

BAB III

METODOLOGI PENELITIAN

3.1 Metodologi Analisis

Analisis kebutuhan bertujuan untuk menentukan kebutuhan yang diperlukan pada sebuah aplikasi. Analisis kebutuhan juga berfungsi sebagai bagaimana cara kerja aplikasi yang dibangun dan tampilan antarmuka yang sesuai dengan aplikasi tersebut. Metode yang digunakan dalam membangun aplikasi ini adalah metode pengumpulan data, perancangan model bisnis, analisis pengguna dan fungsionalitas, kebutuhan *input* dan *output*, analisis antarmuka dan perancangan perangkat lunak.

3.2 Analisis Kebutuhan

Hasil analisis kebutuhan yang dibutuhkan dalam pembuatan aplikasi ini adalah sebagai berikut.

3.2.1 Analisis Pengguna dan Fungsionalitas

Dalam aplikasi ini akan terdapat tiga pengguna yang dalam prakteknya. Pengguna pertama merupakan dapat melihat lokasi, yaitu pengguna Android yang dapat melihat lokasi bengkel yang sudah terdapat dalam lokasi tersebut, kemudian dapat melakukan pemesanan dan melihat posisi pengguna pada saat itu, dimana untuk melakukan hal tersebut pengguna harus daftar dulu dan kemudian *login*. Pengguna kedua merupakan administrator bengkel, yaitu pengguna yang dapat melihat pesanan yang masuk pada bengkel tertentu serta memanipulasinya. Kemudian pengguna ketiga, yaitu pengguna administrator utama yang dapat menghapus, menandai, menambahkan dan bengkel baru, serta melihat data dari bengkel dan pengguna Android.

Pada aplikasi ini akan terdapat perbedaan cara mengakses bagi pengguna. Bagi pengguna Android, media yang digunakan dalam mengakses aplikasi ini adalah *smartphone* berbasis Android, begitu juga dengan pengguna administrator utama menambahkan bengkel dengan *smartphone* Android, serta memanipulasi data pengguna dan bengkel yang dapat diakses melalui *web browser*. Sedangkan bagi administrator bengkel, melakukan pengelolaan pesan masuk dapat diakses

melalui *web browser* komputer, hal ini disebabkan lebih mudah dalam hal melakukan pengolahan.

3.2.2 Perangkat Lunak yang Dibutuhkan

Perangkat lunak yang dibutuhkan dalam pembuatan aplikasi pencarian lokasi bengkel dan pemesanan adalah sebagai berikut:

a) Android Studio

Android studio merupakan IDE (*Integrated Development Environment*) *official IntelliJ IDEA* untuk membuat aplikasi android.

b) Java Development Kit (JDK)

JDK berfungsi untuk mengkompilasi kode – kode java menjadi aplikasi java.

c) XAMPP Server

XAMPP server adalah server lokal yang didalamnya sudah tersedia basisdata MySQL dan untuk membantu pengembangan website.

d) MySQL

MySQL digunakan untuk membuat basisdata dengan menggunakan server hosting sebagai layanannya agar bisa diakses secara *online*.

e) Sublime Text

Sublime text digunakan untuk media dalam penulisan pemrograman PHP, HTML, dan MySQL.

f) Google Chrome

Google chrome digunakan sebagai web browser menampilkan tampilan dari aplikasi yang dibuat.

g) Windows 7 Home Premium

Windows merupakan sistem operasi yang digunakan untuk membuat aplikasi ini.

3.2.3 Perangkat Keras yang Dibutuhkan

Perangkat keras juga dibutuhkan dalam pembuatan aplikasi pencarian lokasi bengkel dan pemesanan adalah sebagai berikut:

a) Laptop Sony Vaio VPCEG28FG

Laptop digunakan untuk membuat aplikasi serta pengujian aplikasi.

b) Smartphone Sony Xperia M2

Smartphone digunakan untuk melakukan uji coba aplikasi android.

c) Kabel Data

Kabel data digunakan untuk media penghubung *smartphone* dan Android Studio untuk melakukan *debugging*.

3.2.4 Analisis Kebutuhan Masukan (*Input*)

Dalam aplikasi ini terdapat beberapa *input* berdasarkan penggunanya, yaitu pengguna Android yang dapat melakukan pemesanan, kemudian pengguna kedua yaitu administrator bengkel serta pengguna Admin utama yang menandai dan menambahkan, serta mengelolah data bengkel dan pengguna Android. Berikut *input* yang harus diisi oleh berbagai pengguna:

a) *Input* yang diberikan oleh pengguna Android, yaitu melakukan pemesanan melalui *smartphone* Android adalah:

1. Nama, diisi dengan nama untuk pendaftaran.
2. Email, diisi dengan alamat email untuk pendaftaran.
3. No hp, diisikan dengan nomor handphone untuk pendaftaran.
4. Password, diisi password untuk pendaftaran.
5. Email, untuk melakukan *login* setelah mendaftar.
6. Password, untuk melakukan *login* setelah mendaftar.
7. Nama, diisi dengan nama pemesan.
8. No plat motor, diisikan dengan nomor plat motor.
9. No hp, diisikan dengan nomor handphone pemesan.
10. Keluhan, diisikan dengan masalah pada motor yang ingin di servis.

b) *Input* yang diberikan oleh pengguna kedua, yaitu administrator bengkel motor melalui *web browser* untuk *login* adalah:

1. Username, diisi dengan username yang telah didaftarkan.
2. Password, diisi dengan username yang telah didaftarkan.
3. Data edit profil bengkel, berupa nama bengkel, alamat bengkel, *latitude* dan *longitude*.

c) *Input* yang diberikan oleh pengguna admin utama, yaitu pengguna yang menandai, menambahkan, serta mengelolah data bengkel dan pengguna Androi melalui *smartphone* Android dan *web browser* adalah:

1. Username, diisi dengan username bengkel motor yang ingin didaftarkan melalui *smartphone* Android.
2. Password, diisi dengan password bengkel motor yang ingin didaftarkan melalui *smartphone* Android.
3. Nama, diisi dengan nama bengkel yang akan tampil pada map melalui melalui *smartphone* Android.
4. Alamat, diisi dengan alamat bengkel melalui *smartphone* Android.
5. *Latitude*, diisi dengan koordinat *latitude* yang menunjukkan lokasi bengkel melalui *smartphone* Android.
6. *Longitude*, diisi dengan koordinat *longitude* yang menunjukkan lokasi bengkel melalui *smartphone* Android.
7. Username dan password untuk masuk ke halaman utama administrator melalui *web browser*.

3.2.5 Analisis Kebutuhan Proses

Proses yang dapat dilakukan oleh pengguna aplikasi pencarian dan pemesanan bengkel ini adalah:

1. Proses pendaftaran pengguna Android.
2. Proses login pengguna Android.
3. Proses pemesanan.
4. Proses login administrator bengkel.
5. Proses edit administrator bengkel.
6. Proses konfirmasi pemesanan.
7. Proses login administrator utama.
8. Prose manipulasi data bengkel dan pengguna Android.

3.2.6 Analisis Kebutuhan Keluaran (*Output*)

Dalama aplikasi ini akan keluar output berdasarkan pengguna yang mengakses aplikasi, yaitu:

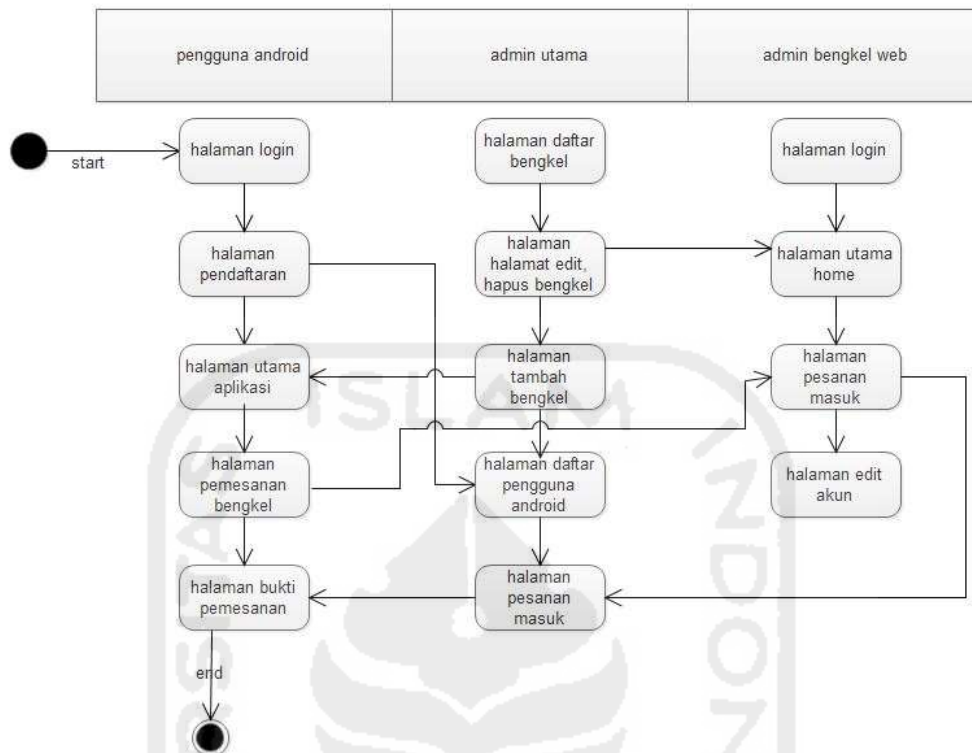
- a) *Output* bagi pengguna android yang melakukan pemesanan melalui *smartphone* android adalah:
1. Informasi lokasi pengguna pada saat mengakses aplikasi.
 2. Informasi lokasi bengkel,
 3. Informasi jumlah antrian pada bengkel.
 4. Informasi rincian pemesanan.
- b) *Output* bagi pengguna administrator bengkel adalah:
1. Informasi pesanan masuk.
 2. Informasi status pesanan.
 3. Informasi edit profil bengkel.
- c) *Output* bagi pengguna admin android yang menandai dan menambahkan bengkel adalah:
1. Informasi data bengkel.
 2. Informasi data pengguna Android.

3.3 Perancangan Aplikasi dan Sistem

Setelah analisis kebutuhan pembuatan aplikasi diuraikan, selanjutnya adalah perancangan sistem dengan menggunakan UML (*Unified Modelling Language*). Dengan menggunakan UML perancangan dilakukan untuk menguraikan relasi antara pengguna dengan aplikasi. Adapun UML yang digunakan adalah *Use Case Diagram* dan *Activity Diagram* untuk menjelaskan cara kerja aplikasi.

3.3.1 Activity Diagram

Activity diagram akan menjelaskan bagaimana proses aplikasi tersebut mulai bekerja sampai aplikasi tersebut selesai digunakan.



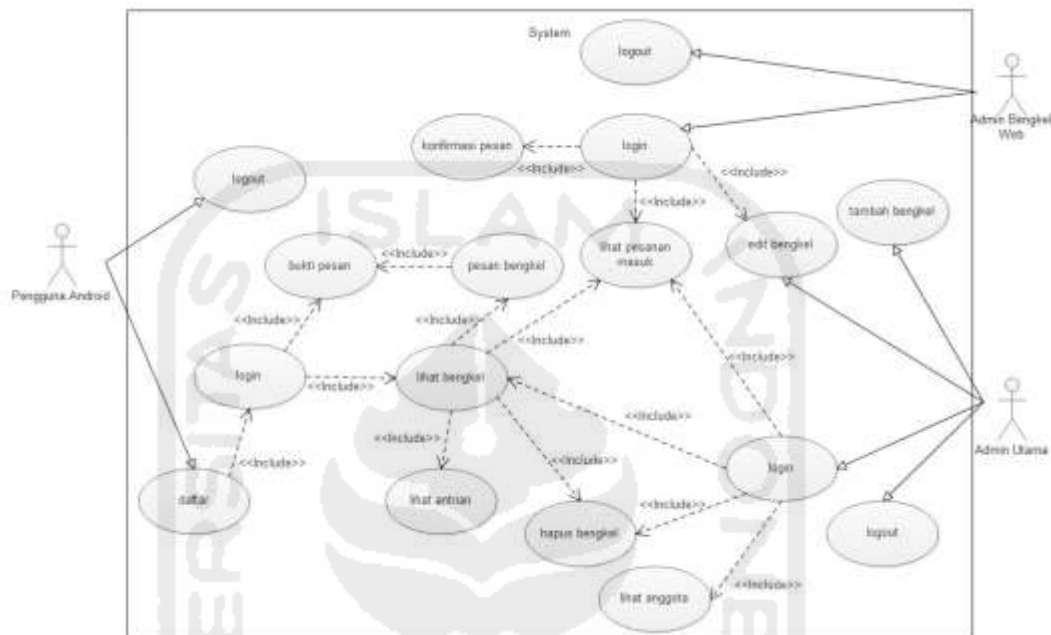
Gambar 3.1 Activity Diagram

Gambar di atas menjelaskan bagaimana hubungan aktivitas – aktivitas pengguna dengan aplikasi, yaitu:

1. Admin bengkel utama menggunakan aplikasi Android untuk melihat daftar bengkel serta aktivitas edit, hapus dan menambahkan atau menandai lokasi, kemudian juga dapat melihat daftar bengkel, pengguna Android, serta pesanan masuk yang dapat diakses melalui *web browser*.
2. Pengguna Android menggunakan aplikasi Android untuk melakukan pendaftaran dan kemudian *login* agar bisa melihat lokasi bengkel pada halaman utama dan melakukan pemesanan sehingga dapat melihat bukti pemesanan.
3. Admin bengkel web menggunakan *web browser* untuk *login* agar bisa masuk ke halaman utama untuk melihat pesanan masuk serta untuk edit akun.

3.3.2 Use Case Diagram

Selanjutnya adalah menggunakan *Use Case Diagram*, diagram ini digunakan untuk melihat hubungan yang terjadi antara pengguna dengan aplikasi serta aktivitas yang dapat dilakukan.



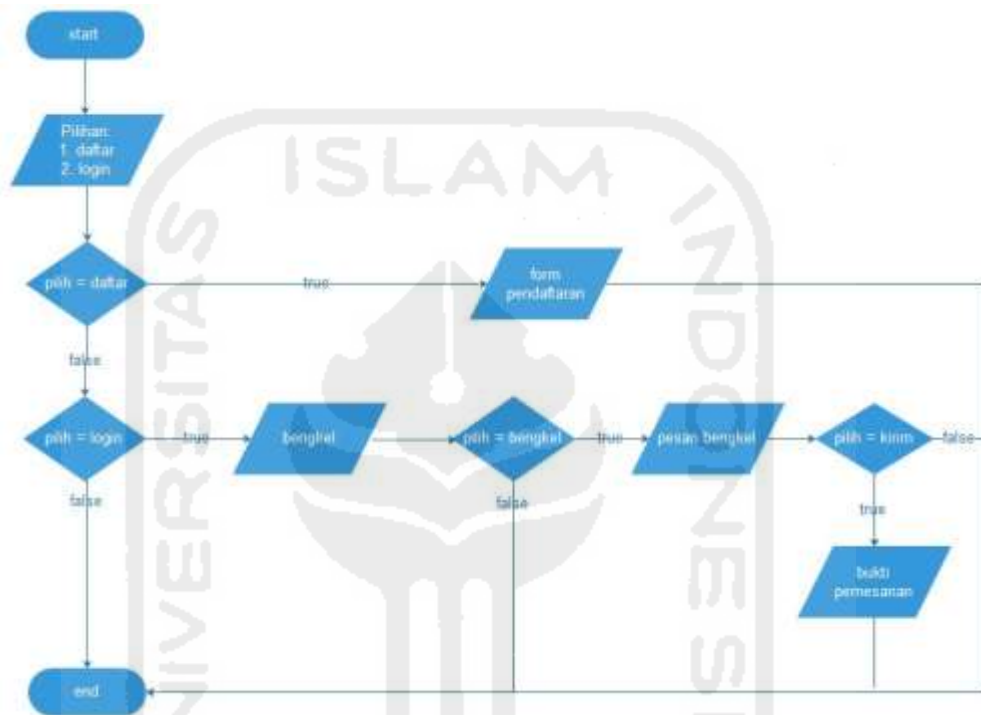
Gambar 3.2 Use Case Diagram

Pada gambar di atas dijelaskan bahwa pengguna Android dapat melakukan pendaftaran terlebih dahulu kemudian *login* agar dapat melakukan pemesanan ke salah satu bengkel yang dipilih. Sedangkan untuk admin bengkel utama, yaitu yang menandai dan menambah bengkel dapat melihat bengkel yang sudah terdaftar serta lokasi bengkel tersebut serta menghapusnya melalui *smartphone* Android, untuk dapat melihat daftar pengguna Android atau anggota, data dari bengkel yang terdaftar dan pesanan yang masuk ke seluruh bengkel harus *login* dahulu melalui *web browser*.

Adapun pada pengguna admin bengkel web, yaitu administrator bengkel harus melakukan *login* dulu untuk masuk ke sistem, kemudian akan dapat melihat pesanan yang masuk ke bengkelnya, hal lain yang dapat dilakukan adalah mengkonfirmasi pesan serta dapat mengedit bengkel.

3.3.3 Flowchart

Flowchart merupakan bagaimana alur logika yang akan diterapkan pada aplikasi atau sistem yang akan dibuat. Pada diagram ini dapat dilihat cara kerja aplikasi pencarian bengkel ini dalam melakukan pemesanan.



Gambar 3.3 *Flowchart*

Pada gambar diatas dijelaskan bagaimana proses pengguna aplikasi dengan aplikasi tersebut adalah sebagai berikut:

1. Pengguna mengakses aplikasi Android dari *smartphone*.
2. Pengguna harus *login* dahulu sebelum masuk aplikasi, jika belum terdaftar harus melakukan pendaftaran dulu.
3. Kemudian aplikasi akan menampilkan bengkel yang telah ada.
4. Pengguna kemudian memilih bengkel yang akan dipesan.
5. Pengguna mengisi form pemesanan dan kemudian dikirim.
6. Pengguna mendapatkan bukti pemesanan.

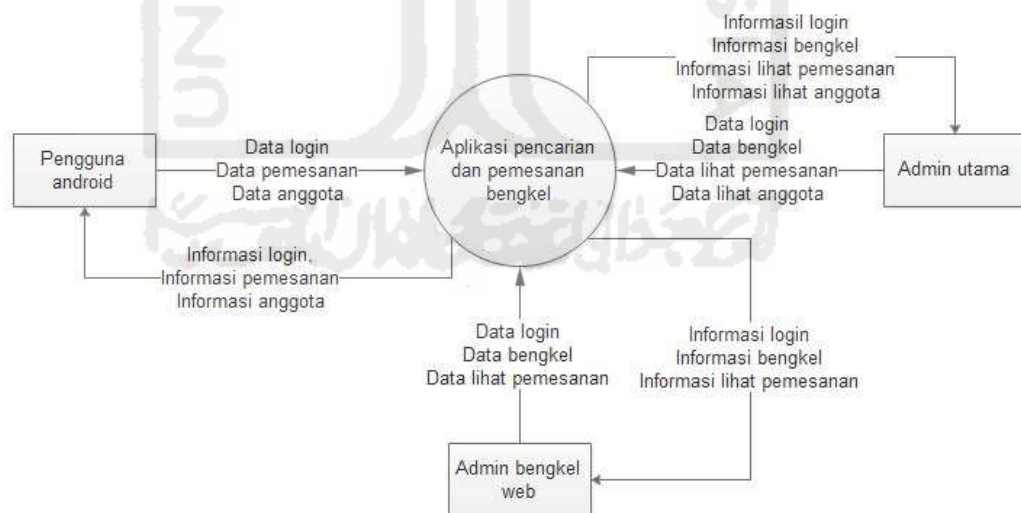
3.4 Pemodelan Proses Bisnis

Pemodelan proses bisnis bertujuan untuk memudahkan pemahaman alur suatu proses secara terintegrasi. Pada aplikasi pencarian dan pemesanan pada bengkel honda ini akan memiliki beberapa pengguna. Pengguna pertama merupakan pengguna yang ingin menservis motornya, dalam hal ini pengguna akan datang pada bengkel motor yang dituju, kemudian akan mengisi form untuk mengisi identitas. Pengguna kedua merupakan bengkel itu sendiri, dalam hal ini pihak bengkel akan mendaftarkan bagi yang ingin menservis motor dan menanyakan keluhan atau kerusakan apa yang ingin diperbaiki.

Diagram yang digunakan dalam proses bisnis ini adalah *Data Flow Diagram* (DFD). DFD merupakan pemodelan yang menggambarkan alur data mulai dari awal masuk kemudian di proses atau disimpan dalam suatu sistem hingga menjadi informasi yang dapat digunakan.

3.4.1 DFD Level 0

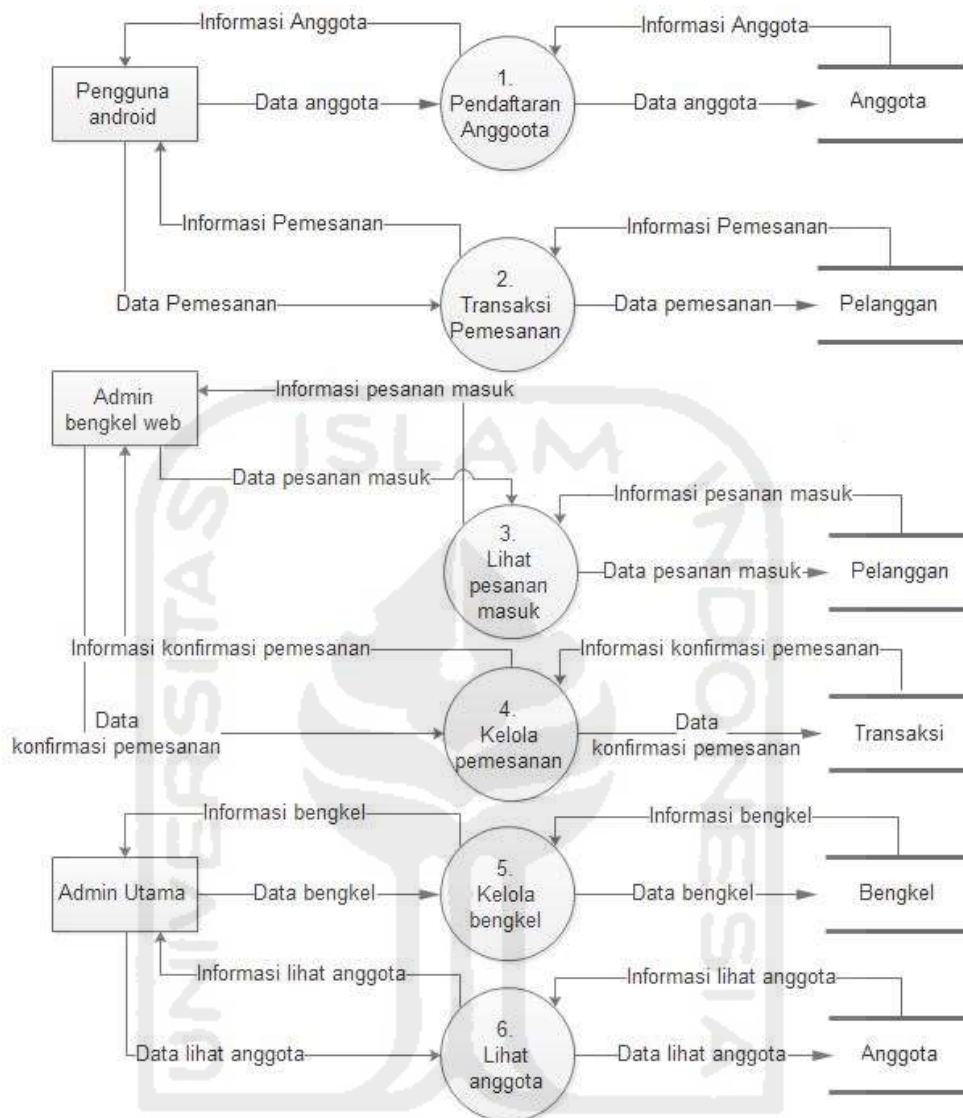
Pada DFD level ini menggambarkan proses bisnis secara umum. Pada aplikasi dan sistem ini terdapat tiga entitas yaitu, pengguna Android, admin utama, dan admin bengkel web.



Gambar 3.4 DFD Level 0

3.4.2 DFD Level 1

Pada DFD level 1 merupakan penggambaran proses bisnis yang lebih rinci dari DFD level 0.



Gambar 3.5 DFD Level 1

3.5 Perancangan Database

Perancangan basis data pada aplikasi pencarian bengkel ini, dibuat berdasarkan kebutuhan *input* yang diperlukan. Terdapat 7 (tujuh) tabel pada aplikasi ini yang akan menyimpan data yang diperlukan oleh aplikasi. Adapun tabelnya adalah sebagai berikut.

3.5.1 Tabel Bengkel

Pada tabel ini akan berisi data bengkel yang dimasukkan. Akan terdapat 7 kolom, yaitu *id_bkl* sebagai *primary key*, kemudian ada *username* yang *unique*, *password*, *nama*, *alamat*, *latitude*, dan *longitude*.

No	Nama Kolom	Tipe Data	Keterangan
1	id_bkl	int	<i>Primary Key</i>
2	username	varchar (50)	<i>unique</i>
3	password	varchar (50)	
4	nama	varchar (50)	
5	alamat	varchar (50)	
6	latitude	varchar (50)	
7	longitude	varchar (50)	

Tabel 3.1 Tabel Bengkel

3.5.2 Tabel Pelanggan

Pada tabel ini akan berisi data pelanggan dan pesanan yang dimasukkan. Akan terdapat 9 (sembilan) kolom yaitu, id_plg sebagai *primary key*, id_bkl sebagai *foreign key*, kemudian ada nama, no_stnk yang *unique*, keluhan, tanggal_plg, no_tsk dan waktu.

No	Nama Kolom	Tipe Data	Keterangan
1	id_plg	int	<i>Primary Key</i>
2	id_bkl	int	<i>Foreign key</i> pada field id_bkl di tabel bengkel
3	nama	varchar (50)	
4	no_stnk	varchar (8)	<i>unique</i>
5	keluhan	varchar (100)	
6	no_hp	varchar (12)	
7	tanggal_plg	datetime	
8	no_tsk	varchar (20)	
9	waktu	varchar (50)	

Tabel 3.2 Tabel Pelanggan

3.5.3 Tabel Transaksi

Pada tabel ini akan juga akan berisikan data pelanggan dan pesanan, akan tetapi data pelanggan yang motornya telah di servis yang datanya berasal dari tabel pelanggan, terdapat 10 (sepuluh) kolom yaitu, id_tsk sebagai *primary key*,

id_bkl, id_plg, kemudian ada nama, no_stnk, keluhan, tanggal_plg, dan no_tsk dan waktu.

No	Nama Kolom	Tipe Data	Keterangan
1	id_tsk	int	<i>Primary Key</i>
2	id_plg	int	
3	id_bkl	int	
4	nama	varchar (50)	
5	no_stnk	varchar (8)	<i>unique</i>
6	keluhan	varchar (100)	
7	no_hp	varchar (12)	
8	tanggal_plg	datetime	
9	no_tsk	varchar (20)	
10	waktu	varchar (50)	

Tabel 3.3 Tabel Transaksi

3.5.4 Tabel Admin

Pada tabel ini akan menyimpan data dari administrator utama untuk melakukan *login* pada *web browser*. Terdapat 3 (tiga) kolom pada tabel ini yaitu, id_admin, username, dan password.

No	Nama Kolom	Tipe Data	Keterangan
1	id_admin	int	<i>Primary Key</i>
2	username	varchar (50)	
3	password	varchar (50)	

Tabel 3.4 Tabel Admin

3.5.5 Tabel Anggota

Pada tabel ini akan menyimpan data pengguna Android yang telah melakukan pendaftaran agar bisa *login*. Terdapat 5 (lima) kolom pada tabel anggota yaitu, id_anggota, nama, password, email, dan nohp.

No	Nama Kolom	Tipe Data	Keterangan
1	id_anggota	int	<i>Primary Key</i>
2	nama	varchar (50)	

3	password	varchar (50)	
4	email	varchar (50)	<i>Unique</i>
5	nohp	varchar (12)	

Tabel 3.5 Tabel Anggota

3.5.6 Tabel History

Pada tabel ini akan menyimpan data pelanggan dan pesanan yang berasal dari tabel pelanggan. Tabel ini akan diakses oleh administrator utama untuk mendapatkan data. Terdapat 10 (sepuluh) kolom yaitu, id_history sebagai *primary key*, id_bkl, id_plg, kemudian ada nama, no_stnk, keluhan, tanggal_plg, dan no_tsk dan waktu.

No	Nama Kolom	Tipe Data	Keterangan
1	id_history	int	<i>Primary Key</i>
2	id_plg	int	
3	id_bkl	int	
4	nama	varchar (50)	
5	no_stnk	varchar (8)	<i>unique</i>
6	keluhan	varchar (100)	
7	no_hp	varchar (12)	
8	tanggal_plg	datetime	
9	no_tsk	varchar (20)	
10	waktu	varchar (50)	

Tabel 3.6 Tabel History

3.5.7 Tabel Bengkelnonaktif

Pada tabel ini akan menyimpan data bengkel yang tidak aktif atau bengkel yang tidak ditampilkan dalam *Google Maps*. Akan terdapat 8 (delapan) kolom, yaitu id_bklnon sebagai *primary key*, kemudian ada id_bkl, username yang *unique*, password, nama, alamat, latitude, dan longitude.

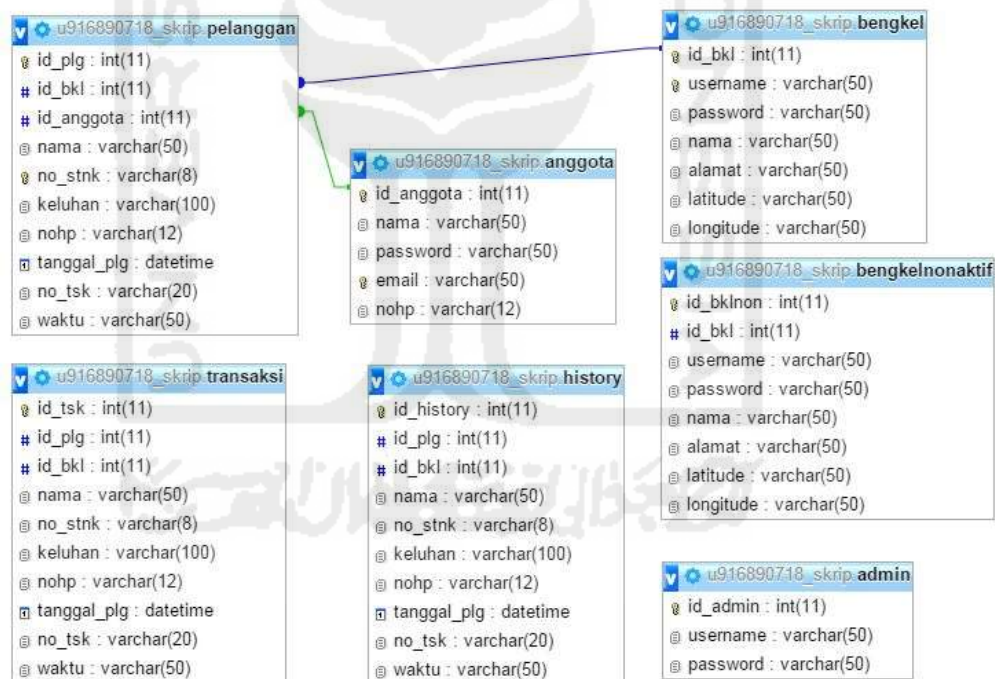
No	Nama Kolom	Tipe Data	Keterangan
1	id_bklnon	int	<i>Primary Key</i>
2	id_bkl	int	

3	username	varchar (50)	<i>unique</i>
4	password	varchar (50)	
5	nama	varchar (50)	
6	alamat	varchar (50)	
7	latitude	varchar (50)	
8	longitude	varchar (50)	

Tabel 3.7 Tabel Bengkelnonaktif

3.5.8 Relasi Tabel

Relasi tabel yang terjadi meliputi 3 (tiga) tabel yaitu tabel bengkel dengan tabel pelanggan. Adapun jenis relasi kedua tabel tersebut adalah one to many, jadi setiap pelanggan hanya dapat memesan satu bengkel. Kemudian tabel pelanggan dengan tabel anggota.



Gambar 3.6 Relasi Tabel

3.6 Perancangan Antarmuka

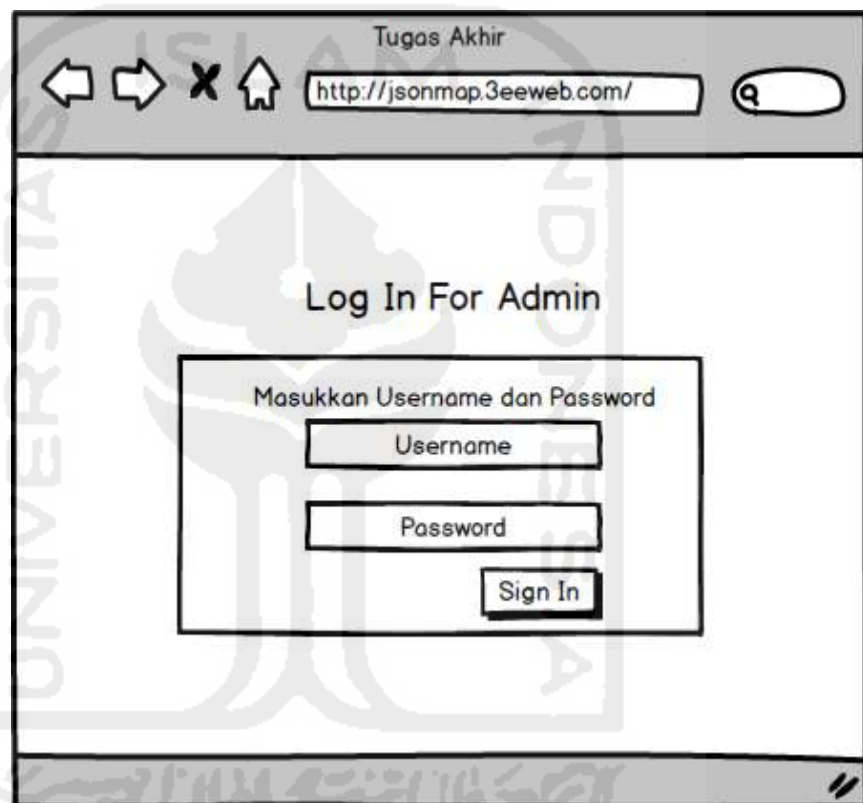
Perancangan antarmuka bertujuan untuk menampilkan aplikasi agar terlihat sederhana. Pada aplikasi pencarian bengkel ini akan terdapat 2 macam antarmuka, yang pertama yaitu antarmuka ketika aplikasi diakses dari *smartphone*

android dan yang kedua yaitu, ketika antarmuka untuk admin berbasis *web* yang diakses melalui *browser* di komputer.

3.6.1 Rancangan Antarmuka pada Browser di Komputer

a) Halaman Login

Pada halaman ini, pengguna kedua yaitu administrator bengkel akan menampilkan halaman *login* untuk masuk ke sistem, yaitu mengisi *username* dan *password*.



Gambar 3.7 Rancang Halaman *Login*

b) Halaman Utama

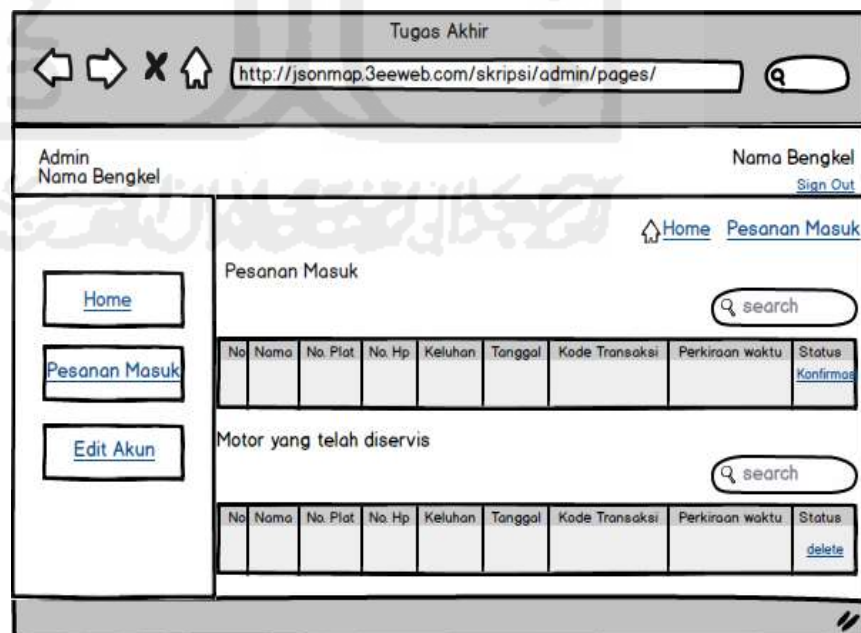
Halaman ini menampilkan tiga menu pilihan untuk dipilih oleh pengguna yang mengakses. Pada pojok kanan atas ketika nama bengkel di klik akan muncul tombol *sign out*.



Gambar 3.8 Rancang Halaman Utama

c) Halaman Pesanan Masuk

Pada halaman ini akan terdapat tampilan berupa tabel, yang akan menampilkan pesanan yang masuk dan motor yang telah selesai di servis.



Gambar 3.9 Rancang Halaman Pesanan Masuk

d) Halaman Edit Akun

Pada halaman ini akan menampilkan form untuk mengelola edit password, serta form untuk mengedit detail bengkel.

The image shows a web browser window with the title "Tugas Akhir" and the URL "http://jsonmap.3eeweb.com/". The page content is for an "Admin" user, with the header showing "Admin Nama Bengkel" and "Nama Bengkel Sign Out". A navigation menu on the left includes "Home", "Pesanan Masuk", and "Edit Akun". The main content area is titled "Halaman Edit Akun" and contains two sections: "Edit Password" and "Edit Detail Bengkel".

Edit Password Section:

- Form fields: Password lama, Password baru, Konfirmasi Password Baru
- Button: Simpan

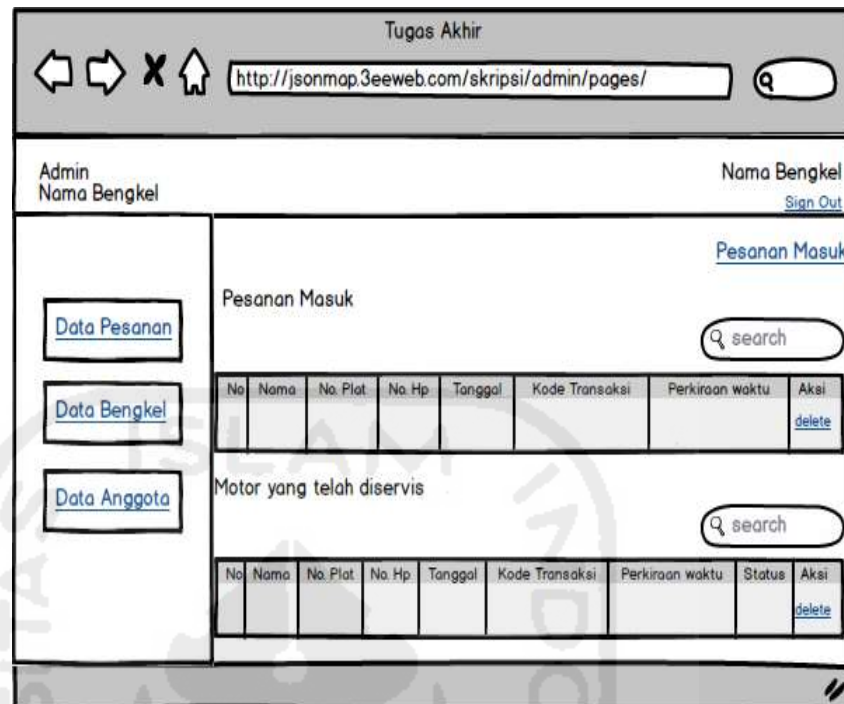
Edit Detail Bengkel Section:

- Form fields: Nama Bengkel, Alamat, Latitude, Longitude
- Button: Simpan

Gambar 3.10 Rancang Halaman Edit Akun

e) Halaman Pesanan Masuk pada Admin Utama

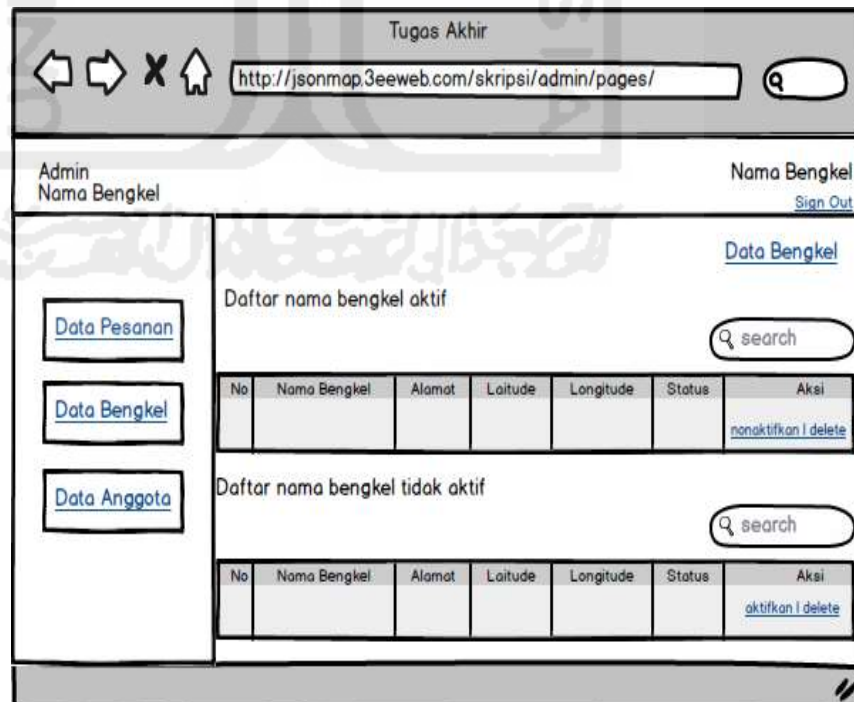
Pada halaman ini juga akan menampilkan pesanan yang masuk pada seluruh bengkel yang ada.



Gambar 3.11 Rancang Halaman Pesanan Masuk Admin Utama

f) Halaman Data Bengkel pada Admin Utama

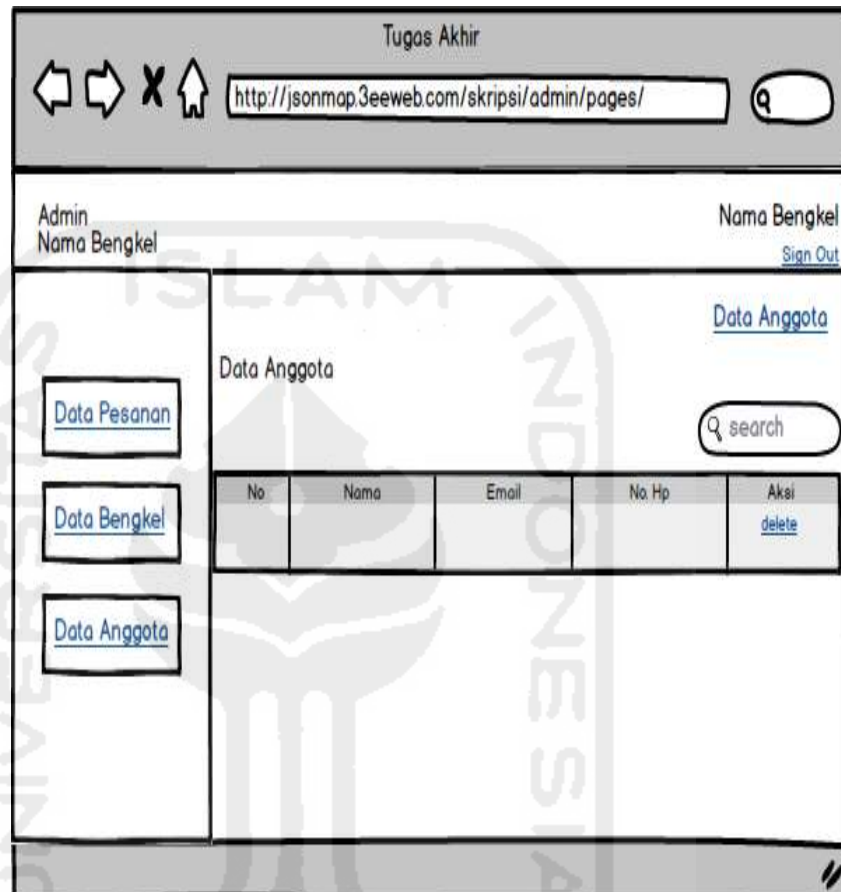
Pada halaman ini akan menampilkan tabel yang berisikan data bengkel yang aktif dan tidak aktif.



Gambar 3.12 Rancang Halaman Data Bengkel Admin Utama

g) Halaman Data Anggota pada Admin Utama

Pada halaman ini akan menampilkan data anggota atau pengguna Android yang telah melakukan pendaftaran.



Gambar 3.13 Rancang Halaman Data Anggota Admin Utama

3.6.2 Rancangan Antarmuka pada *Smartphone* Android

a) Halaman Awal

Pada gambar 3.14 merupakan halaman awal yang akan menampilkan *form* untuk *login* yaitu email dan passwod, sedangkan pada gambar 3.15 akan menampilkan *form* untuk melakukan pendaftaran berupa nama, email, password, dan no hp.

Masukkan email dan username

Email

Password

LOGIN

DAFTAR

Gambar 3.14 Rancang Halaman *Login*

Isilah form berikut

Nama

Email

Password

No. Hp

DAFTAR

Gambar 3.15

Rancang Halaman *Daftar*

b) Halaman *Utama*

Pada gambar 3.16, halaman ini merupakan halaman ketika aplikasi android dibuka setelah melakukan *login*, akan menampilkan peta beserta *marker* lokasi bengkel dan pengguna serta terdapat dua tombol, yaitu *history* dan *reload*.

Nama Bengkel
Jumlah antrian

Reload

History

Gambar 3.16 Rancang Halaman *Utama*

Nama

Masukkan Nama

No. Plat Kendaraan

Masukkan No. Plat Kendaraan

No. Hp

Masukkan No. Hp

Keluhan

(Opsional)

Pesan

Gambar 3.17 Rancang
Halaman *Pemesanan*

c) Halaman Pemesanan

Pada gambar 3.17, halaman ini akan menampilkan *form* yang akan diisi untuk melakukan pemesanan kepada bengkel yang dipilih.

d) Halaman Bukti Pemesanan

Pada gambar 3.18, halaman ini akan menampilkan detail pemesanan berupa nama pemesan, tanggal pemesanan, kode transaksi, dan perkiraan waktu servis setelah melakukan pemesanan.

Gambar 3.18 Rancang Halaman Bukti Pemesanan

Gambar 3.19 Rancang Pendaftaran Bengkel

e) Halaman Pendaftaran Bengkel

Pada gambar 3.19, halaman akan menampilkan form yang akan diisi untuk mendaftar dan menambahkan bengkel baru. Kemudian juga ada tombol latitude dan longitude untuk menangkap lokasi pada saat itu.

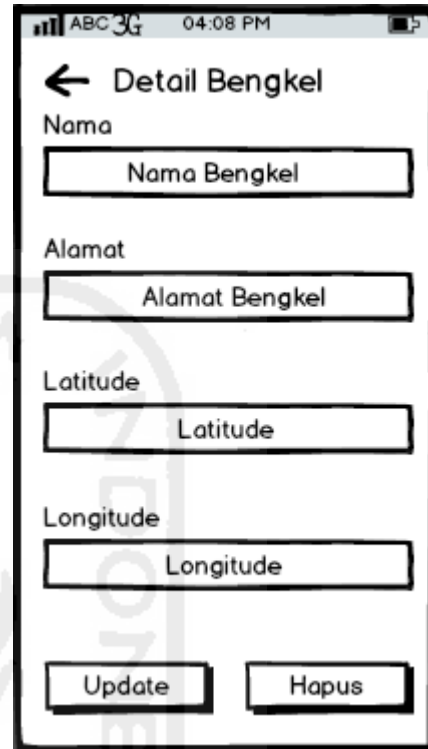
f) Halaman Daftar Bengkel

Pada gambar 3.20, halaman ini akan menampilkan sejumlah bengkel yang telah sudah terdaftar dan dapat diklik untuk menuju halaman

detail bengkel, serta terdapat tombol add untuk menuju halaman pendaftaran bengkel.



Gambar 3.20 Rancang Halaman Daftar Bengkel



Gambar 3.21 Rancang Halaman Detail Bengkel

g) Halaman Detail Bengkel

Pada gambar 3.21, halaman ini akan menampilkan aksi untuk mengedit dan menghapus bengkel yang dipilih, juga terdapat tombol kembali ke halaman daftar bengkel.

3.7 Skema dan Simulasi Aplikasi

Aplikasi digunakan ketika pengguna ingin melakukan servis motor pada bengkel Honda yang berada di kota Yogyakarta, kemudian pengguna biasanya langsung datang ke bengkel yang dituju untuk melakukan servis motor. Aplikasi ini dibangun agar bisa melihat jumlah antrian pada bengkel tertentu, sebelum datang pada bengkel yang diinginkan serta melakukan pemesanan.

Simulasi aplikasi dilakukan dengan cara membangun sebuah studi kasus permasalahan pemesanan pada suatu bengkel tersebut. Dengan studi kasus tersebut dapat dilihat apakah aplikasi yang dibangun sesuai dengan kebutuhan.