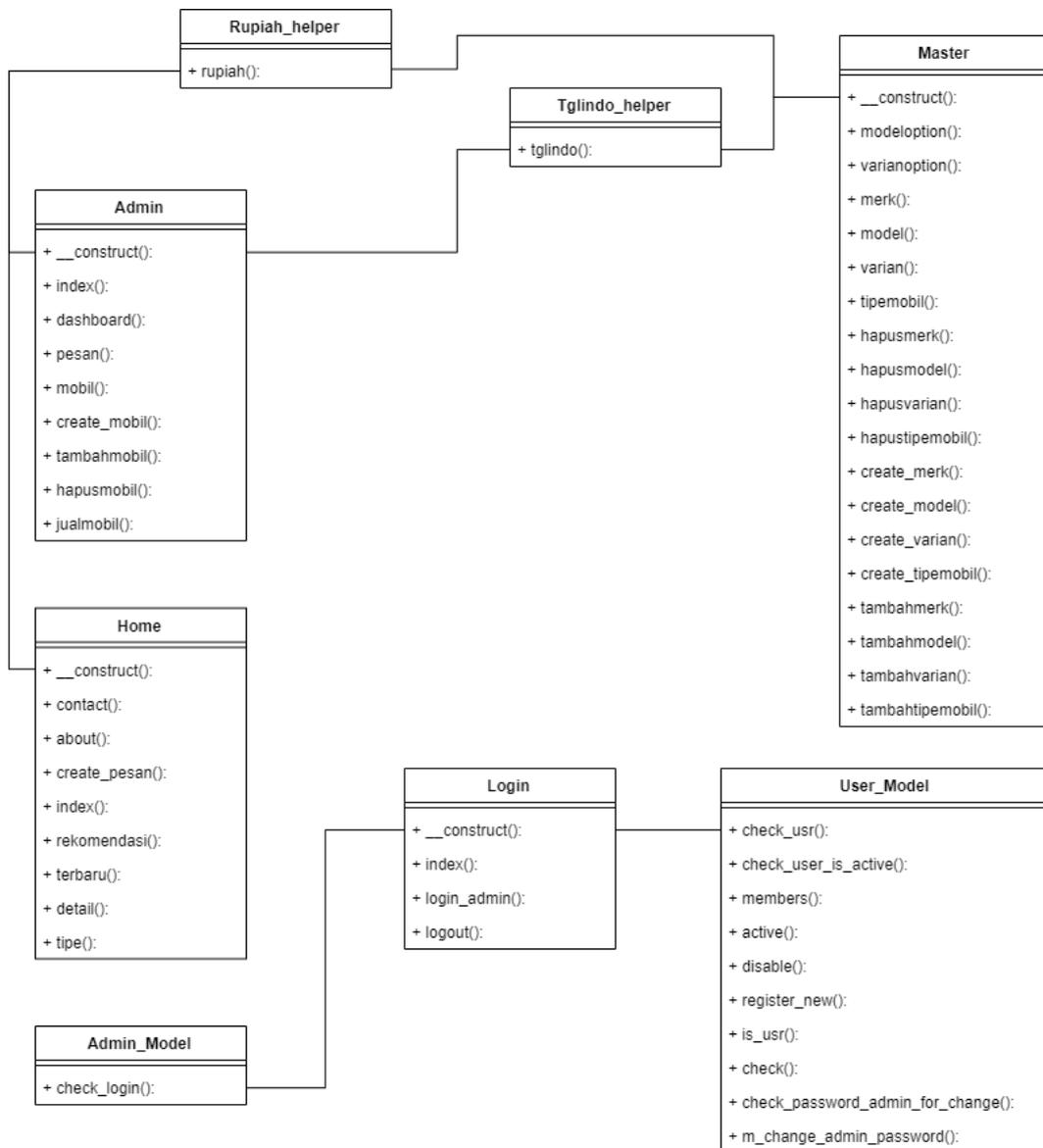
Gambar 3.13 Relasi Antar Tabel

Penjelasan relasi antar tabel yang telah digambarkan seperti pada gambar 3.20 adalah :

- Tabel mobil memiliki relasi *one to many* dengan tabel gambar untuk menghubungkan tabel mobil dengan tabel gambar maka kolom `id_mobil` harus ditambahkan sebagai *foreign key* pada tabel gambar.
- Tabel merk memiliki relasi *one to many* dengan tabel model untuk menghubungkan tabel merk dengan tabel model maka kolom `id_merk` harus ditambahkan sebagai *foreign key* pada tabel model.
- Tabel model memiliki relasi *one to many* dengan tabel varian untuk menghubungkan tabel model dengan tabel varian maka kolom `id_model` harus ditambahkan sebagai *foreign key* pada tabel varian.
- Tabel mobil memiliki relasi *many to one* dengan tabel tipe untuk menghubungkan tabel mobil dengan tabel tipe maka kolom `id_tipe` harus ditambahkan sebagai *foreign key* pada tabel mobil.

- e. Tabel mobil memiliki relasi *many to one* dengan tabel transmisi untuk menghubungkan tabel mobil dengan tabel transmisi maka kolom *id_transmisi* harus ditambahkan sebagai *foreign key* pada tabel mobil.
- f. Tabel mobil memiliki relasi *many to one* dengan tabel kilometer untuk menghubungkan tabel mobil dengan tabel kilometer maka kolom *id_kilometer* harus ditambahkan sebagai *foreign key* pada tabel mobil.
- g. Tabel mobil memiliki relasi *many to one* dengan tabel merk untuk menghubungkan tabel mobil dengan tabel merk maka kolom *id_merk* harus ditambahkan sebagai *foreign key* pada tabel mobil.
- h. Tabel mobil memiliki relasi *many to one* dengan tabel model untuk menghubungkan tabel mobil dengan tabel model maka kolom *id_model* harus ditambahkan sebagai *foreign key* pada tabel mobil.
- i. Tabel mobil memiliki relasi *many to one* dengan tabel varian untuk menghubungkan tabel mobil dengan tabel varian maka kolom *id_varian* harus ditambahkan sebagai *foreign key* pada tabel mobil.

3.6.6 Class Diagram



Gambar 3.14 Class Diagram

Gambar 3.21 adalah *class* diagram yang menunjukkan *class*, *field*, *method* dan relasinya. Relasi antar *class* adalah :

- Class admin* dengan *class rupiah_helper* memiliki relasi antar kelas bermakna umum (*association*), artinya kelas *rupiah_helper* dijalankan untuk dapat menjalankan *class admin*.

- b. *Class admin* dengan *class tglindo_helper* memiliki relasi antar kelas bermakna umum (*association*), artinya kelas *tglindo_helper* dijalankan untuk dapat menjalankan *class admin*.
- c. *Class home* dengan *class rupiah_helper* memiliki relasi antar kelas bermakna umum (*association*), artinya kelas *rupiah_helper* dijalankan untuk dapat menjalankan *class home*.
- d. *Class master* dengan *class rupiah_helper* memiliki relasi antar kelas bermakna umum (*association*), artinya kelas *rupiah_helper* dijalankan untuk dapat menjalankan *class master*.
- e. *Class master* dengan *class tglindo_helper* memiliki relasi antar kelas bermakna umum (*association*), artinya kelas *tglindo_helper* dijalankan untuk dapat menjalankan *class master*.
- f. *Class login* dengan *class admin_model* memiliki relasi antar kelas bermakna umum (*association*), artinya kelas *admin_model* dijalankan untuk dapat menjalankan *class login*.
- g. *Class login* dengan *class user_model* memiliki relasi antar kelas bermakna umum (*association*), artinya kelas *user_model* dijalankan untuk dapat menjalankan *class login*.

3.6.7 Kamus Database

Berikut ini merupakan tabel - tabel dari kamus *database* pada aplikasi “Sistem Informasi *Showroom* Lampung Jaya Motor Berbasis Web”.

1. Kamus Data Admin

Tabel 3. 1 Kamus Data Tabel Admin

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id admin
Nama	Varchar	30		Nama Admin
<i>Username</i>	Varchar	16		<i>Username</i> Admin

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Password	varchar	255		Password Admin

2. Kamus Data Gambar

Tabel 3. 2 Kamus Data Tabel Gambar

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id Gambar
Image	Varchar	100		Image Mobil
Mobil	Int	11	Foreign Key	Id Mobil

3. Kamus Data Kilometer

Tabel 3. 3 Kamus Data Tabel Kilometer

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id Kilometer
Nama	Varchar	30		Nama Kilometer

4. Kamus Data *Merk*

Tabel 3. 4 Kamus Data Tabel *Merk*

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id <i>Merk</i>
Nama	Varchar	30		Nama <i>Merk</i>

5. Kamus Data Mobil

Tabel 3. 5 Kamus Data Tabel Mobil

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id Mobil
Judul	Varchar	30		Judul Mobil

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Kondisi	Enum			Kondisi Mobil
<i>Merk</i>	Int	11	Foreign Key	Id <i>Merk</i> Mobil
<i>Model</i>	Int	11	Foreign Key	Id <i>Model</i> Mobil
Varian	Int	11	Foreign Key	Id Varian Mobil
Tipe	Int	11	Foreign Key	Id Tipe Mobil
Tahun	Char	4		Tahun Mobil
Cakupan_mesin	Char	5		Cakupan Mesin Mobil
Transmisi	Int	11	Foreign Key	Id Transmisi Mobil
Km	Int	11	Foreign Key	Id Kilometer Mobil
Deskripsi	Text			Deskripsi Mobil
Harga	Int	11		Harga Mobil
Created at	Timestamp			Waktu Pembuatan
Rekomendasi	Tinyint	1		Rekomendasi Mobil
Video	Text			Url Video Mobil

6. Kamus Data Model

Tabel 3. 6 Kamus Data Tabel Model

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id Model
Nama	Varchar	30		Nama Model
<i>Merk</i>	Int	11	Foreign Key	Id <i>Merk</i> Mobil

7. Kamus Data Pesan

Tabel 3. 7 Kamus Data Tabel Pesan

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id Pesan
Nama	Varchar	30		Nama Pengirim Pesan
<i>Email</i>	Varchar	30		<i>Email</i> Pengirim Pesan
Pesan	Text			Isi Pesan
Created at	Timestamp			Waktu Pembuatan

8. Kamus Data Tipe

Tabel 3. 8 Kamus Data Tabel Tipe

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id Tipe
Nama	Varchar	30		Nama Tipe

9. Kamus Data Transmisi

Tabel 3. 9 Kamus Data Tabel Transmisi

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id Transmisi
Nama	Varchar	30		Nama Transmisi

10. Kamus Data Varian

Tabel 3. 10 Kamus Data Tabel Varian

Nama Kolom	Tipe Data	Panjang Data	Key	Keterangan
Id	Int	11	Primary Key	Id <i>Model</i>
Nama	Varchar	30		Nama <i>Model</i>
<i>Merk</i>	Int	11	Foreign Key	Id <i>Merk</i> Mobil

3.6.8 Rancangan *User Interface*

User interface merupakan bagian yang menjembatani antara *user* dan sistem. Oleh karena itu, *user interface* merupakan bagian yang penting dalam pembuatan suatu aplikasi. Desain *user interface* yang baik dalam pembuatan aplikasi akan memudahkan pengguna dalam menggunakan aplikasi.

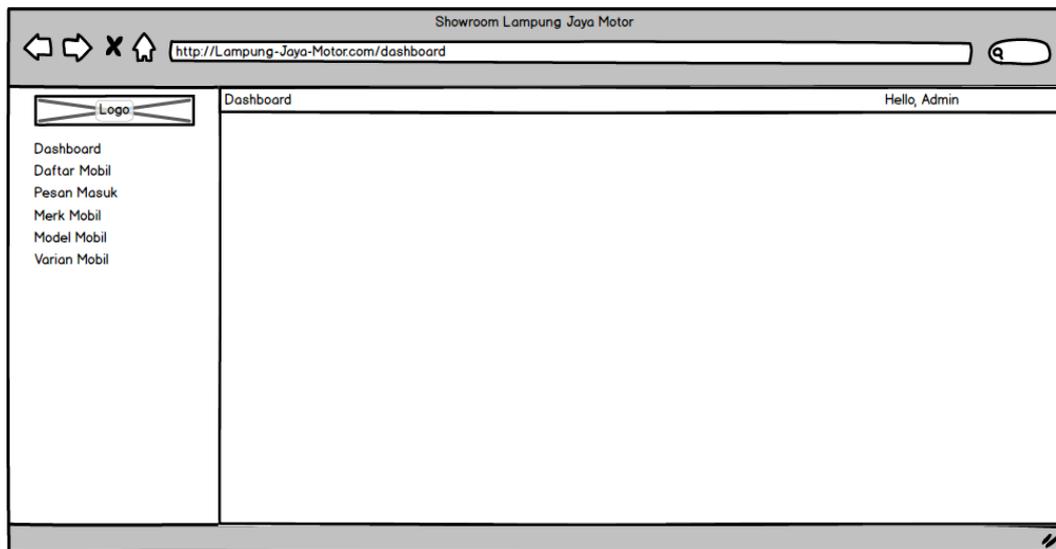
1. Desain *User Interface* Untuk Admin

The image shows a web browser window with the title 'Showroom Lampung Jaya Motor'. The address bar contains 'http://Lampung-Jaya-Motor.com/dashboard'. The main content area displays a 'Login Admin' form with the following elements:

- Username:
- Password:
- Login:
- Bukan admin? [kembali ke halaman beranda](#)

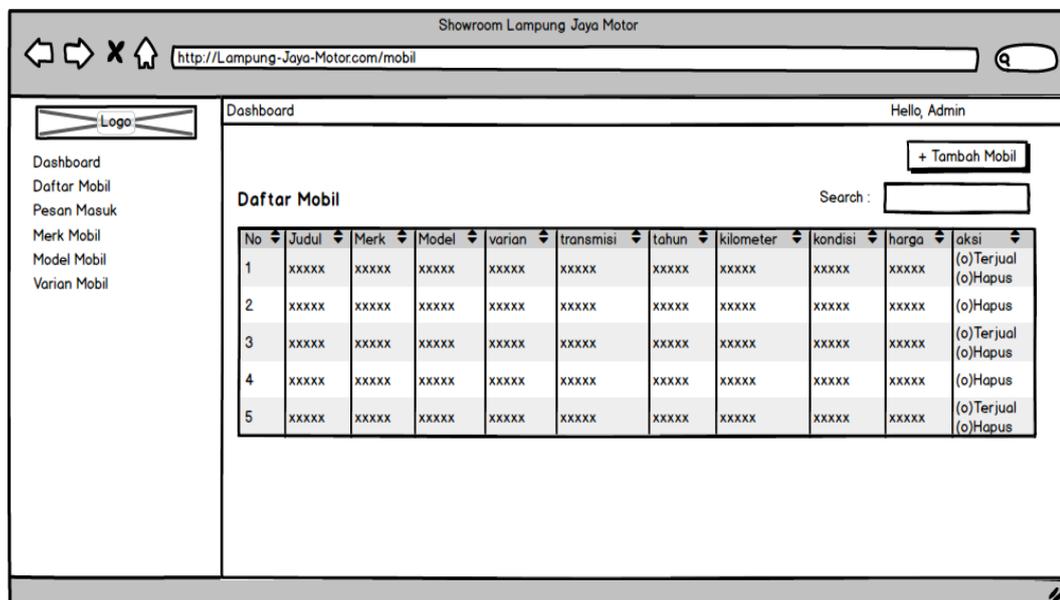
Gambar 3.15 Desain Halaman Login

Gambar 3.22 adalah desain halaman *login* yang berisi *field* bagi admin untuk memasukkan *username* dan *password* admin agar dapat masuk ke halaman *dashboard*, atau ke halaman beranda bagi yang bukan admin.



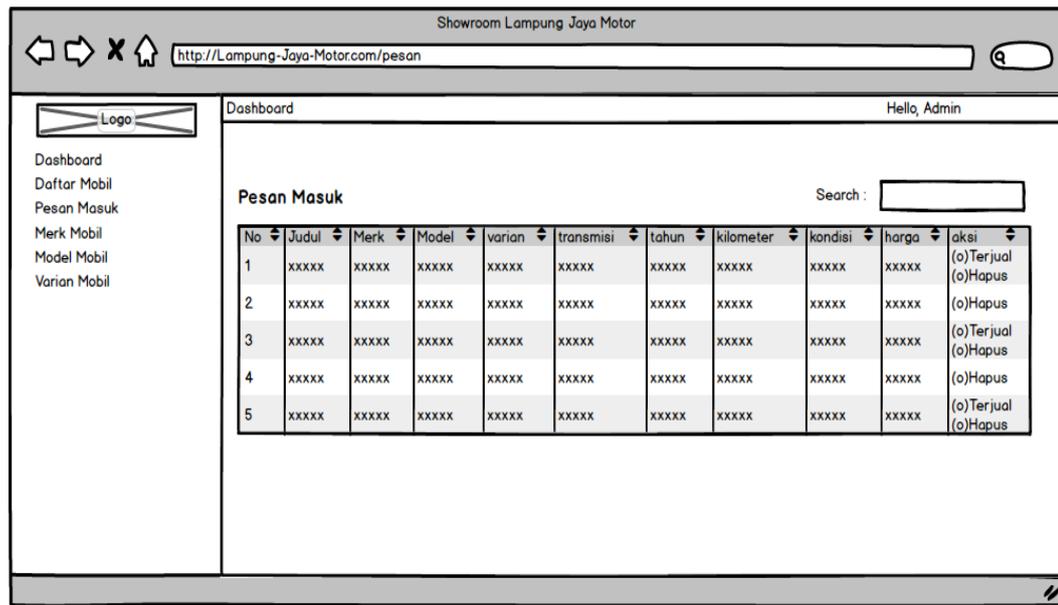
Gambar 3.16 Desain Halaman *Dashboard*

Gambar 3.23 adalah desain halaman *dashboard* yang di tampilkan oleh sistem sebagai halaman awal admin setelah melakukan login.



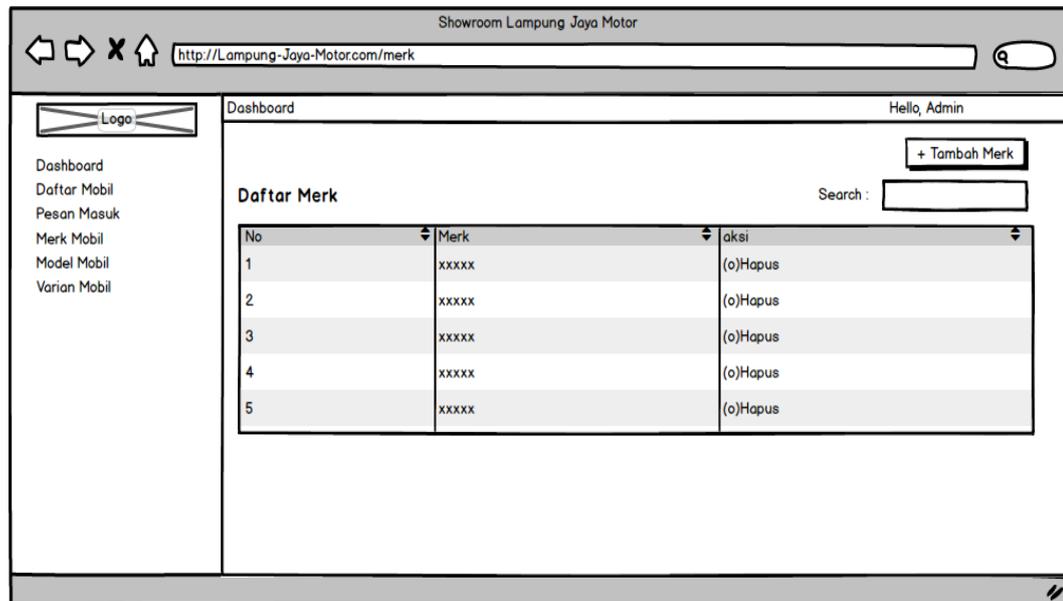
Gambar 3.17 Desain Halaman Daftar Mobil

Gambar 3.24 adalah desain halaman daftar mobil yang berisi *table* daftar mobil dan admin dapat mengelola daftar mobil dimana admin bisa menambahkan mobil baru, menghapus dan *update* mobil yang sudah terjual.



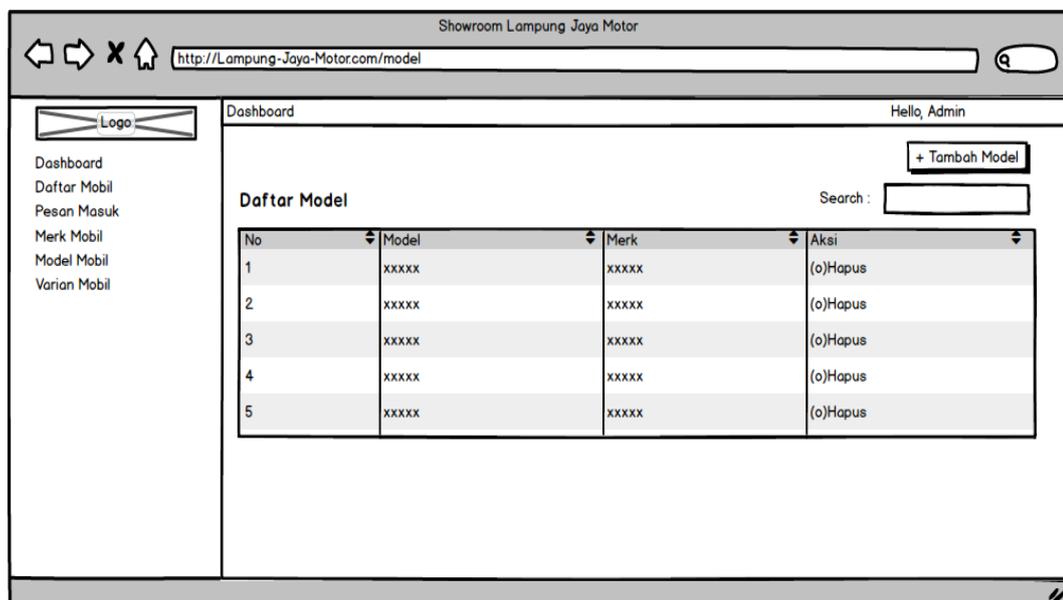
Gambar 3.18 Desain Halaman Pesan Masuk

Gambar 3.25 adalah desain halaman daftar pesan masuk yang berisi *button* search untuk mencari pesan dan *table* untuk menampilkan daftar pesan yang dikirimkan oleh *user*.



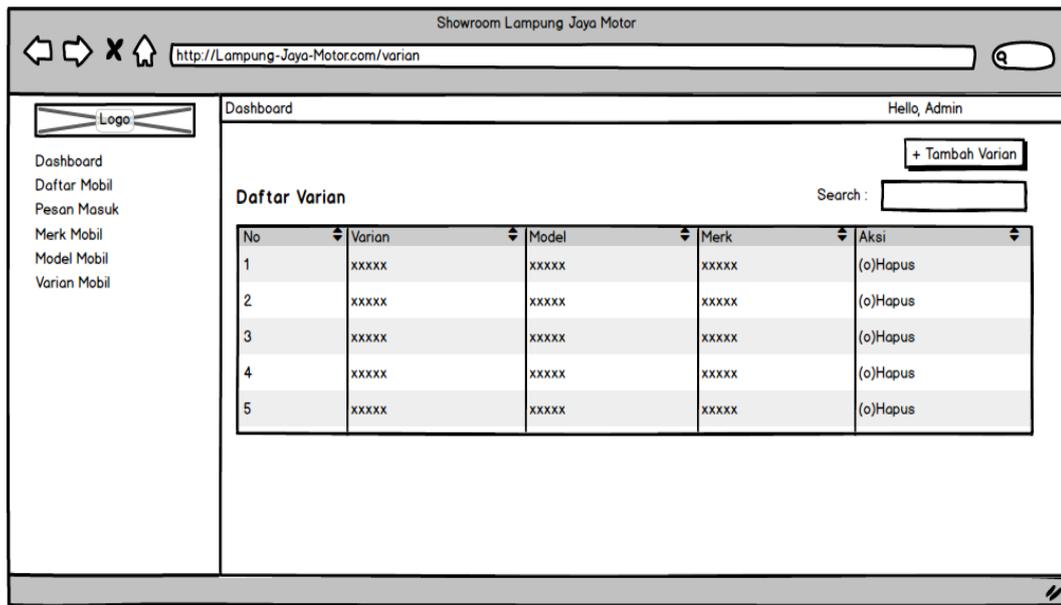
Gambar 3.19 Desain Halaman Daftar *Merk*

Gambar 3.26 adalah desain halaman daftar *merk* yang berisi *table* daftar *merk* mobil dan admin dapat mengelola daftar *merk* mobil dimana admin bisa menambahkan *merk* mobil baru dan menghapus *merk* mobil.



Gambar 3.20 Desain Halaman Daftar *Model*

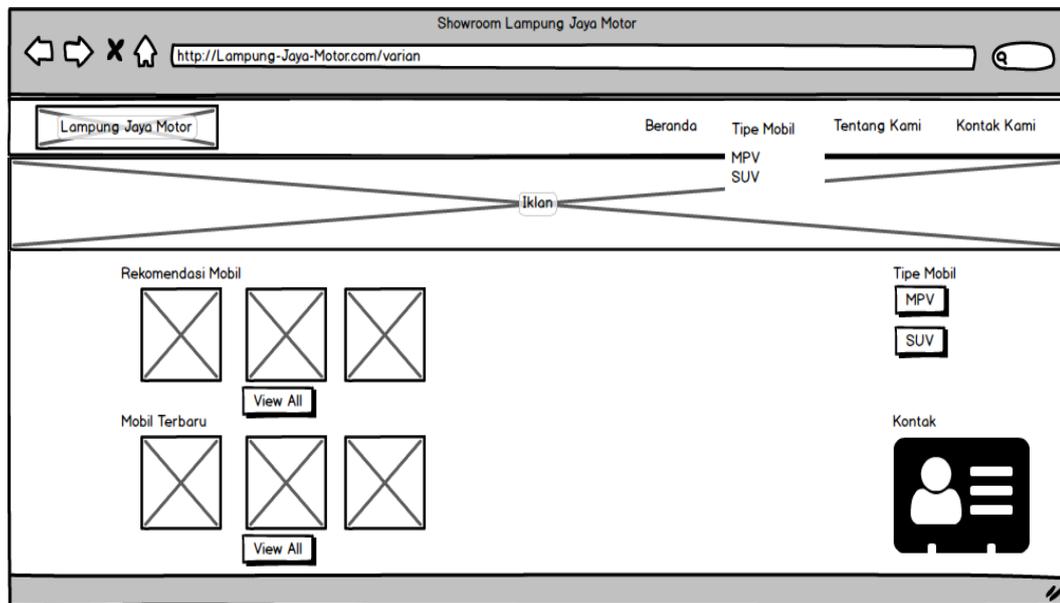
Gambar 3.27 adalah desain halaman daftar *model* yang berisi *table* daftar *model* mobil dan admin dapat mengelola daftar *model* mobil dimana admin bisa menambahkan *model* mobil baru dan menghapus *model* mobil.



Gambar 3.21 Desain Halaman Daftar Varian

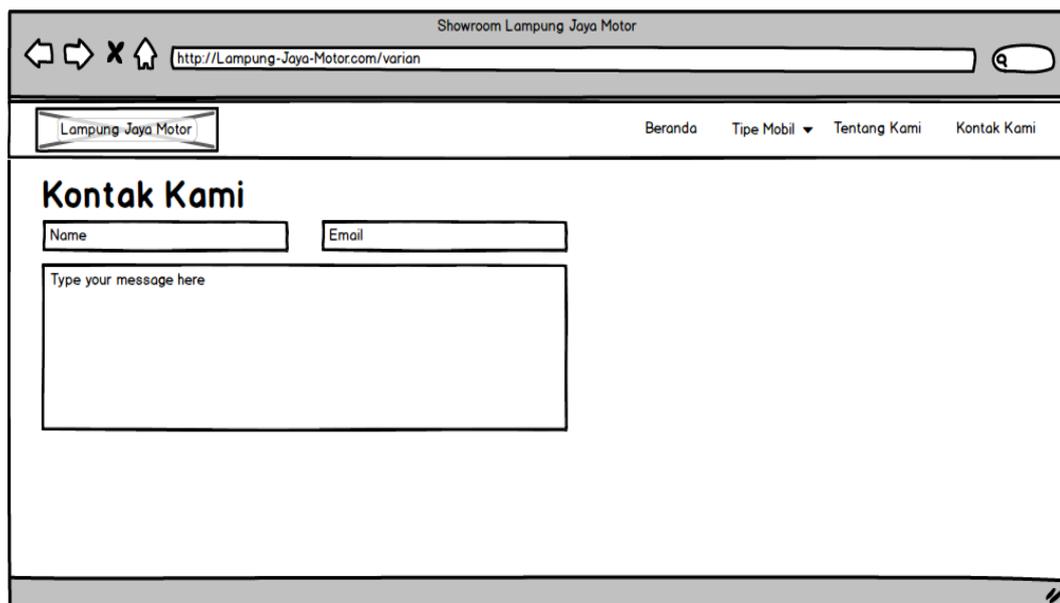
Gambar 3.28 adalah desain halaman daftar varian yang berisi *table* daftar varian mobil dan admin dapat mengelola daftar varian mobil dimana admin bisa menambahkan varian mobil baru dan menghapus varian mobil.

2. Desain *User interface* untuk *User*



Gambar 3.22 Desain Halaman Beranda

Gambar 3.29 adalah desain halaman beranda yang menjadi halaman awal bagi *user*, dimana halaman ini menampilkan rekomendasi mobil, *list* mobil terbaru, list tipe mobil, kontak admin dan informasi perusahaan.



Gambar 3.23 Desain Halaman Kontak Kami

Gambar 3.30 desain halaman kontak kami yang berisi *field* untuk *user* mengirimkan pesan yang berupa permintaan mobil kepada admin. *User* diminta mengirimkan nama dan *email* juga agar admin dapat menghubungi *user* kembali menggunakan *email*.