

**APLIKASI LAYANAN BENGKEL MOBIL BERBASIS ANDROID
DI KOTA BANDAR LAMPUNG**

(Skripsi)

Oleh
IVAN ALFATHI SAPUTRA



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG
2017**

ABSTRACT

APLICATION OF WORKING SERVICES BASED ON ANDROID IN BANDAR LAMPUNG CITY

By

IVAN ALFATIH SAPUTRA

The usage of GPS (*Global Positioning System*) in telecommunication device causes to emerge the technology of the local based service in Smartphone. The service proves the information which it can be accessed using the cellular network and it can be used to detect the user geographic position. In this research, the application is made to give the car repairing shop service which base on the android in Bandar Lampung, includes the changing profile, searching the location of car repairing shop service, the calling menu of car repairing shop service, and the history. The application consists of the Android Mobile for the user and the car repairing shop service along with web based system for super Administrator and administrator that it connects with web service technology. The result of the research showed that the application can give the car repairing shop based on the android in Bandar Lampung and the test is done by using the Black-Box equivalence partitioning that it can prove running as well as the requirement..

Keywords: *Android, Black-Box, Equivalence Partitioning, Global Positioning System (GPS), Location Based Service (LBS), repairing , web service*

ABSTRAK

APLIKASI LAYANAN BENGKEL MOBIL BERBASIS ANDROID BERBASIS ANDROID DI KOTA BANDARLAMPUNG

Oleh

IVAN ALFATIH SAPUTRA

Pemanfaatan teknologi GPS (*Global Positioning System*) pada alat telekomunikasi memicu munculnya teknologi Layanan Berbasis Lokasi pada *Smartphone*. Layanan ini menyediakan informasi yang dapat di akses melalui jaringan selular dan dapat menggunakan posisi geografis pengguna. Dalam penelitian ini, aplikasi dibuat untuk memberikan layanan bengkel mobil berbasis android di Bandarlampung, termasuk di dalamnya terdapat ubah profil, mencari bengkel, menu telepon bengkel dan *history*. Aplikasi terdiri dari *mobile Android* untuk pelanggan dan bengkel serta sistem berbasis web untuk super Administrator dan Administrator yang dihubungkan dengan teknologi *web service*. Hasil penelitian menunjukkan aplikasi dapat memberikan layanan bengkel mobil berbasis android di kota Bandarlampung dan pengujian yang dilakukan dengan menggunakan *Black-Box equivalence partitioning* menunjukkan aplikasi berjalan sesuai dengan *requirement*.

Kata Kunci: Android, Black-Box, Equivalence Partitioning, Global Positioning System (GPS), Location Based Service (LBS), bengkel, web service.

**APLIKASI LAYANAN BENGKEL MOBIL BERBASIS
ANDROID DI KOTA BANDAR LAMPUNG**

Oleh

IVAN ALFATIH SAPUTRA

Skripsi

Sebagai Salah Satu Syarat Untuk Memperoleh Gelar
SARJANA KOMPUTER

pada

Jurusan Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam



**JURUSAN ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS LAMPUNG**

2017

Judul Skripsi

**: APLIKASI LAYANAN BENGKEL MOBIL
BERBASIS ANDROID DI KOTA BANDAR
LAMPUNG**

Nama Mahasiswa

: Ivan Alfatih Saputra

Nomor Pokok Mahasiswa

: 1217051034

Jurusan

: Ilmu Komputer

Fakultas

: Matematika dan Ilmu Pengetahuan Alam

MENYETUJUI

1. Komisi Pembimbing



Febi Eka Febriansyah, S.T., M.T.
NIP 19800219 200604 1 001



Astria Hijriani, S.Kom., M.Kom.
NIP 19810308 200812 2 002

2. Ketua Jurusan Ilmu Komputer



Dr. Ir. Kurnia Muludi, M.S.Sc.
NIP 19640616 198902 1 001

MENGESAHKAN

1. Tim Penguji


Ketua

: **Febi Eka Febriansyah, S.T., M.T.**



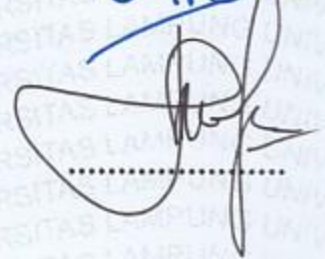
Sekretaris

: **Astria Hijriani, S.Kom., M.Kom.**



Penguji

Bukan Pembimbing : **Dwi Sakethi, S.Si., M.Kom.**

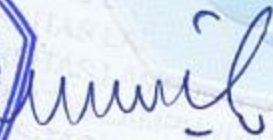


2. Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam



Prof. Warsito, S.Si., D.E.A., Ph.D.

NIP 19710212 199512 1 001



Tanggal Lulus Ujian Skripsi : 20 Juni 2017

PERNYATAAN

Saya yang bertanda tangan di bawah ini, menyatakan bahwa skripsi saya yang berjudul “Aplikasi Layanan bengkel mobil berbasis android di kota Bandar lampung” merupakan karya saya sendiri dan bukan hasil karya orang lain. Semua tulisan yang tertuang di skripsi ini telah mengikuti kaidah penulisan karya ilmiah Universitas Lampung. Apabila di kemudian hari terbukti bahwa skripsi saya merupakan hasil penjiplakan atau dibuat orang lain, maka saya bersedia menerimasanksi berupa pencabutan gelar yang telah saya terima.

Bandar Lampung, 20 Juni 2017



Ivan Alfatih Saputra
NPM. 1217051034

RIWAYAT HIDUP



Penulis dilahirkan pada tanggal November 1994 di Metro, Kota Metro. Penulis merupakan anak ketiga dari tiga bersaudara dengan ayah bernama Arzon Yudarni dan ibu bernama Esmiati Sundari. Penulis menyelesaikan pendidikan dasar di SD Pertiwi Teladan Kota Metro. Metro Lampung pada tahun 2006.

Kemudian melanjutkan pendidikan menengah pertama di SMP Negeri 4 Metro, Lampung dan selesai pada tahun 2009. Pendidikan menengah atas di SMA N 4 Metro Lampung yang diselesaikan penulis pada tahun 2012.

Pada tahun 2012, penulis terdaftar sebagai mahasiswa Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung melalui jalur UML (Ujian Masuk Lokal). Pada bulan Januari tahun 2015, penulis melakukan Kerja Praktik di Dinas Komunikasi dan Informatika Bandar Lampung. Pada bulan Juli tahun 2015 penulis melaksanakan Kuliah Kerja Nyata di Kabupaten Tulang Bawang Barat. Selama menjadi mahasiswa, penulis cukup aktif berorganisasi, diantaranya sebagai berikut:

1. Anggota Bidang Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer pada tahun periode 2013-2014.

2. Anggota Bidang Kaderisasi Himpunan Mahasiswa Jurusan Ilmu Komputer
pada tahun periode 2014-2015

.

PERSEMBAHAN

Puji dan syukur saya ucapkan kepada Allah SWT atas segala nikmat dan karunia-Nya sehingga skripsi ini dapat diselesaikan.

Kupersembahkan karya kecilku ini untuk:

*Ibuku, yang telah melahirkanku, merawatku, membesarkanku,
dan yang telah mendidikku.*

Ayahku tercinta, yang telah membesarkanku dengan seluruh kasih dan sayangnya, memberikan pengetahuannya, dan selalu mendukung serta mendoakan untuk keberhasilanku.

Kaka Adik serta keluarga besarku yang selalu kusayangi

dan, Almamater yang kubanggakan

UNIVERSITAS LAMPUNG

MOTTO

“Aksi tanpa visi hanya menghabiskan waktu.

Visi tanpa aksi hanyalah angan-angan.

Tetapi visi dengan misi dapat mengubah dunia.”

(Nelson Mandela)

Ingatlah bahwa seorang ahli atau pro berasal dari orang yang amatir juga.

Mulai capai mimpimu sekarang.”

(Ivan)

“ Kamu tidak bisa lari dari tanggung jawabmu besok dengan menghindarinya hari ini.”

(Abraham Lincoln)

SANWACANA

Puji dan syukur penulis ucapkan kehadiran Allah SWT atas segala rahmat, hidayah dan kesehatan yang diberikan kepada penulis sehingga dapat menyelesaikan penulisan skripsi ini. Skripsi ini disusun sebagai syarat untuk memperoleh gelar Sarjana Komputer di Jurusan Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Universitas Lampung. Judul dari skripsi ini adalah “Aplikasi Pencarian Bengkel Mobil Berbasis Android di Kota Bandar Lampung”.

Dalam penyusunan skripsi ini, penulis banyak menghadapi kesulitan. Namun, berkat bantuan dan dorongan dari berbagai pihak, akhirnya penulis dapat menyelesaikan skripsi ini. Untuk itu pada kesempatan ini, penulis mengucapkan terimakasih kepada:

1. Kedua orang tua, Bapak Arzon Yudarni, S.IP dan Ibu Esmiati Sundari, Kakak Rendi, Yuk Widy, Yuk Cindy dan Kak Bagus , serta keluarga besar yang selalu memberikan doa, motivasi dan kasih sayang yang tak terhingga.
2. Bapak Febi Eka Febriansyah, S.T., M.T. sebagai pembimbing I, pembimbing akademik. Yang telah membimbing penulis dan memberikan ide, kritik serta saran sehingga penulisan skripsi ini dapat diselesaikan.
3. Ibu Astria Hijriani, S. Kom., M. Kom.sebagai pembimbing II, yang telah memberikan saran, bantuan dan membimbing penulis dalam pembuatan skripsi ini.

4. Bapak Dwi Sakethi S.Si, M.Kom. sebagai pembahas, yang telah memberikan masukan-masukan yang bermanfaat dalam perbaikan skripsi ini.
5. Bapak Prof. Warsito, S.Si., D.E.A., Ph.d. selaku Dekan FMIPA Universitas Lampung.
6. Bapak Dr. Ir. KurniaMuludi, M.S.Sc. selaku Ketua Jurusan Ilmu Komputer FMIPA Universitas Lampung
7. Bapak Didik Kurniawan, S.Si., M.T. selaku Sekertaris Jurusan Ilmu Komputer FMIPA Universitas Lampung.
8. Dosen dan Karyawan Jurusan Ilmu Komputer FMIPA Universitas Lampung.
9. Seluruh angkatan Ilmu Komputer 2012 dan Anak Kontrakan. Kebersamaan yang telah dilalui menjadi pengalaman berharga bagi penulis.
10. Kepada Dian Kusuma Dewi, S.E dan Keluarga yang selalu menyemangati dan memberikan saran dalam skripsi sampai selesai.
11. Kepada Wildcats Baseball-Softball yang selali memberikan motivasi agar terselesaikan nya skripsi.
12. Kepada Usop, Vicky, Ardi yang selalu memberikan motivasi agar terselesaikan nya skripsi.
13. Kepada teman Pancenista yang selalu memberikan *support* agar terselesaikan nya skripsi.

Penulis menyadari bahwa skripsi ini masih jauh dari kesempurnaan, akan tetapi sedikit harapan semoga skripsi ini bermanfaat bagi perkembangan ilmu pengetahuan terutama bagi teman-teman Ilmu Komputer.

Bandar Lampung, 20 Juni 2017

Ivan Alfatih Saputra

Daftar Isi

ISI	HALAMAN
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xvi
DAFTAR TABEL.....	xvii
DAFTAR KODE	xix

BAB I PENDAHULUAN

1.1 Pendahuluan.....	1
1.2 Rumusan Masalah.....	3
1.3 Batasan Masalah	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5

BAB II TINJAUAN PUSTAKA

2.1 Android	6
2.1.1 Arsitektur Android.....	7
2.1.3 Android SDK	10
2.1.4 Eclipse.....	11
2.1.5 Android Development Tools (ADT).....	13

2.1.6 Fundamental Aplikasi	13
2.2 <i>Global Positioning System</i>	15
2.3 Local Based System	15
2.4 Point of Interest	17
2.5 Google Maps API	18
2.5.1 Pengertian API	18
2.5.2. Pengertian Google Maps API	18
2.6 Pengertian PHP	18
2.6.1 Keunggulan PHP	19
2.7 Pengertian Mysql	20
2.7.1 Keunggulan Mysql	20
2.8 JSON(Java Script Object Natation)	21
2.9 Pengertian HTML	22
2.10 Xampp	24
2.11 Bengkel	24
2.12 Metodologi Pengembangan Sistem	25
2.12.1 <i>Extreme Pemrogramming</i>	25
2.12.2 <i>Unified Modeling Language (UML)</i>	27
2.13 Pengujian Perangkat Lunak	35
2.13.1 <i>Black Box</i>	37

BAB III METODOLOGI PENELITIAN

3.1 Waktu Tempat dan Penelitian	39
---------------------------------------	----

3.2 Alat Pendukung Penelitian.....	39
3.3 Metodologi Penelitian.....	40
3.3.1 Diagram Alir Metodologi Penelitian	40
3.3.2 Metode Pengembangan Sistem.....	44
3.3.3 Metode Pengumpulan Data.....	45
3.4 Analisis dan Perancangan Sistem	46
3.4.1 Identifikasi Masalah	46
3.4.2 Analisis Sistem.....	47
3.4.3 Analisis <i>User Requirement</i>	47
3.4.4 Perancangan Sistem.....	48
3.4.4.1 Perancangan UML (<i>Unified Modelling Language</i>)	48
3.4.4.2 Perancangan Antarmuka.....	65
3.5 Metode Pengujian Sistem	99

BAB IV HASIL DAN PEMBAHASAN

4.1 Hasil	104
4.2 Implementasi Sistem	105
4.3 Tampilan Sistem Super Admin.....	110
4.3.1 Tampilan Halaman Login	110
4.3.2 Tampilan Halaman Data Super Admin.....	111
4.3.3 Tampilan Halaman Edit Super Admin	112
4.3.4 Tampilan Halaman Tambah SuperAdmin	113

4.4.4 Tampilan Halaman produsen	113
4.4.5 Tampilan Halaman Tambah Produsen.....	115
4.4.6 Tampilan Halaman Ubah Produsen	115
4.4.7 Tampilan Halaman Kerusakan	116
4.4.8 Tampilan Halaman Tambah Kerusakan	117
4.4.9 Tampilan Halaman Edit Kerusakan	118
4.4.10 Tampilan Halaman Data Bengkel.....	118
4.4.11 Tampilan Halaman Tambah Bengkel	119
4.4.12 Tampilan Halaman Edit Bengkel.....	120
4.4.13 Tampilan Halaman Data Pelanggan	121
4.4.14 Tampilan Halaman Data Montir	121
4.4.15 Tampilan Halaman Tambah Montir	123
4.4.16 Tampilan Halaman Edit Montir.....	123
4.4.17 Tampilan Halaman Perbaikan.....	124
4.5 Tampilan Sistem Admin	125
4.5.1 Tampilan Halaman Admin	125
4.5.2 Tampilan Halaman Beranda	125
4.5.3 Tampilan Halaman Profil	126
4.5.4 Tampilan Halaman Data Montir	127
4.5.5 Tampilan Halaman Tambah Montir	128
4.5.6 Tampilan Halaman Edit Montir.....	129
4.5.7 Tampilan Halaman Data Kerusakan	130
4.5.8 Tampilan Halaman Tambah Kerusakan	131

4.5.9 Tampilan Halaman Edit Kerusakan	131
4.5.10 Tampilan Halaman Perbaikan.....	132
4.6 Tampilan Aplikasi Berbasis Mobile(Bengkel)	133
4.6.1 Tampilan Spalsh Screen	133
4.6.2 Tampilan Menu Home	135
4.6.3 Tampilan Lokasi Pengguna	136
4.6.4 Tampilan Menu Profil.....	138
4.6.5 Tampilan Menu Montir.....	140
4.6.6 Tampilan Menu Kerusakan.....	141
4.6.7 Tampilan Menu Perbaikan.....	143
4.7 Tampilan Aplikasi Berbasis Mobile (Pelanggan).....	144
4.7.1 Tampilan Menu Registrasi.....	145
4.7.2 Tampilan Home	147
4.7.3 Tampilan Ubah Profil	149
4.7.4 Tampilan Daftar Kerusakan.....	151
4.7.5 Tampilan Peta Keluhan.....	153
4.7.6 Tampilan Informasi	155
4.7.7 Tampilan <i>History</i>	157
4.8 Pengujian Aplikasi.....	159
4.8.1 Pengujian Fungsional.....	159
4.8.1.1 Pengujian versi android	160
4.8.1.2 Pengujian User Interface	161
4.8.1.3Penguian Fungsi dari Menu Aplikasi	162

4.8.2 Pengujian Non Fungsional.....	167
4.8.3 Analisis Hasil Pengujian Non Fungsional	166

BAB V KESIMPULAN

5.1 Kesimpulan	171
5.2 Saran	172

DAFTAR PUSTAKA

LAMPIRAN

DAFTAR TABEL

	Halaman
3.1 Pengujian Versi Android	99
3.2 Pengujian Interface	99
3.3 Pengujian Fungsi Menu Aplikasi	100
3.4 Pengujian <i>user interface</i>	102
3.5 Pengujian Fungsi Menu Aplikasi	102
4.1 Daftar Utama Sistem Berbasis <i>Web</i> (Super Admin).....	105
4.2 Daftar Utama Sistem Berbasis <i>Web</i> (Admin).....	106
4.3 Daftar Utama <i>web service</i> (admin & <i>super admin</i>).....	106
4.4 Daftar <i>Class</i> dan <i>Layout</i> Utama Aplikasi Android (Bengkel).....	107
4.5 Daftar <i>Class</i> dan <i>Layout</i> Utama Aplikasi Android (Pelanggan).....	108
4.6 Daftar <i>Class</i> dan <i>Layout</i> Aplikasi Android (Pelanggan).....	109
4.7 Daftar <i>Class</i> dan <i>Layout</i> Aplikasi Android (Bengkel).....	109

DAFTAR GAMBAR

HALAMAN

2.1 Daftar Arsitektur	7
2.2 Komponen dasar LBS.....	17
2.3 <i>Point of Interest</i>	17
2.4 Siklus <i>Extreme Programming</i>	26
3.1 Diagram Alir Penelitian.....	41
3.2 Use Case Diagram	49
3.3 Diagram Cari bengkel.....	51
3.4 <i>Activity Diagram</i> Input Data Kerusakan Mobil	52
3.5 <i>Activity Diagram</i> History	53
3.6 <i>Activity Diagram</i> Akses Kerusakan Mobil Pelanggan.	54
3.7 <i>Activity Diagram</i> Kelola Data Bengkel	55
3.8 Diagram Kelola Data Seluruh Bengkel.....	56
3.9 <i>Sequence Diagram Logout</i>	57
3.10 <i>Activity Diagram</i> Kelola Data <i>Logout</i>	57
3.11 <i>Sequence Diagram</i> Cari Bengkel.....	59
3.12 <i>Sequence Diagram</i> Input Data Kerusakan Mobil	59
3.13 <i>Sequence Diagram History</i>	60
3.14 <i>Sequence Diagram</i> Kerusakan Mobil Pelanggan	60
3.15 <i>Sequence Diagram</i> Kerusakan Mobil Pelanggan	62
3.16 <i>Sequence Diagram</i> Kelola Data Seluruh Bengkel	63
3.17 <i>Sequence Diagram</i> Kelola Data Bengkel.....	64
3.18 <i>Class Diagram</i> Bengkel mobil	64
3.19 <i>Design Layout Splash Screen</i>	67

3.20 <i>Design Layout</i> Menu Profil	68
3.21 <i>Design Layout</i> Ubah profil	69
3.22 <i>Design Layout</i> kerusakan	70
3.23 <i>Design Layout</i> Respon Bengkel	71
3.24 <i>Design Layout</i> Cek no.telepon	72
3.25 <i>Design Layout</i> Pilih Produsen	73
3.26 <i>Design Layout</i> Isi Nama	74
3.27 <i>Design Layout</i> Menu History	75
3.28 <i>Design Layout</i> Login Bengkel.....	76
3.29 <i>Design Layout</i> List Kerusakan	77
3.30 <i>Design Layout</i> Input Kerusakan Mobil	78
3.31 <i>Design Layout</i> Profil	79
3.32 <i>Design Layout</i> Ubah Profil Bengkel	80
3.33 <i>Design Layout</i> Montir	81
3.34 <i>Design Layout</i> Tambah Data Montir	82
3.35 <i>Design Layout</i> Ubah Data Montir	83
3.36 <i>Design Layout</i> Kerusakan	84
3.37 <i>Design Layout</i> Tambah Kerusakan	85
3.38 <i>Design Layout</i> Ubah Kerusakan.....	86
3.39 <i>Design Layout</i> Perbaikan.....	87
3.40 <i>Design Layout</i> Login Admin	88
3.41 <i>Design Layout</i> Menu Home.....	88
3.42 <i>Design Layout</i> Menu Admin	89
3.43 <i>Design Layout</i> Menu Kerusakan.	90
3.44 <i>Design Layout</i> Menu Produsen	90
3.45 <i>Designx Layout</i> Menu Bengkel	91
3.46 <i>Design Layout</i> Menu Montir.	92
3.47 <i>Design Layout</i> Menu Pelanggan.....	92
3.48 <i>Design Layout</i> Menu Perbaikan	93
3.49 <i>Design Layout</i> Menu Tambah Data.....	93
3.50 <i>Design Layout</i> Menu Edit Data	94

3.51 <i>Design Layout</i> Login Admin	95
3.52 <i>Design Layout</i> Menu <i>Home</i>	95
3.53 <i>Design Layout</i> Menu Kerusakan	96
3.54 <i>Design Layout</i> Menu <i>Montir</i>	97
3.55 <i>Design Layout</i> Menu Kerusakan	97
3.56 <i>Design Layout</i> Menu Perbaikan	98
4.1 Halaman Super Admin	110
4.2 Halaman Beranda	110
4.3 Halaman Data Super Admin.....	111
4.4 Halaman Edit Super Admin.....	112
4.5 Halaman Tambah Super Admin	113
4.6 Halaman Data Produsen	113
4.7 Halaman Tambah Produsen.....	115
4.8 Halaman Ubah produsen	116
4.9 Tampilan Halaman Kerusakan	117
4.10 Halaman Tambah Kerusakan	117
4.11 Halaman Edit Kerusakan	118
4.12 Halaman Data Bengkel.....	118
4.13 Halaman Tambah Bengkel	119
4.14 Halaman Edit Bengkel.....	120
4.15 Halaman Data Pelanggan	121
4.16 Halaman Data montir	121
4.17 Halaman Tambah Montir	123
4.18 Halaman Ubah Montir.....	123
4.19 Halaman Perbaikan.....	124
4.20 Halaman Admin.....	125
4.21 Halaman Beranda	125
4.22 Halaman Profil.....	126
4.23 Halaman Data Montir	127
4.24 Halaman Tambah Montir	128
4.25 Halaman Edit Montir.....	129

4.26 Tampilan Halaman Data Kerusakan.....	130
4.27 Halaman Tambah Kerusakan	131
4.28 Halaman Edit Kerusakan	131
4.29 Halaman Perbaikan.....	132
4.30 Tampilan Splash Screen	133
4.31 Tampilan Menu Home.....	135
4.32 Tampilan Menu Home.....	136
4.33 Tampilan Menu Profil	138
4.34 Tampilan Menu Montir	140
4.35 Tampilan Menu Kerusakan	141
4.36 Tampilan Menu Perbaikan	143
4.37 Tampilan Register Pelanggan.....	145
4.38 Tampilan Home	147
4.39 Tampilan Ubah Profil	149
4.40 Tampilan Daftar kerusakan	151
4.41 Tampilan Peta Keluhan	153
4.42 Tampilan Informasi	155
4.43 Tampilan <i>History</i>	157

DAFTAR KODE

HALAMAN

4.1 Daftar Kode Halaman Super Admin	112
4.2 Daftar Kode halaman Produsen	114
4.3 Daftar Kode Halaman Kerusakan.....	117
4.4 Daftar Kode Halaman Data Bengkel	119
4.5 Daftar Kode Halaman Data Pelanggan.....	121
4.6 Daftar Kode Halaman Data Montir	122
4.7Daftar Kode Halaman Perbaikan.....	125
4.8 Daftar Kode Halaman Profil	127
4.9 Daftar Kode Halaman Data Montir	128
4.10 Daftar Kode Halaman Kerusakan.....	130
4.11 Daftar Kode Halaman Perbaikan	133
4.12 Daftar Kode Tampilan Splash Screen	134
4.13 Potongan Kode Program XML Tampilan <i>Splash Screen</i>	134
4.14 Potongan Kode Program Tampilan Menu Home	135
4.15 Potongan Kode Program XML Tampilan Menu Home	136

4.16 Potongan Kode Program Tampilan Menu Lokasi Pengguna	137
4.17 Potongan Kode Program XML Tampilan Menu Lokasi Pengguna	138
4.18 Potongan Kode Program Tampilan Menu Profil.....	139
4.19 Potongan Kode Program XML Tampilan Menu Profil.....	139
4.20 Potongan Kode Program Tampilan Menu Montir.....	140
4.21 Potongan Kode Program XML Tampilan Menu Montir.....	141
4.22 Potongan Kode Program Tampilan Menu Kerusakan.....	142
4.23 Potongan Kode Program Menu Perbaikan	144
4.24 Potongan Kode Program XML Menu Perbaikan	145
4.25 Potongan Kode Program Register Pelanggan.....	146
4.26 Potongan Kode Program XML Tampilan Register Pelanggan.	147
4.27 Potongan Kode Program Tampilan Home	148
4.28 Potongan Kode Program XML Tampilan Home.....	148
4.29 Potongan Kode Program Ubah Profil.....	149
4.30 Potongan Kode Program XML Tampilan Ubah Profil.....	150
4.31 Potongan Kode Program Daftar Kerusakan	152
4.32 Potongan Kode Program XML Tampilan Daftar Kerusakan	153
4.33 Potongan Kode Program Peta Keluhan	154
4.34 Potongan Kode Program XML Tampilan Peta Keluhan	154
4.35 Potongan Kode Program Informasi	156

4.36 Potongan Kode Program XML Tampilan Informasi	157
4.37 Potongan Kode Program <i>History</i>	158
4.38 Potongan Kode Program XML Tampilan <i>History</i>	159

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pada zaman modern ini, *smartphone* merupakan sebuah kebutuhan. Saat ini *smartphone* juga telah dilengkapi dengan fasilitas-fasilitas canggih yang memungkinkan *user* melakukan apa pun yang dibutuhkan seperti fasilitas kamera berkualitas tinggi dengan jumlah *resolusi* yang besar untuk mengambil gambar dengan hasil yang maksimal baik untuk foto maupun video, ditambah fitur lain seperti GPS, *Wi-Fi hotspot*, dan juga berbagi koneksi. Selain itu, saat ini *smartphone* juga telah dilengkapi dengan sistem informasi seperti Android, iOS, dan juga Windows Phone. Khusus untuk sistem operasi Android yang dikembangkan oleh Google, hampir semua *source code* untuk aplikasi-aplikasinya dipublikasikan, sehingga setiap orang dapat mengembangkan aplikasi tersebut sesuai dengan keinginan dan juga kebutuhan masing-masing *user*.

Seiring dengan perkembangan ilmu dan teknologi, muncul suatu perkembangan teknologi yaitu GPS. Yang bertujuan untuk mengetahui letak tempat yang akan dituju dan mengetahui keberadaan pengguna dengan bantuan sinyal satelit. GPS dapat memberikan informasi yang tepat dan akurat mengenai posisi, kecepatan, arah dan waktu (Lukmana, 2014).

Smartphone dengan sistem operasi Android sudah tertanam GPS, dengan demikian setiap pengguna dapat mengetahui posisi di mana berada. Dengan adanya teknologi GPS pada Android ini dapat dimanfaatkan sebagai pemandu dalam pencarian suatu tempat. Salah satunya adalah bengkel mobil yang terdapat di kota Bandar Lampung.

Apabila ketika pengendara mobil mengalami kesulitan pada kendaraannya di tengah perjalanan seperti masalah pada radiator, masalah AC, masalah mesin, masalah lampu, dsb. Seringkali pengendara yang bukan berasal dari Bandar Lampung atau pun yang berasal dari Bandar Lampung mengalami kesulitan mencari letak bengkel untuk memperbaiki mobilnya.

Berdasarkan rumusan masalah tersebut perlu dibuat sebuah “Aplikasi layanan bengkel mobil berbasis Android di kota Bandar Lampung”. Menurut informasi tersebut pengguna aplikasi ini dapat mencari secara langsung letak bengkel mobil kapanpun dan di manapun di sekitar kota Bandar Lampung, sehingga tidak memerlukan waktu yang lama dalam pencarian letak bengkel mobil.

1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini adalah “Bagaimana merancang dan membangun aplikasi layanan bengkel mobil di kota Bandar Lampung”.

1.3 Batasan Masalah

Aplikasi layanan bengkel mobil berbasis Android di kota Bandar Lampung ini memiliki batasan sebagai berikut.

- 1 Aplikasi ini menggunakan *platform* Android, sehingga hanya dapat dijalankan pada perangkat bergerak berbasis Android dengan versi minimal Lollipop karena smartphone sekarang tidak ada yang memakai versi awal Android.
- 2 Aplikasi ini membutuhkan koneksi internet untuk dapat mengakses Google Maps.
- 3 Hanya bengkel mobil yang ada di dalam *database* agar lebih mudah untuk menambahkan dan menghapus.
- 4 Aplikasi ini mempunyai tombol telepon langsung agar bisa tersambung ke bengkel, sehingga memudahkan pengguna bila bengkel merespon terlalu lama.
- 5 Aplikasi ini mempunyai 4 *user*, yaitu pelanggan, bengkel dan super admin dan admin.
- 6 Aplikasi ini bisa menunjukkan jalan antara pengendara dan bengkel. Begitu pun sebaliknya, bengkel bisa mengetahui letak pasti pengendara yang mengalami kerusakan, sehingga pihak bengkel tidak perlu bingung lagi mencari dimana pengendara mengalami kerusakan.

1.4 Tujuan

Penelitian ini bertujuan untuk merancang dan membangun aplikasi layanan bengkel mobil di kota Bandar Lampung.

1.5 Manfaat

Manfaat yang diperoleh dari penelitian ini diharapkan dapat membantu pengguna sebagai petunjuk untuk mengakses informasi yang berkaitan dengan letak bengkel mobil sehingga dapat memudahkan pengguna dalam mencari letak bengkel di kota Bandar Lampung dengan menggunakan *mobile phone* berbasis Android.

BAB II

TINJAUAN PUSTAKA

2.1 Android

Android adalah sebuah sistem operasi untuk perangkat *mobile* yang menyertakan *middleware* (*virtual machine*) dan sejumlah aplikasi utama. Android merupakan modifikasi dari kernel Linux (Andry, 2011). Pada awalnya sistem operasi ini dikembangkan oleh sebuah perusahaan bernama Android Inc. Dari sinilah awal mula nama Android muncul. Android Inc. Adalah sebuah perusahaan *start-up* kecil yang berlokasi di Palo Alto, California, Amerika Serikat yang didirikan oleh Andy Rubin bersama Rich Miner, Nick Sears, dan Chris White. Pada bulan Juli 2005, perusahaan tersebut diakuisisi oleh Google dan para pendirinya bergabung ke Google. Andy Rubin sendiri kemudian diangkat menjadi Wakil Presiden divisi *Mobile* dari Google.

Tujuan pembuatan sistem operasi ini adalah untuk menyediakan *platform* yang terbuka, yang memudahkan orang mengakses internet menggunakan telepon seluler. Android juga dirancang untuk memudahkan pengembang membuat aplikasi dengan batasan yang minim sehingga kreativitas pengembang menjadi lebih berkembang (Andry,2011).

Sebagai *Open Source* dan bebas dalam memodifikasi, di dalam Android tidak ada ketentuan yang tetap dalam konfigurasi *Software* dan *Hardware*. Fitur- fitur yang didapat dalam Android antara lain (Lee, 2011) .

1. *Storage* - Menggunakan SQLite, *database* yang ringan, untuk sebuah penyimpanan data.
2. *Connectivity* - Mendukung GSM/EDGE, IDEN, CDMA, EV-DO, UMTS, *Bluetooth* (termasuk A2DP dan AVRCP), WiFi, LTE, dan WiMax.
4. *Messaging* –Mendukung SMS dan MMS.
5. *Web Browser* – Berbasiskan open-source WebKit, bersama mesin.
6. *Chrome's V8 JavaScript*.
7. *Media support* – Termasuk mendukung untuk beberapa media berikut :
H.263,H.264 (dalam bentuk 3GP or MP4), MPEG-4 SP, AMR,
AMRWB (dalam bentuk 3GP), AAC, HE-AAC (dalam bentuk MP4 atau 3GP), MP3, MIDI, Ogg Vorbis, WAV, JPEG, GIF, dan BMP.
8. *Hardware support* – Sensor akselerasi, Kamera, Kompas Digital, Sensor kedekatan, GPS.
9. *Multi-touch* – Mendukung *multi-touch screens*.
10. *Multi-tasking* – Mendukung aplikasi *multi-tasking*.
11. *Flash-support* – Android 2.3 mendukung *Flash* 10.1.
12. *Tethering* – Mendukung pembagian dari koneksi internet sebagai *wired/wireless hotspot*.

2.1.1 Arsitektur Android

Arsitektur Android dapat dilihat pada Gambar 2.1.



Gambar 2.1 Arsitektur Android (Developers, 2011).

Berikut ini merupakan penjelasan dari Arsitektur Android tersebut.

1. Linux Kernel

Pada bagian paling bawah dari Android adalah Linux Kernel. Android menggunakan modifikasi dari Linux Kernel versi 2.6. Bagian ini bertanggung jawab untuk mengelola dan berkomunikasi dengan perangkat keras di mana android berjalan. Pemilihan Linux Kernel sebagai inti dari Android adalah karena dukungan dan kestabilan terhadap berbagai macam komponen perangkat

keras. Pada bagian ini disediakan *driver* (program pengendali) perangkat keras, pengelolaan memori, pengelolaan proses, pengelolaan jaringan, dan keamanan.

2. *Libraries*

Libraries merupakan kumpulan kode yang dapat digunakan oleh komponen atau program lain. Pada bagian ini kodenya ditulis menggunakan bahasa pemrograman C/C++. Fungsi masing-masing komponen di *libraries* ini adalah sebagai berikut ,

1. *Surface Manager* : mengelola penggambaran dan komposisi komponen dilayar(jendela, tombol, teks input, objek tiga dimensi dan sebagainya).
2. Open GL|ES : *engine* grafis tiga dimensi.
3. SGL : *engine* grafis dua dimensi.
4. Media Framework : *codecs* untuk merekam dan memutar berbagai format video dan audio.
5. Free Type : untuk *rendering font*.
6. SSL : untuk keamanan komunikasi jaringan.
7. SQLite : pengelolaan database untuk penyimpanan data.
8. Webkit : *web browser engine* Android.
9. Libc : *library C* standar

3. Android Runtime

Pada bagian inilah aplikasi Android berjalan. Inilah bagian *middleware* yang disebutkan pada definisi Android di awal. Aplikasi Android ditulis menggunakan bahasa pemrograman Java dan untuk menjalankannya diperlukan sebuah *virtual machine* yang akan menerjemahkan instruksi-instruksi program Java tersebut ke instruksi yang dimengerti oleh sistem operasi. Namun berbeda dengan *platform* lainnya, android tidak menggunakan *Java Virtual Machine* (JVM), melainkan menggunakan *Dalvik Virtual Machine* (DVM). *Dalvik Virtual Machine* adalah sebuah *virtual machine* yang dioptimasi untuk perangkat yang memiliki memori kecil, sumber tenaga terbatas, dan kemampuan prosesor yang kecil.

4. Application Framework

Bagian ini adalah framework atau perangkat yang dapat digunakan oleh pengembang aplikasi dalam membuat aplikasi Android. Bagian ini menyertakan program-program yang mengelola fungsi-fungsi dasar seperti informasi lokasi, fungsi telepon, pengelolaan perpindahan antarprogram atau proses, dan sebagainya. Pengembang aplikasi dapat memiliki akses *framework* yang sama dengan yang dapat diakses oleh aplikasi-aplikasi utama dari android (*phone dialer*, buku telepon, SMS, *e-mail*, kalender, dan sebagainya). di sini terlihat keinginan dari android untuk menjadi platform yang terbuka, sehingga memungkinkan pengembang aplikasi untuk lebih kreatif menciptakan aplikasi yang menarik dan inovatif.

5. Applications

Pada bagian ini terdapat aplikasi-aplikasi yang dapat digunakan oleh pengguna perangkat Android. Selain aplikasi inti seperti yang ada di diagram arsitektur, aplikasi-aplikasi tambahan yang dipasang sendiri oleh pengguna akan menempati bagian ini dan memiliki hak akses yang sama terhadap *Application Framework*. Apabila diperlukan, pengguna ataupun *vendor* dapat menggantikan aplikasi inti yang ada dengan aplikasi buatan sendiri yang memiliki fungsi yang sama. Hal ini membuat kustomisasi sistem operasi Android jauh lebih bebas dibandingkan sistem operasi perangkat *mobile* lainnya.

2.1.2 Versi Android

Sejak pertama kali muncul sampai sekarang, Android telah memiliki sejumlah pembaharuan. Pembaharuan ini dilakukan untuk memperbaiki *bug* dan menambah fitur-fitur yang baru. pada aplikasi ini minimal memakai versi yaitu. (Developers, 2014). Pembaruan utama terbaru versi Android adalah Lollipop 5.0, yang dirilis pada 3 November 2014. Lollipop adalah update Android paling besar dan ambisius dengan lebih dari 5.000 API baru untuk para *developer*. Perangkat yang menggunakan OS Android L ini akan mampu berintegrasi antar perangkat seperti *smartphone*, tablet dan *smartwatch* berbasis Android.

2.1.3 Android SDK

Android SDK adalah tools API (*Application Programming Interface*) yang diperlukan untuk mulai mengembangkan aplikasi pada platform Android

menggunakan bahasa pemrograman Java. Android merupakan *subset* perangkat lunak untuk ponsel yang meliputi sistem operasi, *middleware* dan aplikasi kunci yang akan di-*release* oleh Google. Saat ini disediakan Android SDK (*Software Development Kit*) sebagai alat bantu dan API untuk mulai mengembangkan aplikasi pada platform Android menggunakan bahasa pemrograman Java. Sebagai platform aplikasi netral, Android memberikan kita kesempatan untuk membuat aplikasi yang dibutuhkan yang bukan merupakan aplikasi bawaan *Handphone* atau *Smartphone* (Developers, 2014).

2.1.4 Eclipse

Eclipse adalah sebuah komunitas bagi individu dan organisasi yang ingin berkolaborasi secara *commercially-friendly* perangkat lunak bersifat *opensource*. Proyek perusahaan terfokus pada membangun sebuah platform pengembangan terbuka terdiri dari *extensible framework*, *tools* dan *runtimes* untuk membangun, menyebarkan dan mengelola perangkat lunak (Eclipse, 2014).

Berikut ini adalah sifat dari Eclipse.

1. *Multi-platform*: Target sistem operasi Eclipse adalah Microsoft Windows, Linux, Solaris, AIX, HP-UX dan Mac OS X.

2. *Mult-language*: Eclipse dikembangkan dengan bahasa pemrograman Java, akan tetapi Eclipse mendukung pengembangan aplikasi berbasis bahasa pemrograman lainnya, seperti C/C++, COBOL, Python, Perl, PHP, dan lain sebagainya.
3. *Multi-role*: Selain sebagai IDE untuk pengembangan aplikasi, Eclipse pun bisa digunakan untuk aktivitas dalam siklus pengembangan perangkat lunak, seperti dokumentasi, *test* perangkat lunak, pengembangan web, dan lain sebagainya.

Eclipse pada saat ini merupakan salah satu IDE favorit dikarenakan gratis dan *open source*, yang berarti setiap orang boleh melihat kode pemrograman perangkat lunak ini. Selain itu, kelebihan dari Eclipse yang membuatnya populer adalah kemampuannya untuk dapat dikembangkan oleh pengguna dengan komponen yang dinamakan *plugin*. Eclipse dibuat dari kerja sama antara perusahaan-perusahaan anggota 'Eclipse Foundation' (beserta individu-individu lain). Banyak nama besar yang ikut dalam 'Eclipse Foundation', termasuk IBM, BEA, Intel, Nokia, Borland. Eclipse bersaing langsung dengan Netbeans IDE. *Plugin* tambahan pada Eclipse jauh lebih banyak dan bervariasi dibandingkan IDE lainnya. Berikut ini merupakan *plugin* tambahan pada Eclipse ,

1. IntelliJ IDEA
2. Oracle JDeveloper
3. Xinox JCreator

2.1.5 Android Development Tools (ADT)

Android Development Tools adalah *plugin* yang didesain untuk IDE Eclipse yang memberikan kita kemudahan dalam mengembangkan aplikasi Android. Dengan adanya ADT untuk eclipse akan memudahkan *developer* dalam membuat aplikasi *project* Android, membuat aplikasi GUI, dan menambahkan komponen-komponen yang lainnya, begitu juga *user* dapat melakukan *running* aplikasi menggunakan Android SDK melalui Eclipse. Dengan ADT juga kita dapat membuat *package* Android (.apk) yang digunakan untuk distribusi aplikasi Android yang kita rancang (Developers, 2014).

2.1.6 Fundamental Aplikasi

Aplikasi Android ditulis dalam bahasa pemrograman Java, kode Java dikompilasi bersama dengan data *file resource* yang dibutuhkan oleh aplikasi di mana prosesnya di-*package* oleh *tools* yang dinamakan “*apt tools*” ke dalam paket Android sehingga menghasilkan *file* dengan ekstensi apk (*Android Package*). File apk itulah yang sebenarnya *user* sebut dengan aplikasi yang dapat diinstal di perangkat *mobile* nantinya. Ada empat jenis komponen pada aplikasi Android yaitu (Nazruddin, 2012).

1. *Activites*

Suatu *activity* akan menyajikan *User Interface* (UI) kepada pengguna, sehingga pengguna dapat melakukan interaksi. Sebuah aplikasi Android bisa

jadi hanya memiliki satu *activity*, tetapi umumnya aplikasi memiliki banyak *activity* tergantung pada tujuan aplikasi dan desain dari aplikasi tersebut. Satu *activity* biasanya akan dipakai untuk menampilkan aplikasi atau yang bertindak sebagai *user interface* saat aplikasi diperlihatkan kepada *user*. Untuk pindah dari satu *activity* ke *activity* yang lain *user* dapat melakukan dengan satu even misalnya klik tombol, memilih opsi atau menggunakan *triggers* tertentu. Secara hirarki sebuah windows *activity* dinyatakan dengan method `Activity setContentView()`. `ContentView` adalah objek yang berada pada *root* hirarki.

2. *Service*

Service tidak memiliki visual *user interface* (UI), tetapi *service* berjalan secara *background*, sebagai contoh dalam memainkan musik, *service* mungkin memainkan musik atau mengambil data dari jaringan, tetapi setiap *service* haruslah berada dalam kelas induknya. Misalnya media player sedang memutar lagu dari *list* yang ada, aplikasi ini akan memiliki dua atau lebih *activity* yang memungkinkan *user* untuk memilih lagu atau menulis SMS sambil *player* sedang jalan. Untuk menjaga musik tetap dijalankan, *activity player* dapat menjalankan *service* untuk membuat aplikasi tetap berjalan. *Service* dijalankan pada *thread* utama dari proses aplikasi.

3. *Broadcast Receiver*

Broadcast Receiver berfungsi menerima dan bereaksi untuk menyampaikan notifikasi. *Broadcast Receiver* tidak memiliki *user interface* (UI), tetapi memiliki sebuah *activity* untuk merespon informasi yang mereka terima, atau

mingin menggunakan *Notification Manager* untuk memberitahu kepada pengguna, seperti lampu latar atau *vibrating* (getaran) perangkat, dan lain sebagainya.

4. *Content Provider*

Content provider membuat kumpulan aplikasi data secara spesifik sehingga bisa digunakan oleh aplikasi lain. Data disimpan dalam *file system* seperti *database SQLite*. *Content Provider* menyediakan cara untuk mengakses data yang dibutuhkan oleh suatu *activity*. Misalnya ketika *user* menggunakan aplikasi yang membutuhkan peta atau aplikasi yang membutuhkan cara untuk mengakses data kontak untuk navigasi, maka disinilah fungsi *content provider*.

2.2 Global Positioning System

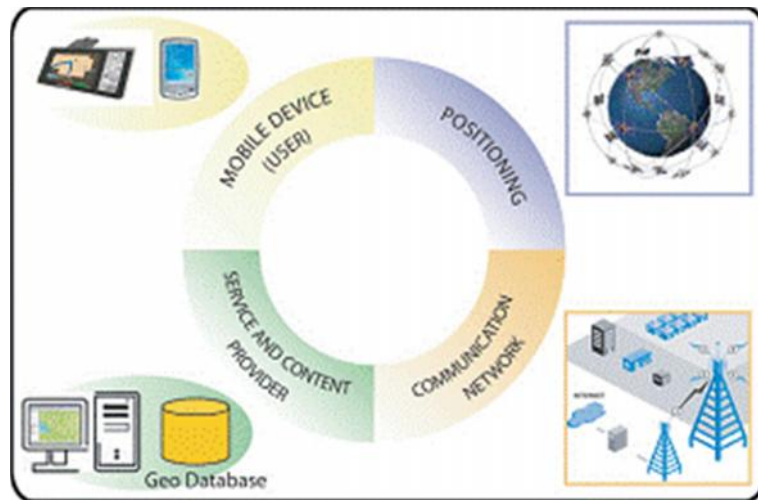
Global Positioning System (GPS) adalah suatu sistem radio navigasi penentuan posisi menggunakan satelit. GPS dapat memberikan posisi suatu objek di muka bumi dengan akurat dan cepat (koordinat tiga dimensi x, y, z) dan memberikan informasi waktu serta kecepatan bergerak secara kontinyu di seluruh dunia. (Lengkong, 2015).

2.3 Location Based Services

Location Based Services (LBS) adalah layanan informasi yang mengambil kemampuan untuk menggunakan informasi lokasi dari perangkat bergerak dan dapat diakses dengan perangkat bergerak melalui jaringan telekomunikasi

bergerak (Steiniger,2006). dalam layanan berbasis lokasi terdapat Lima komponen penting seperti terlihat pada Gambar 2.3. Setiap komponen mempunyai fungsi (Steiniger, 2006) .

1. *Mobile Devices*: suatu alat yang digunakan oleh pengguna untuk meminta informasi yang dibutuhkan. Informasi dapat diberikan dalam bentuk suara, gambar, dan teks.
2. *Communication Network*: komponen kedua adalah jaringan komunikasi yang mengirim data pengguna dan informasi yang diminta dari *Mobile terminal* ke *Service Provider* kemudian mengirimkan kembali informasi yang diminta ke pengguna. *Communication network* dapat berupa jaringan seluler (GSM, CDMA), *Wireless Local Area Network (WLAN)*, atau *Wireless Wide Area Network (WWAN)*.
3. *Positioning Component*: untuk memproses suatu layanan maka posisi pengguna harus diketahui.
4. *Service and Application Provider*: penyedia layanan menawarkan berbagai macam layanan kepada pengguna dan bertanggung jawab untuk memproses informasi yang diminta oleh pengguna.
5. *Data and Content Provider*: penyedia layanan tidak selalu menyimpan semua data yang dibutuhkan yang bisa diakses oleh pengguna. Untuk itu, data dapat diminta dari *data* dan *content provider*.



Gambar 2.2 Komponen Dasar LBS (Steiniger, 2006).

2.4 Point Of Interest

Point Of Interest (POI) merupakan sebuah tempat, produk atau layanan dengan sebuah lokasi tetap, khususnya diidentifikasi berdasarkan nama, alamat dan juga berdasarkan tipenya (Shekhar et al. 2004). Contoh utama dari POI seperti SPBU, restoran, hotel, *Automatic Teller Machine* (ATM) dan lain sebagainya. Contoh gambar dari *point of interest* disajikan pada Gambar 2.3.



Gambar 2.3 *Point of Interest* (Shekhar et al. 2004).

2.5 Google Maps API

Terdapat 2 pengertian di dalam Google Maps API yaitu pengertian tentang API dan juga pengertian tentang Google Maps itu sendiri.

2.5.1 Pengertian API

Application Programming Interface (API) bukan hanya satu *set class* dan *method* atau fungsi dan *signature* yang sederhana. API yang bertujuan utama untuk mengatasi “*clueless*” dalam membangun *software* yang berukuran besar, berawal dari sesuatu yang sederhana sampai ke yang kompleks dan merupakan perilaku komponen yang sulit dipahami (Halim, 2011).

2.5.2 Pengertian Google Maps API

Seperti yang tercatat oleh Svennerberg, Google Maps API adalah API yang paling populer di internet. Pencatatan yang dilakukan pada bulan Mei tahun 2010 ini menyatakan bahwa 43% mashup (aplikasi dan situs web yang menggabungkan dua atau lebih sumber data) menggunakan Google Maps API. Beberapa tujuan dari penggunaan Google Maps API adalah untuk melihat lokasi, mencari alamat, mendapatkan petunjuk mengemudi dan lain sebagainya (Halim, 2011).

2.6 Pengertian PHP

Menurut Didik Dwi Presetyo (2004 : 76), PHP merupakan bahasa scripting server

-side, dimana pemrosesan datanya dilakukan pada sisi server. Sederhananya, serverlah yang akan menerjemahkan skrip program, baru kemudian hasilnya akan dikirim kepada client yang melakukan permintaan.

2.6.1 Keunggulan PHP

Keunggulan PHP

Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan sistem *database* di dalam web. Kelebihan-kelebihan dari PHP diantaranya adalah :

- a. PHP mudah dibuat dan dijalankan, maksudnya PHP dapat berjalan dalam Web Server dan dalam Sistem Operasi yang berbeda pula.
- b. PHP adalah *software open-source* yang gratis dan bebas didistribusikan kembali di bawah lisensi GPL (*GNU Public License*). User dapat *download* kode-kode PHP tanpa harus mengeluarkan uang atau khawatir dituntut oleh pihak pencipta PHP.
- c. PHP bisa dioperasikan pada platform Linux ataupun Windows.
- d. PHP sangat efisien, karena PHP hanya memerlukan *resource system* yang sangat sedikit dibanding dengan bahasa pemrograman lain.
- e. Ada banyak Web Server yang mendukung PHP, seperti Apache, PWS, IIS, dan lain-lain.

- f. PHP juga didukung oleh banyak database, seperti MySQL, PostgreSQL, Interbase, SQL, dan lain-lain.
- g. Bahasa pemrograman PHP sintaknya sederhana, singkat dan mudah untuk dipahami.
- h. *HTML-embedded*, artinya PHP adalah bahasa yang dapat ditulis dengan menempelkan pada sintak-sintak HTML.

2.7 Pengertian MySQL

Menurut Didik Dwi Prasetyo (2004 :18) MySQL merupakan salah satu database server yang berkembang di lingkungan open source dan didistribusikan secara free (gratis) dibawah lisensi GPL. MySQL merupakan RDBMS (*Relational Database Management System*) server. RDBMS adalah program yang memungkinkan pengguna database untuk membuat, mengelola, dan menggunakan data pada suatu model relational. Dengan demikian, tabel-tabel yang ada pada database memiliki relasi antara satu tabel dengan tabel lainnya.

2.7.1 Keunggulan MySql

Beberapa keunggulan dari MySQL yaitu :

- a. Cepat, handal dan Mudah dalam penggunaannya MySQL lebih cepat tiga sampai empat kali dari pada *database server* komersial yang beredar saat ini, mudah diatur dan tidak memerlukan seseorang yang ahli untuk mengatur administrasi pemasangan MySQL.

- b. Didukung oleh berbagai bahasa *Database server* MySQL dapat memberikan pesan error dalam berbagai bahasa seperti Belanda, Portugis, Spanyol, Inggris, Perancis, Jerman, dan Italia.
- c. Mampu membuat tabel berukuran sangat besar 23.
- d. Ukuran maksimal dari setiap tabel yang dapat dibuat dengan MySQL adalah 4 GB sampai dengan ukuran file yang dapat ditangani oleh sistem operasi yang dipakai.
- e. Lebih Murah MySQL bersifat *open source* dan didistribusikan dengan gratis tanpa biaya untuk UNIX *platform*, OS/2 dan *Windows platform*.
- f. Melekatnya integrasi PHP dengan MySQL.
- g. Keterikatan antara PHP dengan MySQL yang sama-sama software *open-source* sangat kuat, sehingga koneksi yang terjadi lebih cepat jika dibandingkan dengan menggunakan *database server* lainnya. Modul MySQL di PHP telah dibuat *built-in* sehingga tidak memerlukan konfigurasi tambahan pada *file* konfigurasi *php.ini*.

2.8 JSON (Java Script Object Natation)

JSON adalah JSON (JavaScript Object Notation) adalah format pertukaran data (lightweight data interchange format), mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 –Desember 1999. JSON merupakan format teks yang tidak

bergantung pada bahasa pemrogramanapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.(Safaat, 2011).

2.9 Pengertian HTML

Menurut Anhar menjelaskan bahwa “Hypertext Markup Language (HTML) adalah sekumpulan simbol-simbol atau tag-tag yang dituliskan dalam sebuah file yang digunakan untuk menampilkan halaman pada web browser”.

Bermula dari sebuah bahasa yang sebelumnya banyak digunakan di dunia penerbitan dan percetakan yang disebut dengan SGML (Standard Generalized Markup Language), HTML adalah sebuah standar yang digunakan secara luas untuk menampilkan halaman web. HTML saat ini merupakan standar internet yang didefinisikan dan dikendalikan penggunaannya oleh World Wide Web Consortium (W3C). HTML pertama kali dibuat oleh kolaborasi Caillau TIM dengan Berners-Lee Robert ketika mereka bekerja di CERN, sebuah lembaga penelitian fisika energi tinggi di Jenewa-Swiss pada tahun 1989 .

HTML diawali dan diakhiri dengan suatu perintah yang sudah baku yang disebut dengan tag dan diantara kedua tag tersebut terdapat nilai-nilai dan atribut-atribut. HTML merupakan bahasa pemrograman yang fleksibel karena bisa meletakkan script dari bahasa pemrograman lain seperti PHP, Javascript, VB dan C .

Dokumen HTML adalah file teks murni yang dapat dibuat dengan sembarang editor (misalnya notepad). Dokumen HTML merupakan halaman situs yang dibaca browser dan berisikan informasi tentang halaman situs tersebut. Dokumen HTML harus disimpan dalam sebuah file dengan ekstensi “.html. Tidak seperti bahasa pemrograman lainnya, HTML tidak mengenal jumping ataupun looping. Script-script HTML dibaca oleh browser dari atas kebawah tanpa ada pengulangan ataupun slompatan .

Jika HTML tidak dapat mendukung suatu perintah pemrograman tertentu browser tidak akan menampilkan kotak dialog *syntax error*, jika terdapat penulisan kode yang keliru pada skrip HTML. *Hypertext* dalam HTML berarti bahwa kita dapat menuju ke suatu tempat, misalnya website atau halaman homepage lain, dengan cara memilih suatu link ke website atau homepage halaman lain, *hypertext* ini juga memungkinkan kita menuju ke salah satu bagian dalam suatu teks itu sendiri. Sedangkan *Markup Language* merupakan suatu fasilitas yang berupa tanda tertentu dalam skrip HTML dimana kita bisa merancang judul, garis, tabel, gambar dan lain-lain dengan perintah khusus. Adapun elemen pada HTML yang akan memisahkan dokumen menjadi beberapa bagian, antara lain:

- a. Elemen `<HEAD>...</HEAD>` : merupakan bagian untuk menuliskan keterangan mengenai judul dokumen, nama pengarang dan lain-lain.
- b. Elemen `<BODY>...</ BODY >` : merupakan bagian untuk menuliskan kode-kode Layout untuk tampilan homepage.

- c. Elemen `<FRAMESET >...</FRAMESET >` : merupakan bagian untuk meletakkan kode-kode frame .

2.10 Xampp

Menurut Wicaksono (2008:7) menjelaskan bahwa “XAMPP adalah sebuah software yang berfungsi untuk menjalankan website berbasis PHP dan menggunakan pengolah data MYSQL di komputer lokal”. XAMPP berperan sebagai *server* web pada komputer lokal. XAMPP juga dapat disebut sebuah Cpanel server virtual, yang dapat membantu melakukan *preview* sehingga dapat dimodifikasi website tanpa harus online atau terakses dengan internet.

2.11 Bengkel

Pada kondisi tertentu, kendaraan bermotor memerlukan perawatan atau perbaikan. Perawatan dan perbaikan kendaraan harus dilakukan agar umur pakai kendaraan lebih panjang atau paling tidak sama dengan umur pakai yang telah diprediksikan dan dirancang oleh pabrik pembuat. Meskipun demikian, perawatan dan perbaikan kendaraan bukan merupakan pekerjaan yang mudah. Hal tersebut memerlukan pengetahuan khusus.

Untuk memperoleh pengetahuan tersebut, tentu saja dibutuhkan kemauan dan waktu. Namun sebagian besar pemilik kendaraan bermotor biasanya merasa dirinya tidak memiliki kedua hal tersebut. Berdasarkan hal tersebut, terbuka

peluang bagi pihak lain yang memiliki keahlian dan peralatan kerja di bidang kendaraan bermotor (otomotif) untuk membuka usaha perbengkelan. Terjadilah transaksi antara orang yang membutuhkan perawatan atau perbaikan di bidang otomotif dan mereka yang memiliki keahlian serta peralatan di bidang tersebut. Hal ini dilakukan di bengkel otomotif.

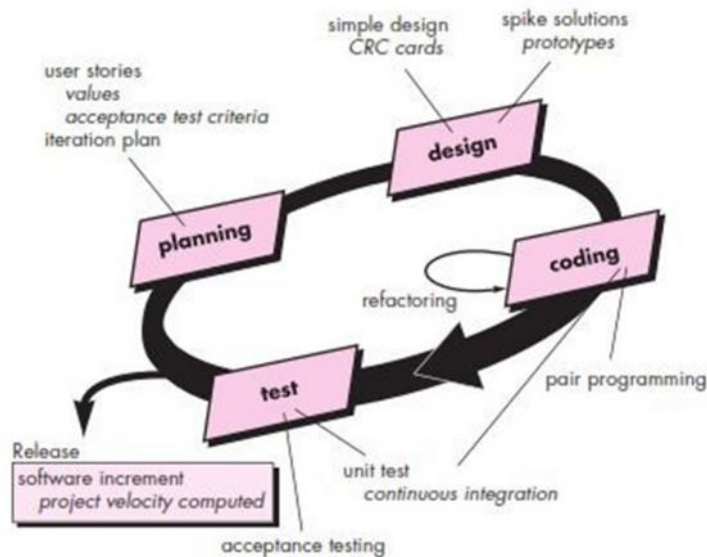
2.12 Metodologi Pengembangan Sistem

Adapun metodologi yang digunakan dalam pembuatan aplikasi layanan lokasi bengkel mobil terdekat yang ada di Kota Bandar Lampung ini menggunakan teknologi *location based service* berbasis android yang meliputi: *Unified Extreme Programming* (XP) dan desain menggunakan *Unified Modeling Language* (UML). Berikut merupakan penjelasan lebih rinci dari masing-masing metodologi.

2.12.1 *Extreme Programming* (XP)

Extreme Programming (XP) merupakan suatu pendekatan yang paling banyak digunakan untuk pengembangan perangkat lunak cepat. Alasan menggunakan metode *Extreme Programming* (XP) karena sifat dari aplikasi yang dikembangkan dengan cepat melalui tahapan-tahapan yang ada meliputi: *Planning*/Perencanaan, *Design*/Perancangan, *Coding*/Pengkodean dan *Testing*/Pengujian. (Pressman, 2010).

Adapun tahapan pada *Extreme Programming* dapat dijelaskan sebagai berikut:



Gambar 2. 4 Siklus *Extreme Programming* (Pressman, 2010).

Siklus *Extreme Programming* membagi sebuah proyek menjadi 4 fase besar (Pressman, 2010).

1. *Planning*/Perencanaan

Pada tahap perencanaan ini dimulai dari pengumpulan kebutuhan yang membantu tim teknis untuk memahami konteks bisnis dari sebuah aplikasi. Selain itu pada tahap ini juga mendefinisikan *output* yang akan dihasilkan, fitur yang dimiliki oleh aplikasi dan fungsi dari aplikasi yang dikembangkan.

2. *Design*/Perancangan

Metode ini menekankan desain aplikasi yang sederhana, untuk mendesain aplikasi dapat menggunakan *Class-Responsibility-Collaborator* (CRC) *cards* yang mengidentifikasi dan mengatur *class* pada *object-oriented*.

3. *Coding*/Pengkodean

Konsep utama dari tahapan pengkodean pada *extreme programming* adalah *pair programming*, melibatkan lebih dari satu orang untuk menyusun kode.

4. *Testing*/Pengujian

Pada tahapan ini lebih fokus pada pengujian fitur dan fungsionalitas dari aplikasi.

2.12.2. *Unified Modeling Language* (UML)

Unified Modeling Language (UML) adalah keluarga notasi grafis yang didukung oleh meta-model tunggal, yang membantu pendeskripsian dan desain sistem perangkat lunak, khususnya sistem yang dibangun menggunakan pemrograman berorientasi objek (OO). Definisi ini merupakan definisi yang sederhana. Pada kenyataannya, pendapat orang – orang tentang UML berbeda satu sama lain. Hal ini dikarenakan oleh sejarahnya sendiri dan oleh perbedaan persepsi tentang apa yang membuat sebuah proses rancang – bangun perangkat lunak efektif.

Unified Modeling Language (UML) merupakan standar yang relatif terbuka yang dikontrol oleh *Object Management Group* (OMG), sebuah konsorsium terbuka yang terdiri dari banyak perusahaan. OMG dibentuk untuk membuat


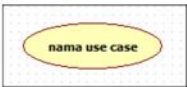
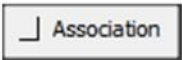



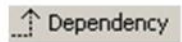

standar yang mendukung interoperabilitas, khususnya interoperabilitas *system* berorientasi objek. OMG mungkin lebih dikenal dengan standar – standar COBRA (*Common Object Request Broker Architecture*).

UML lahir dari penggabungan banyak bahasa permodelan grafis berorientasi objek yang berkembang pesat pada akhir 1980-an dan awal 1990-an. UML dibuat oleh Grady Booch, James Rumbaugh, dan Ivar Jacobson di bawah bendera *Rational Software Corp.* UML menyediakan notasi - notasi yang membantu memodelkan sistem dari berbagai perspektif. UML tidak hanya digunakan dalam pemodelan perangkat lunak, namun hampir dalam semua bidang yang membutuhkan pemodelan (Fowler, 2004) UML dideskripsikan oleh beberapa diagram, yaitu sebagai berikut.

1. Use Case Diagram

Use case diagram dapat digunakan selama proses analisis untuk menangkap *requirements* sistem dan untuk memahami bagaimana sistem seharusnya bekerja. Selama tahap desain, *use case diagram* menetapkan perilaku (*behavior*) sistem saat diimplementasikan. Dalam sebuah model mungkin terdapat satu atau beberapa *use case diagram*. Contoh diagram *use case* disajikan dalam Gambar 2.1.






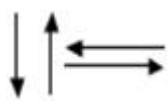
Tabel 2.1 simbol *use case diagram*

No	Simbol	Nama	Keterangan
1		<i>Actor</i>	Merupakan orang, proses atau sistem lain yang berinteraksi dengan sistem yang akan dibuat. Jadi walaupun simbol aktor dalam diagram usecase berbentuk orang, namun aktor belum tentu orang
2		<i>Use case</i>	Merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling berinteraksi atau bertukar pesan antar unit maupun aktor..
3		<i>Association</i>	Merupakan relasi yang digunakan untuk menggambarkan interaksi antara usecase dan aktor. Asosiasi juga menggambarkan berapa banyak objek lain yang bisa berinteraksi dengan suatu objek atau disebut multiplicity .
4		<i>Extend</i>	Menghubungkan antara satu objek dengan objek lain.
5		<i>generalization</i>	Hubungan dimana objek berbagai perilaku dan struktur data dari objek yang ada di atasnya objek induk
6		<i>Include</i>	Menspesifikasi bahwa <i>use case</i> sumber secara eksplisit
7		<i>Dependency</i>	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri
8		<i>System boundary</i>	Menspesifikasikan paket yang menampilkan sistem secara terbatas.

2. Activity Diagram

Activity diagram memodelkan alur kerja (*work flow*) sebuah proses bisnis dan urutan aktivitas dalam suatu proses. Diagram ini sangat mirip dengan sebuah *flowchart* karena *user* dapat memodelkan sebuah alur kerja dari satu aktivitas ke aktivitas lainnya atau dari satu aktivitas ke dalam keadaan sesaat (*state*). Beberapa simbol - simbol yang ada pada diagram *activity diagram* disajikan dalam tabel 2.2..

Tabel 2.2 simbol *activity diagram*


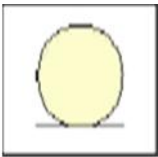
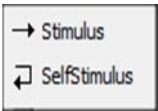
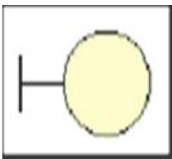
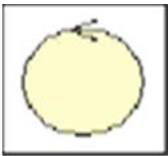
No	Gambar	Nama	Keterangan
1		<i>Activity</i>	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		<i>Action</i>	State dari sistem yang mencerminkan eksekusi dari suatu aksi .
3		<i>Initial node</i>	Bagaimana objek dibentuk atau diawali
4		<i>Activity final node</i>	Objek dibentuk untuk diakhiri
5		<i>Decesion</i>	Digunakan untuk menggambarkan suatu keputusan / tindakan yang harus diambil pada kondisi tertentu
6		<i>Line connector</i>	Digunakan untuk menghubungkan suatu symbol dengan simbol lainnya

3. Sequence Diagram



Diagram sekuensial atau *sequence diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *use case*. Misalkan, pada *use case* “menarik uang”

mempunyai beberapa kemungkinan, seperti penarikan uang secara normal, percobaan penarikan uang tanpa kecukupan ketersediaan dana, penarikan dengan penggunaan PIN yang salah, dan lainnya (Sholiq, 2006). Contoh diagram sekuensial disajikan pada tabel 2.3.

Tabel 2.3 simbol *sequence diagram*

No	Simbol	Nama	keterangan
1		Aktor	Menggambarkan orang yang berinteraksi dengan sistem.
2		Entity class	Menggambarkan hubungan kegiatan yang akan dilakukan.
3		Stimulus	Menyatakan suatu objek mengirimkan pesan untuk menjalankan oprasi yang ada pada objek lain.
4		Boundry class	Menggambarkan sebuah penggambaran dari form.
5		Control class	Menggambarkan penghubung antara boundary dengan tabel.

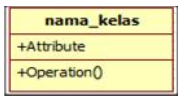

Lanjutan tabel 2.3 simbol *sequence diagram*

6		Lifeline	Menggambarkan tempat mulai dan berakhirnya sebuah pesan.
7		Line Message	Menggambarkan pengiriman pesan.



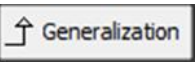


4. Class Diagram

Class adalah sebuah spesifikasi yang akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* menggambarkan keadaan (*atribut/property*) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (*metode/fungsi*). *Class Diagram* menggambarkan struktur dan deskripsi *class*, *package* dan objek beserta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain (Fowler, 2004). Bentuk dari *class diagram* dapat terlihat pada tabel 2.4.

Tabel 2.4 simbol *class diagram*

No	Simbol	Nama	Keterangan
1		Kelas	Kelas pada stuktur sistem.
2		Antarmuka (Interface)	Sama dengan konsep interface dalam pemrograman berorientasi objek.

Lanjutan Tabel 2.4 *Class Diagram*

No	Simbol	Nama	Keterangan
3	 Association	Asosiasi (Association)	Relasi antar kelas dengan makna umum, asosiasi biasanya juga di sertai dengan multiplicity.
4	 DirectedAssociation	Asosiasi berarah (Directed Association)	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi berarah biasanya juga disertai dengan multiplicity.
5	 Generalization	Generalisasi (Generalization)	Relasi antar kelas dengan makna generalisasi-spesialisasi (Umum-khusus)
6	 Dependency	Kebergantungan (Dependency)	Relasi antar kelas dengan makna kebergantungan antar kelas.
7	 Aggregation	Agregasi (Aggregation)	Relasi antar kelas dengan makna semua-bagian (Whole-part)

Class memiliki tiga area pokok.

1. Nama (*Class Name*)
2. Atribut
3. Metode (*Operations*)

Pada UML, *class* digambarkan dengan segiempat yang dibagi beberapa bagian. Bagian atas merupakan nama dari *class*. Bagian yang tengah merupakan struktur dari *class* (atribut) dan bagian bawah merupakan sifat dari *class* (metode/operasi). Atribut dan metode dapat memiliki salah satu sifat berikut (Fowler, 2004).

1. *Private*, tidak dapat dipanggil dari luar *class* yang bersangkutan.
2. *Protected*, hanya dapat dipanggil oleh *class* yang bersangkutan dan *class* lain yang mewarisinya.
3. *Public*, dapat dipanggil oleh *class* lain.

5. Hubungan antar *class*

Asosiasi, yaitu hubungan statis antar *class*. Umumnya menggambarkan *class* yang memiliki atribut berupa *class* lain, atau *class* yang harus mengetahui eksistensi *class* lain (Fowler, 2004). Berikut ini merupakan atribut dari suatu *class*.

1. Agregasi, yaitu hubungan yang menyatakan bagian (“terdiri atas”).
2. Pewarisan, yaitu hubungan hirarki antar *class*. *Class* dapat diturunkan

dari *class* lain dan mewarisi semua atribut dan metode *class* asalnya serta bisa menambahkan fungsionalitas baru. Sehingga *class* tersebut disebut anak dari *class* yang diwarisinya.

3. Hubungan dinamis, yaitu rangkaian pesan (*message*) yang di *class*, dari satu *class* kepada *class* lain. Hubungan dinamis dapat digambarkan dengan menggunakan *sequence diagram* yang akan dijelaskan kemudian.

2.13 Pengujian Perangkat Lunak

Pengujian perangkat lunak adalah proses menjalankan dan mengevaluasi sebuah perangkat lunak secara manual maupun otomatis untuk menguji apakah perangkat lunak sudah memenuhi persyaratan atau belum (Clune, 2011). Singkat kata, pengujian adalah aktivitas untuk menemukan dan menentukan perbedaan antara hasil yang diharapkan dengan hasil sebenarnya. Berikut merupakan penjelasan lebih rinci dari pengujian perangkat lunak baik itu teknik maupun metode yang digunakan di dalam pengujian perangkat lunak.

Ada dua macam pendekatan kasus uji yaitu *white-box* dan *black-box*. Pendekatan *white-box* adalah pengujian untuk memperlihatkan cara kerja dari produk secara rinci sesuai dengan spesifikasinya (Jiang, 2012). Jalur logika perangkat lunak akan di tes dengan menyediakan kasus uji yang akan mengerjakan kumpulan kondisi dan pengulangan secara spesifik. Sehingga melalui penggunaan metode ini akan dapat memperoleh kasus uji yang

menjamin bahwa semua jalur independen pada suatu model telah digunakan minimal satu kali, penggunaan keputusan logis pada sisi benar dan salah, pengeksekusian semua *loop* dalam batasan dan batas operasional rekayasa, serta penggunaan struktur data internal guna menjamin validitasnya (Pressman, 2010). Pendekatan *black-box* merupakan pendekatan pengujian untuk mengetahui apakah semua fungsi perangkat lunak telah berjalan semestinya sesuai dengan kebutuhan fungsional yang telah didefinisikan (Jiang, 2012). Kasus uji ini bertujuan untuk menunjukkan fungsi perangkat lunak tentang cara beroperasinya. Teknik pengujian ini berfokus pada domain informasi dari perangkat lunak, yaitu melakukan kasus uji dengan mempartisi domain *input* dan *output* program. Metode *black-box* memungkinkan rekayasa perangkat lunak mendapatkan serangkaian kondisi *input* yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini berusaha menemukan kesalahan dalam kategori fungsi - fungsi yang tidak benar atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses basis data eksternal, kesalahan kinerja, dan inisialisasi dan kesalahan terminal (Pressman, 2010).

2.13.1 Black-Box

Pengujian *black-box* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *black-box* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *black-box* bukan

merupakan alternative dari teknik *white-box*, tetapi merupakan pendekatan komplementer yang kemungkinan besar mampu mengungkap kelas kesalahan daripada metode *white-box*.

Pengujian *black-box* berusaha menemukan kesalahan dalam kategori sebagai berikut.

1. Fungsi–fungsi yang tidak benar atau hilang,
2. Kesalahan Interface,
3. Kesalahan dalam struktur data atau akses database eksternal,
4. Kesalahan Kinerja,
5. Inisialisasi dan kesalahan terminasi.

Tidak seperti pengujian *white-box*, yang dilakukan pada saat awal proses pengujian, pengujian *black-box* cenderung diaplikasikan selama tahap akhir pengujian. Karena pengujian *black-box* memperhatikan struktur control, maka perhatian berfokus pada domain informasi. Pengujian di desain untuk menjawab pertanyaan - pertanyaan berikut .

1. Bagaimana validasi fungsional diuji?
2. Kelas *input* apa yang akan membuat *test case* menjadi baik?
3. Apakah sistem sangat *sensitive* terhadap harga input tertentu?
4. Bagaimana batasan dari suatu data di isolasi?
5. Kecepatan data apa dan *volume* data apa yang akan dapat di tolerir oleh sistem?

6. Apa pengaruh kombinasi tertentu dari data terhadap operasi sistem?

Dengan mengaplikasikan teknik *black-box*, maka *user* menarik serangkaian *test case* yang memenuhi *criteria* berikut ini :

1. *Test case* yang mengurangi, dengan harga lebih dari satu, jumlah *test case* tambahan yang harus didesain untuk mencapai pengujian yang dapat dipertanggungjawabkan,

2. *Test case* yang *member* tahu kesalahan yang berhubungan hanya dengan pengujian spesifik yang ada (Pressman, 2002).

BAB III

METODOLOGI PENELITIAN

3.1 Waktu dan Tempat Penelitian

Penelitian ini dilakukan di Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Lampung yang berada di jalan Soemantri Brojonegoro No.1 Gedung Meneng, Bandar Lampung. Penelitian ini dilaksanakan mulai dari semester genap tahun ajaran 2015/2016 sampai dengan semester ganjil tahun ajaran 2015/2016.

3.2 Alat Pendukung Penelitian

Penelitian ini dilakukan dengan menggunakan alat untuk mendukung dan menunjang pelaksanaan penelitian.

1. Perangkat Keras

Perangkat keras yang digunakan pada pengembangan sistem ini yaitu *laptop* dengan spesifikasi. Berikut ini spesifikasi perangkat keras yang digunakan dalam melakukan penelitian.

- a. Prosessor Intel Core i3-2310M CPU @ 2,1 GHz
- b. VGA Intel HD Graphic 3000

- c. RAM 2GB
- d. Hard disk 500GB

2. Perangkat Lunak

Perangkat lunak yang digunakan dalam pengembangan sistem ini yaitu.

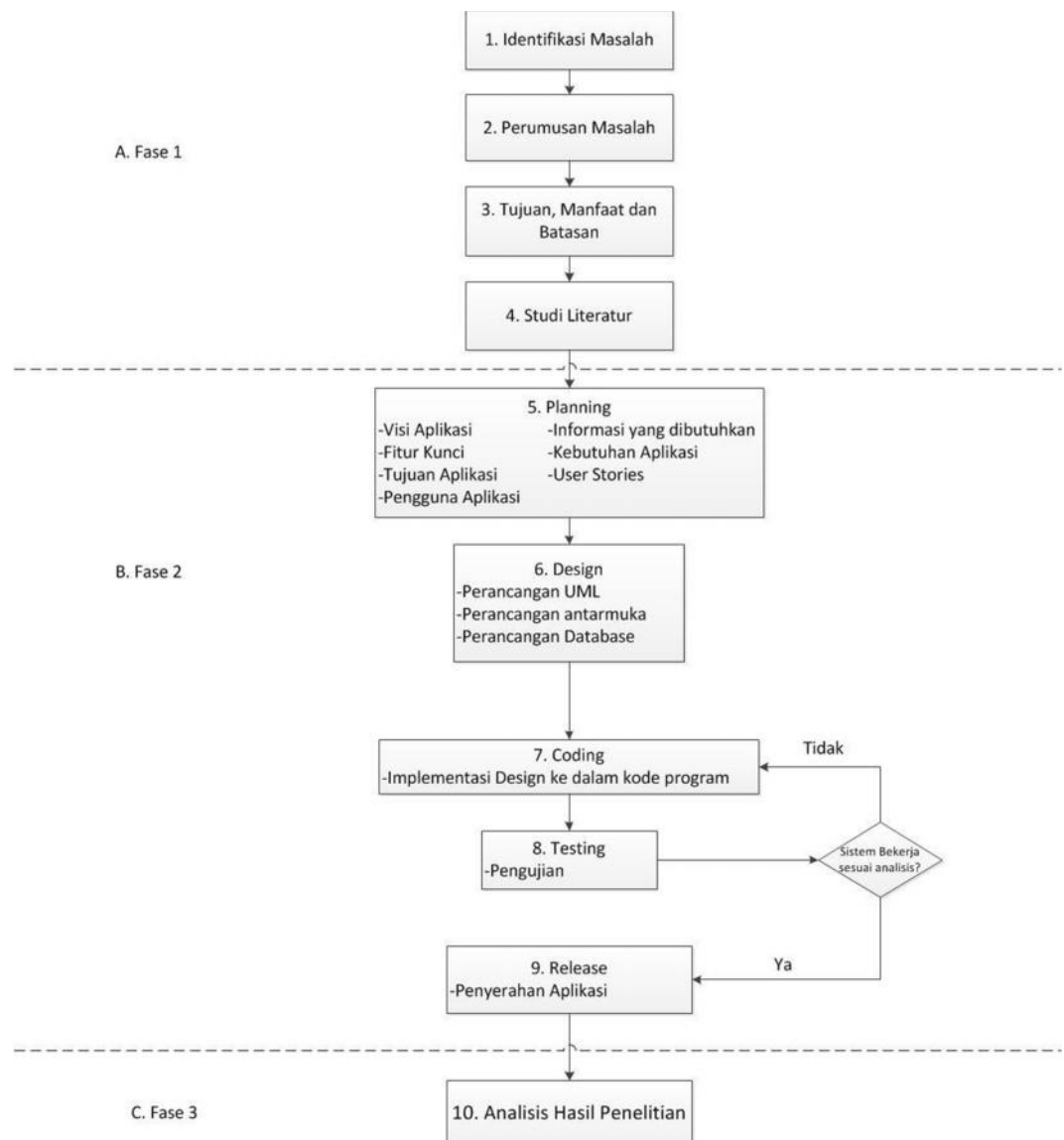
- a. Sistem Operasi Windows 7 (64 Bit),
- b. Eclipse
- c. Paint Version 6.3
- e. Microsoft Picture Manager
- f. Adobe Photo Shop CS6
- g. *Web Browser* (Google Chrome)

3.3 Metodologi Penelitian

Pada tahapan metodologi penelitian ini terdapat beberapa tahapan, seperti alir penelitian, metode pengembangan sistem, jadwal kegiatan penelitian, metode pengumpulan data, serta jenis dan sumber data. Berikut merupakan penjelasan secara rinci dari masing-masing tahapan tersebut.

3.3.1 Diagram Alir Metodologi Penelitian

Penelitian dilakukan berdasarkan diagram alir metodologi penelitian yang terdapat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Penelitian

Penjelasan dari diagram alir penelitian metodologi penelitian pada Gambar 3.1 adalah sebagai berikut.

A. Fase 1

1. Tahap pertama pada penelitian ini adalah mengidentifikasi masalah, yaitu menentukan masalah yang akan dipecahkan dalam penelitian.
2. Tahap ke-dua adalah perumusan masalah, yaitu pertanyaan penelitian yang membutuhkan jawaban melalui pengumpulan data yang umumnya disusun dalam bentuk kalimat tanya.
3. Tahap ke-tiga adalah menentukan tujuan, manfaat, dan batasan masalah, yaitu penentuan tujuan dan manfaat dari penelitian yang dilakukan. Serta batasan digunakan untuk membatasi pembahasan dan ruang lingkup penelitian.
4. Tahap ke-empat adalah studi literature, yaitu memperoleh teori-teori dan hasil-hasil penelitian terlebih dahulu yang akan dijadikan sebagai landasan penelitian.

B. Fase 2

5. Tahap ke-lima dalam fase ke-dua yaitu pengembangan sistem. Pada penelitian ini, metode yang digunakan adalah *eXtreme Programming* (XP). Tahap pertama yang dilakukan dalam pengembangan sistem ini adalah *planning*. Pada tahap ini dilakukan pengumpulan data penelitian, seperti jumlah objek wisata yang ada di Kabupaten Tanggamus, fitur kunci dari aplikasi, tujuan aplikasi, kebutuhan aplikasi.

6. Tahap ke-enam adalah design, yaitu perancangan sistem dan perancangan antarmuka yang menggunakan *Unified Modelling Language* (UML) serta perancangan basis data dari sistem yang akan dibangun. Pada tahap ini akan diketahui semua entitas luar, input dan output yang terlibat dalam sistem serta *usecase*, *class diagram*, *activity diagram*, *sequence diagram* yang digunakan dalam analisis sistem.

7. Selanjutnya akan dilakukan perancangan antarmuka (*interface*), perancangan antarmuka ini dilakukan untuk merancang tata letak sistem sesuai dengan analisis kebutuhan sistem.

Tahap ke-tujuh adalah *coding*, yaitu tahap pembuatan program aplikasi.

8. Tahap ke-delapan adalah *testing*, yaitu tahap pengujian program.

9. Program yang telah selesai dibuat kemudian diuji dengan metode pengujian perangkat lunak *Black Box*. Apabila program aplikasi tidak berjalan sesuai dengan analisis maka akan kembali ke tahap *coding*.

Namun, apabila telah berjalan maka akan ke tahap selanjutnya

Tahap ke-sembilan adalah *release*, yaitu penyerahan aplikasi kepada pengguna melalui *Play Store*.

10. Tahap ke-sepuluh adalah analisis hasil penelitian, yaitu penjelasan apakah suatu penelitian telah berhasil memecahkan permasalahan yang telah dideskripsikan di awal.

3.3.2 Metode Pengembangan Sistem

Penelitian dilakukan berdasarkan metode pengembangan sistem yang dipilih yaitu *eXtreme Programming* (XP). Tahap-tahap yang dilakukan dalam Pengembangan Sistem Informasi Geografis Objek Wisata di Kabupaten Tanggamus ini adalah sebagai berikut.

1. *Planning*

Dalam tahap ini dikumpulkan kebutuhan awal user atau dalam XP disebut user stories. Hal ini dibutuhkan agar pengembang mengerti bisnis konten, kebutuhan output sistem, dan fitur utama dari software yang dikembangkan.

2. *Design*

Desain dari sistem pada penelitian ini digambarkan dengan model UML berupa *use case* diagram, *activity* diagram, dan relation table. Pembuatan desain pada XP tetap mengedepankan prinsip *Keep it Simple* (KIS). Desain disini merupakan representasi dari sistem guna mempermudah pengembang dalam membangun sistem.

3. *Coding*

Pada tahap ini dibangun aplikasi yang menjadi objek penelitian. Aplikasi dibangun berdasarkan desain yang telah dibuat pada tahap sebelumnya. Dalam pembangunan aplikasi ini menggunakan bahasa pemrograman Java, serta XML untuk tampilan pada Android. Pada tahap *coding* juga disisipkan tahap refactoring. Refactoring adalah proses mengubah sistem

perangkat lunak sedemikian rupa dengan tidak mengubah eksternal kode dalam memperbaiki struktur internalnya. Hal ini dilakukan untuk meminimalkan kemungkinan adanya bug.

4. *Testing*

Pada tahap ini aplikasi yang telah dibangun diuji apakah kebutuhan awal user atau user stories sudah dipenuhi dan apabila terpenuhi aplikasi siap di rilis.

3.3.3 Metode Pengumpulan Data

Metode pengumpulan data merupakan teknik atau cara yang dilakukan untuk mengumpulkan data, adapun metode pengumpulan data yang digunakan dalam penelitian ini, antara lain.

1. Studi Literatur

Studi literatur yang digunakan yaitu buku-buku, jurnal, prosiding dan internet yang menyajikan informasi tentang, *Eclipse*, Android dan bahasa pemrograman Java.

2. Metode Spasial

Metode ini digunakan untuk mendapatkan informasi pengamatan yang dipengaruhi efek ruang atau lokasi. Pengaruh efek ruang tersebut disajikan dalam bentuk koordinat lokasi (*longitude, latitude*).

3.4 Analisis dan Perancangan Sistem

Pada tahapan analisis dan perancangan sistem ini terdapat beberapa tahapan seperti identifikasi masalah, analisis sistem, analisis user requirement, perancangan sistem, dan juga metode pengujian sistem.

3.4.1 Identifikasi Masalah

Setiap orang membutuhkan kemudahan didalam melakukan berbagai macam hal, tidak terkecuali bengkel. Bagi pelanggan kendaraan bermobil yang mengalami kerusakan mesin dll, Bengkel merupakan elemen penting agar kendaraan tersebut bisa melaju sampai tujuan. Saat ini belum terdapat sebuah aplikasi yang dapat memudahkan pengendara untuk mencari dan menemukan dimana lokasi bengkel resmi mobil terdekat dari tempat pengendara berada dan bisa langsung datang montir dari bengkel tersebut. Orang akan kesulitan untuk mencari dimana lokasi bengkel terdekat dari tempat berada apalagi jika orang itu merupakan turis ataupun wisatawan yang sedang berkunjung dan juga transit di suatu daerah. Oleh karena itu saat ini dibutuhkan suatu aplikasi yang dapat memudahkan pelanggan untuk melakukan pencarian lokasi bengkel resmi mobil dan bisa memanggil montir mobil untuk langsung datang ke lokasi pelanggan, dan diharapkan aplikasi ini dapat mempermudah pelanggan untuk mencari letak bengkel terdekat dan memanggil montir ke lokasi, sehingga tidak ada lagi pengendara yang mogok di jalan karena tidak tahu letak bengkel resmi mobil.

3.4.2 Analisis Sistem

Berdasarkan identifikasi masalah di atas, maka sistem yang dibutuhkan adalah sebagai berikut.

1. Menggunakan perangkat android dan web.
2. Aplikasi terdapat super admin, admin, bengkel, pelanggan.
3. Peta atau rute-rute lokasi bengkel di Kota Bandar Lampung yang dapat diakses menggunakan perangkat android dan web.

3.4.3 Analisis *User Requirement*

Kebutuhan dasar aplikasi ini adalah sebagai berikut.

1. Sistem dapat menampilkan peta atau rute-rute lokasi bengkel di Kota Bandar Lampung menggunakan perangkat android.
2. Sistem dari super admin dapat menampilkan menu super admin berupa informasi data seluruh bengkel
3. Sistem dari admin dapat menampilkan menu admin berupa informasi data bengkel.
4. Sistem dari pelanggan dapat menampilkan isi perbaikan kerusakan yang langsung terhubung ke bengkel.

5. Sistem dari bengkel dapat menampilkan letak pelanggan dan menerima isi perbaikan kerusakan dan langsung mengirim montir ke lokasi.

3.4.4 Perancangan Sistem

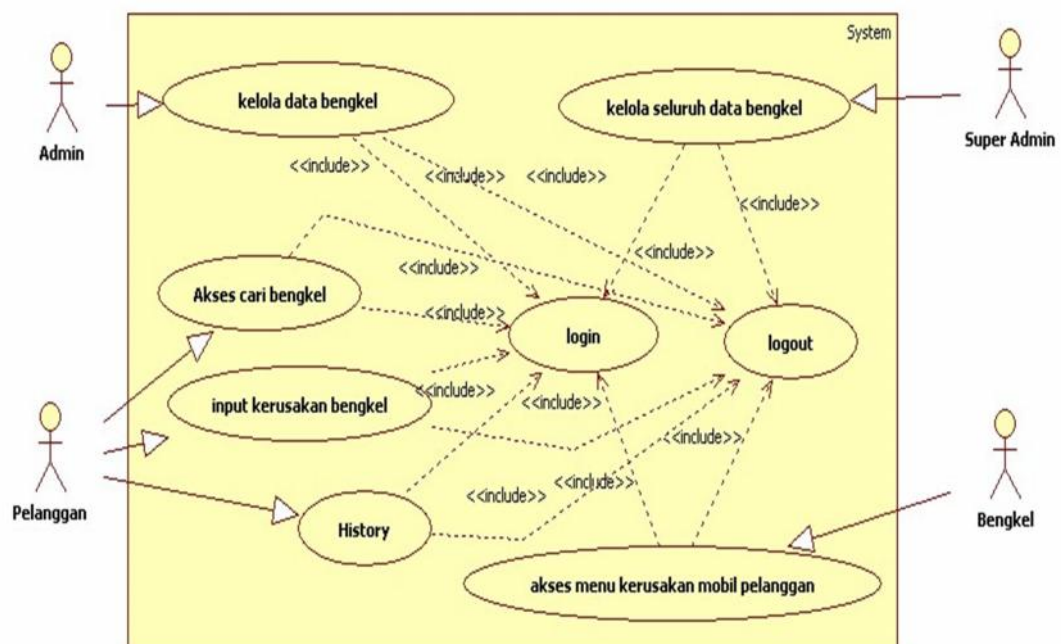
Perancangan sistem adalah tahap setelah analisis dari siklus pengembangan sistem. Perancangan sistem di sini berupa penggambaran, perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah ke dalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem menentukan bagaimana suatu sistem akan menyelesaikan apa yang mesti diselesaikan. Tahap ini termasuk mengkonfigurasi dari komponen-komponen perangkat lunak dan perangkat keras dari suatu sistem sehingga setelah dilakukan instalasi akan benar-benar memuaskan rancang bangun yang telah ditetapkan pada akhir tahap analisis sistem. Berikut ini merupakan tahapan-tahapan yang ada di dalam perancangan sistem.

3.4.4.1 Perancangan UML (*Unified Modelling Language*)

Pemodelan (*modelling*) adalah tahap merancang perangkat lunak sebelum melakukan tahap pembuatan program (*coding*). Pada penelitian ini, perancangan sistem dilakukan dengan memodelkan permasalahan dalam bentuk diagram - diagram UML.

1. Use Case Diagram

Use case Diagram dibawah ini menggambarkan sistem dari sudut pandang pelanggan, super admin, bengkel, admin, sehingga pembuatan use case diagram ini lebih dititikberatkan pada fungsionalitas yang ada pada sistem, bukan berdasarkan alur atau urutan kejadian. Pada aplikasi ini terdapat 5 interaksi antara lain “Cari bengkel”, “Isi keterangan kerusakan mobil”, “History”, “Kelola data bengkel”, “Kelola data seluruh bengkel” dan juga “akses kerusakan mobil pelanggan” . Use case diagram aplikasi Bengkel resmi mobil dapat dilihat pada Gambar 3.2.



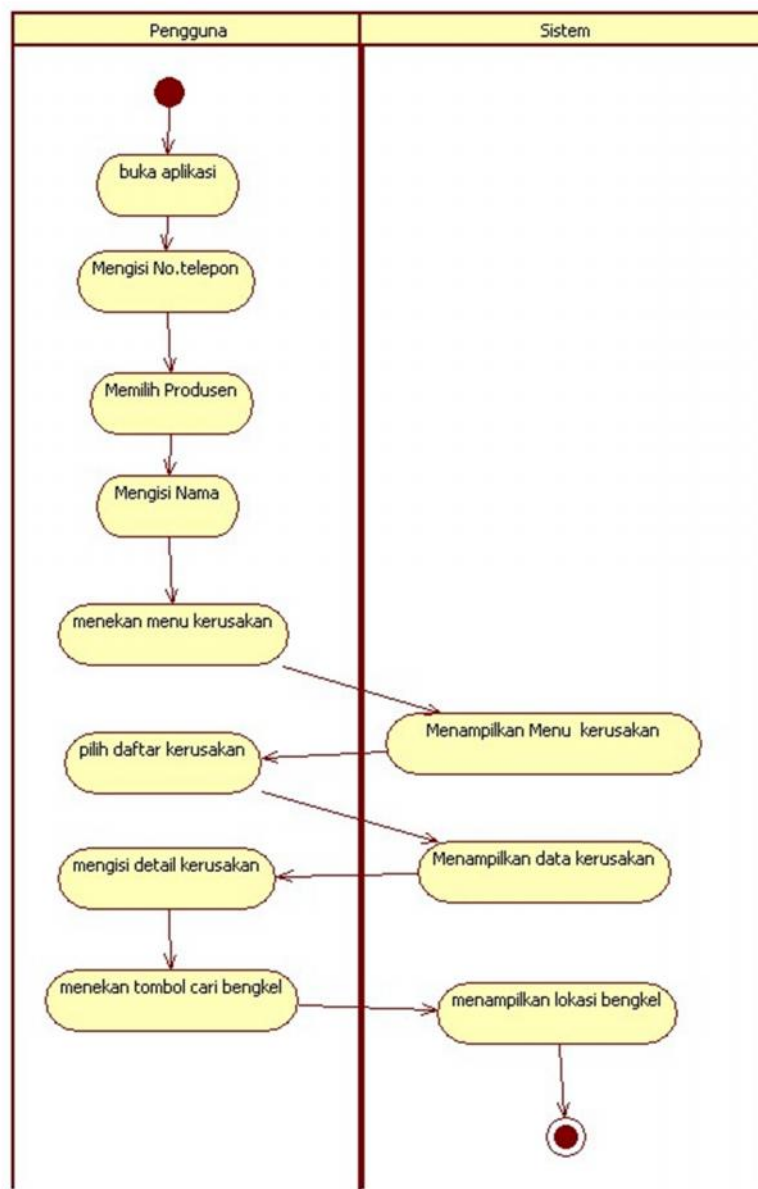
Gambar 3.2 Use Case Diagram

2. Activity Diagram

Activity diagram menggambarkan rangkaian aliran dari aktivitas, digunakan untuk mendeskripsikan aktivitas yang dibentuk dalam satu operasi sehingga dapat juga untuk aktivitas lainnya. Pada aplikasi Bengkel resmi mobil terdapat 8 (delapan) *activity* diagram yaitu sebagai berikut .

1. Activity Diagram Cari Bengkel

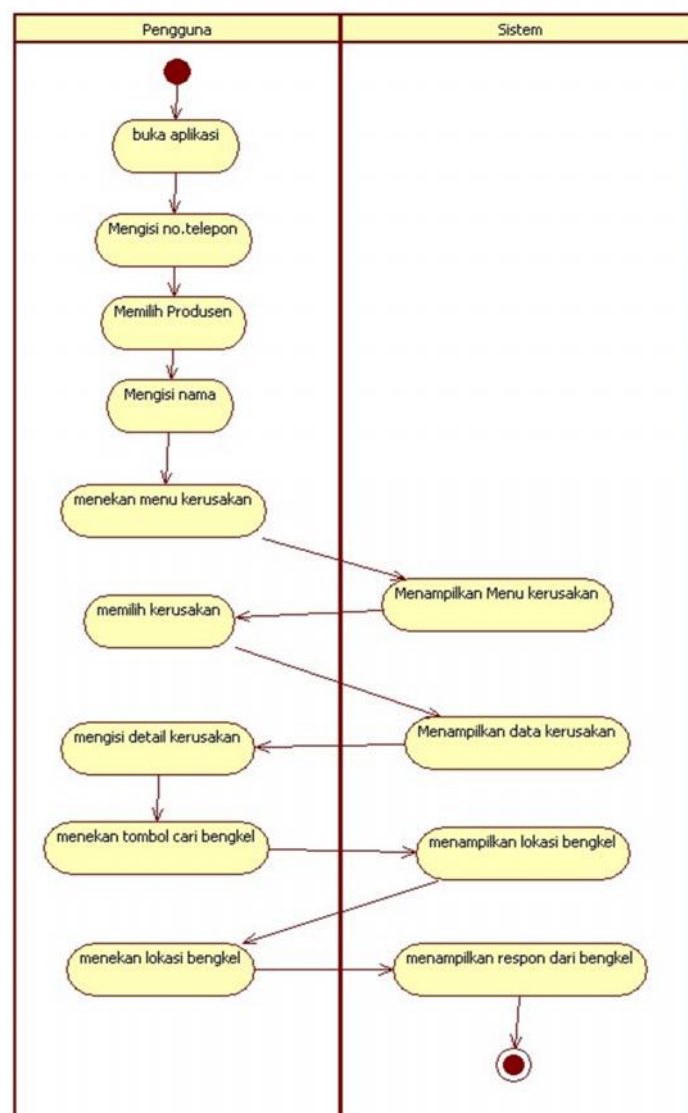
Activity diagram Cari Bengkel dimulai oleh pelanggan akan langsung membuka aplikasi, mengisi no.telepon, memilih produsen, isi nama, menekan menu cari bengkel dan menuju Menu utama dari aplikasi memilih menu kerusakan lalu pilih kerusakan, mengisi detail kerusakan dan memilih bengkel berdasarkan maps. *Activity* diagram cari Bengkel disajikan pada Gambar 3.3.



Gambar 3.3 Diagram Cari bengkel.

2. Activity Diagram Input Data kerusakan Mobil

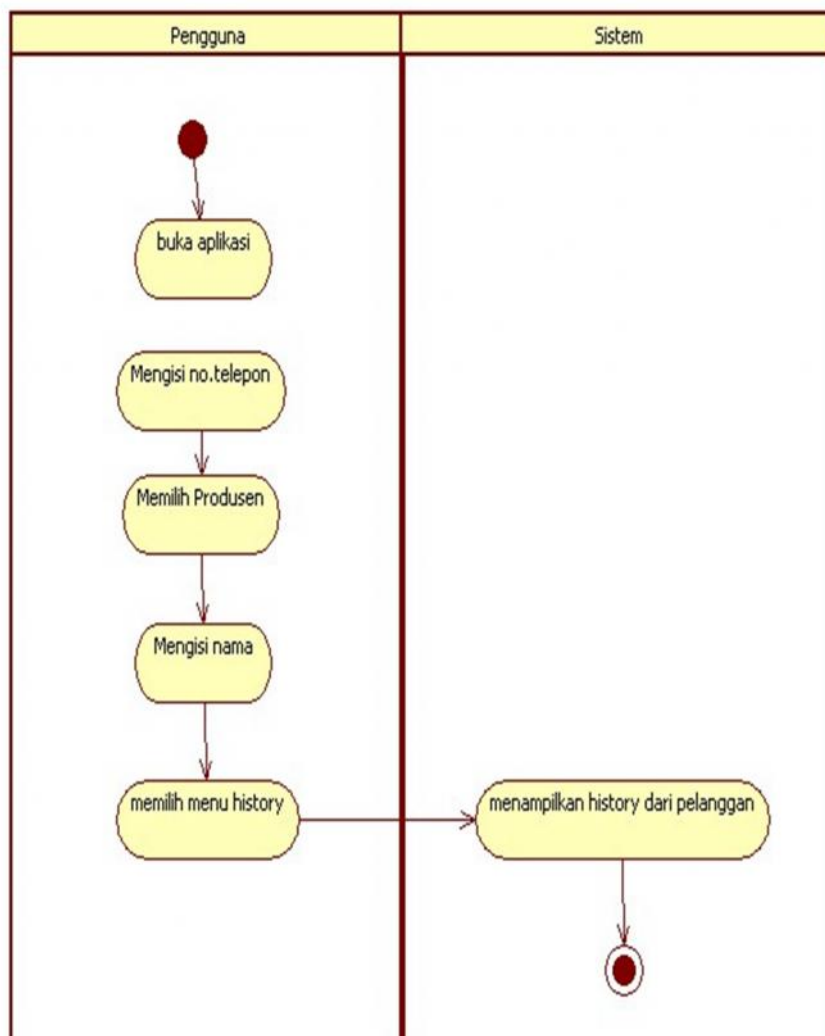
Activity diagram input data kerusakan mobil akan langsung menuju Menu utama dari aplikasi lalu memilih menu kerusakan lalu memilih kerusakan, mengisi detail kerusakan mobil, memilih bengkel berdasarkan maps dan menunggu respon dari bengkel. *Activity* diagram cari Bengkel. *Activity* diagram *input* data kerusakan mobil disajikan pada Gambar 3.4.



Gambar 3.4 Activity Diagram Input Data Kerusakan Mobil.

3. Activity Diagram History

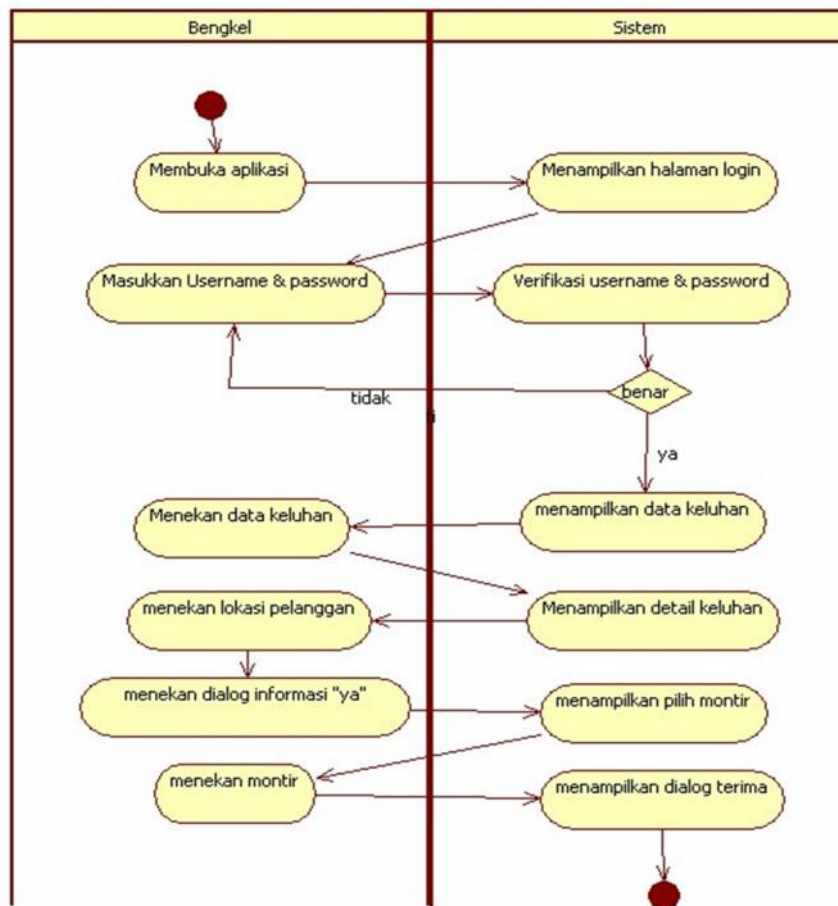
Activity diagram history adalah menu yang sangat penting pada aplikasi ini karena pelanggan dapat melihat transaksi setiap saat membuka aplikasi. menggunakan aplikasi. *Activity diagram history* disajikan pada Gambar 3.5.



Gambar 3.5 Activity Diagram History.

5. Activity Diagram Akses Kerusakan Mobil Pelanggan

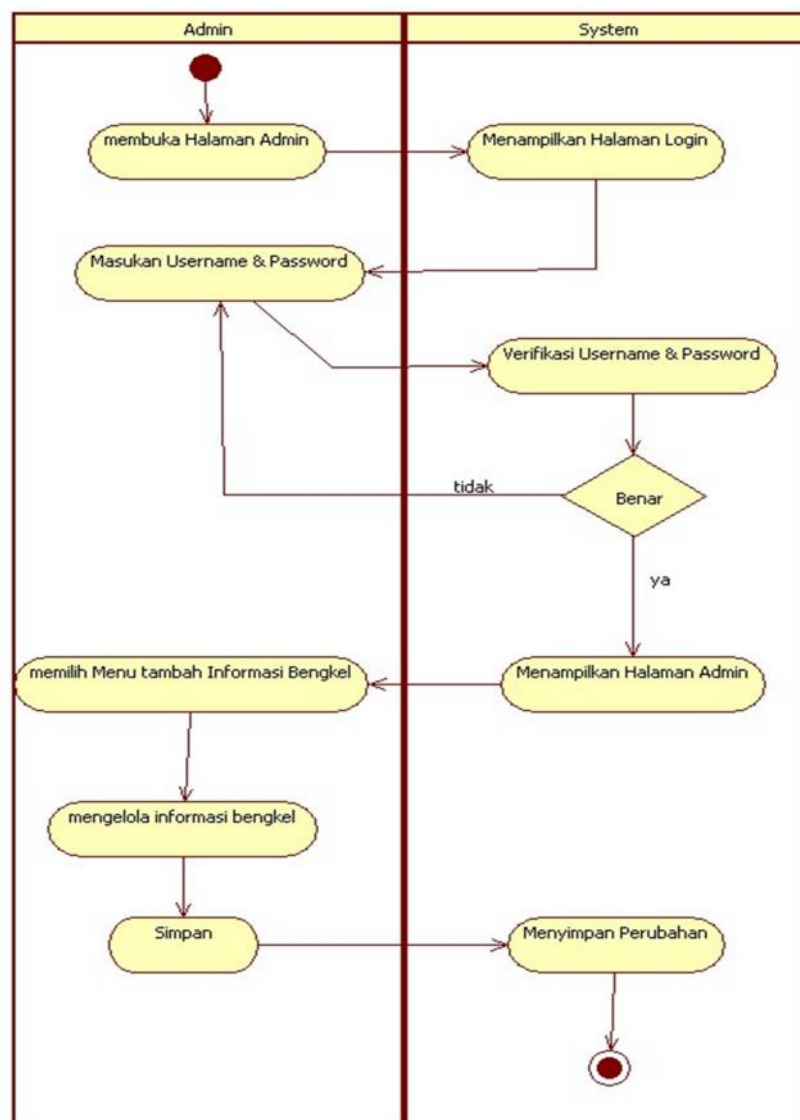
Activity diagram pengaturan dimulai oleh membuka aplikasi akan langsung, lalu membuka notifikasi di *smartphone* dan menampilkan *list* kerusakan dari berbagai pelanggan, menekan *list* kerusakan mobil, langsung menampilkan kerusakan dari pelanggan kemudian memilih montir lalu menekan letak pelanggan dan akan menampilkan koordinat letak pelanggan, menekan tombol setuju dan menampilkan pemberitahuan setuju kemudian menampilkan status bila sudah selesai di perbaiki montir langsung menekan tombol status selesai. Activity diagram akses kerusakan mobil pelanggan disajikan pada Gambar 3.6.



Gambar 3.6 Activity Diagram Akses Kerusakan Mobil Pelanggan.

6. Activity Diagram Kelola Data Bengkel

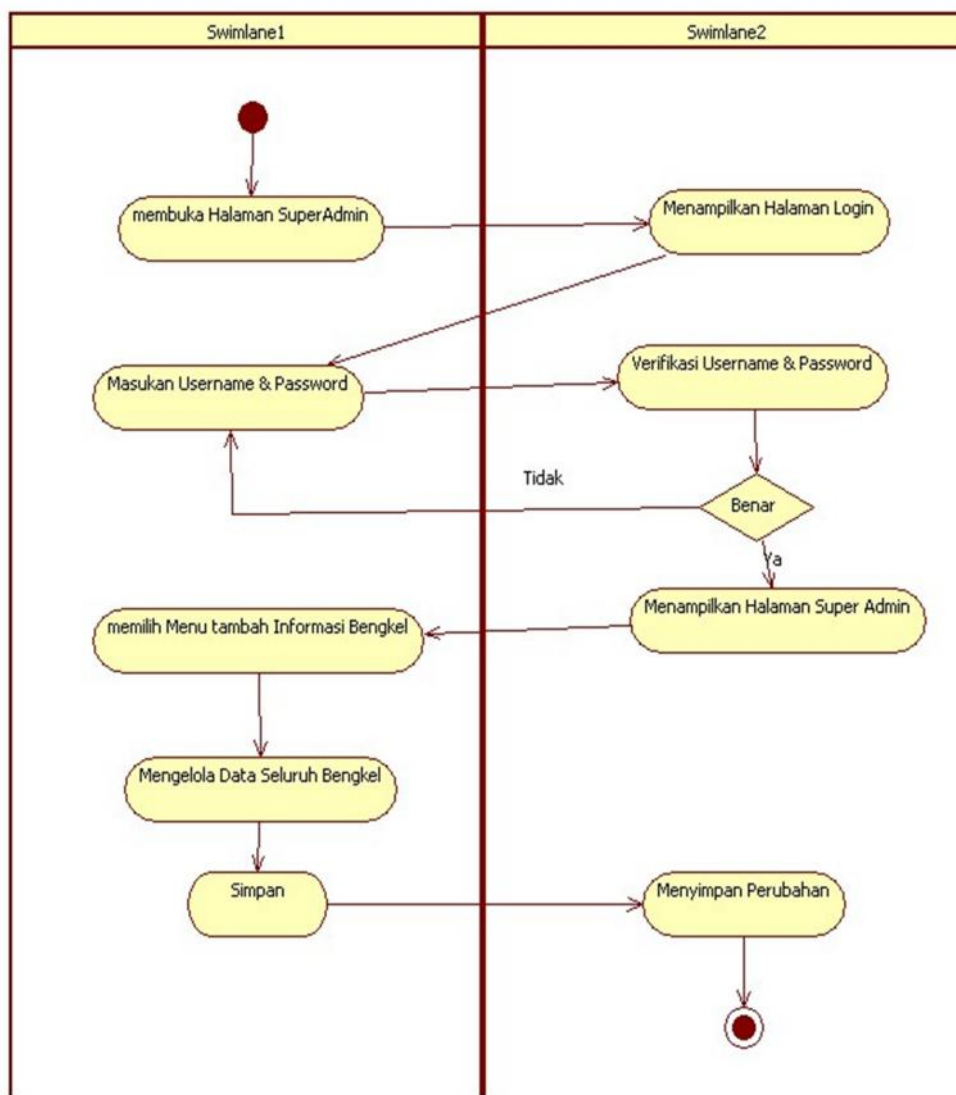
Activity diagram di mulai oleh membuka web lanjut ke *login*, masukan *username* dan *password* bila benar akan langsung menuju halaman utama kemudian *input*, hapus, edit data bengkel kemudian simpan. Activity diagram kelola data bengkel disajikan pada Gambar 3.7.



Gambar 3.7 Activity Diagram Kelola Data Bengkel.

7. Activity Diagram Kelola Data Seluruh Bengkel

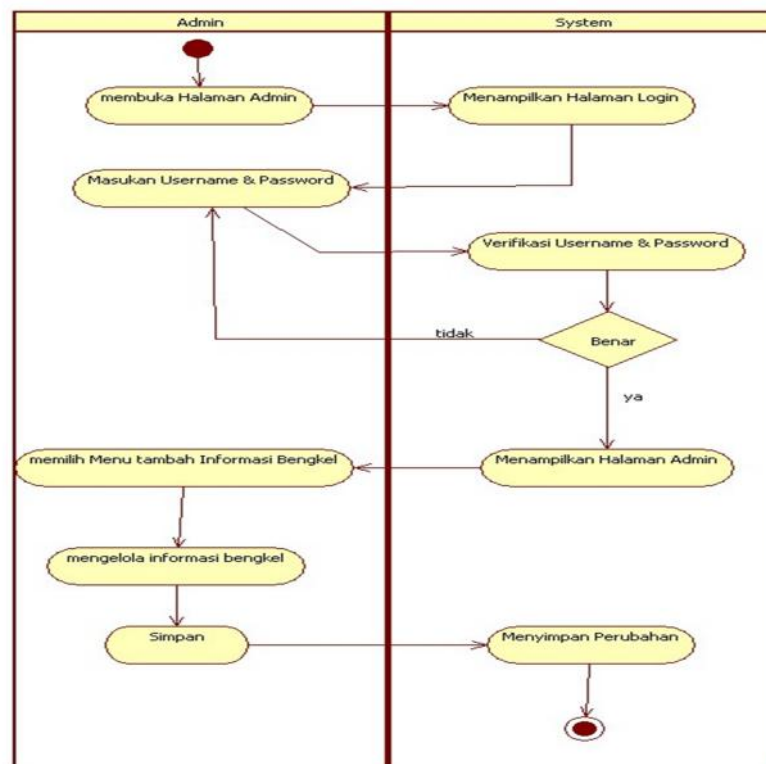
Activity diagram di mulai oleh membuka web lanjut ke *login*, masukan *username* dan *password* bila benar akan langsung menuju halaman utama kemudian *input*, hapus, edit data bengkel kemudian simpan. Activity diagram kelola data bengkel disajikan pada Gambar 3.8.



Gambar 3.8 Activity Diagram Kelola Data Seluruh Bengkel.

1. Sequence Diagram Logout

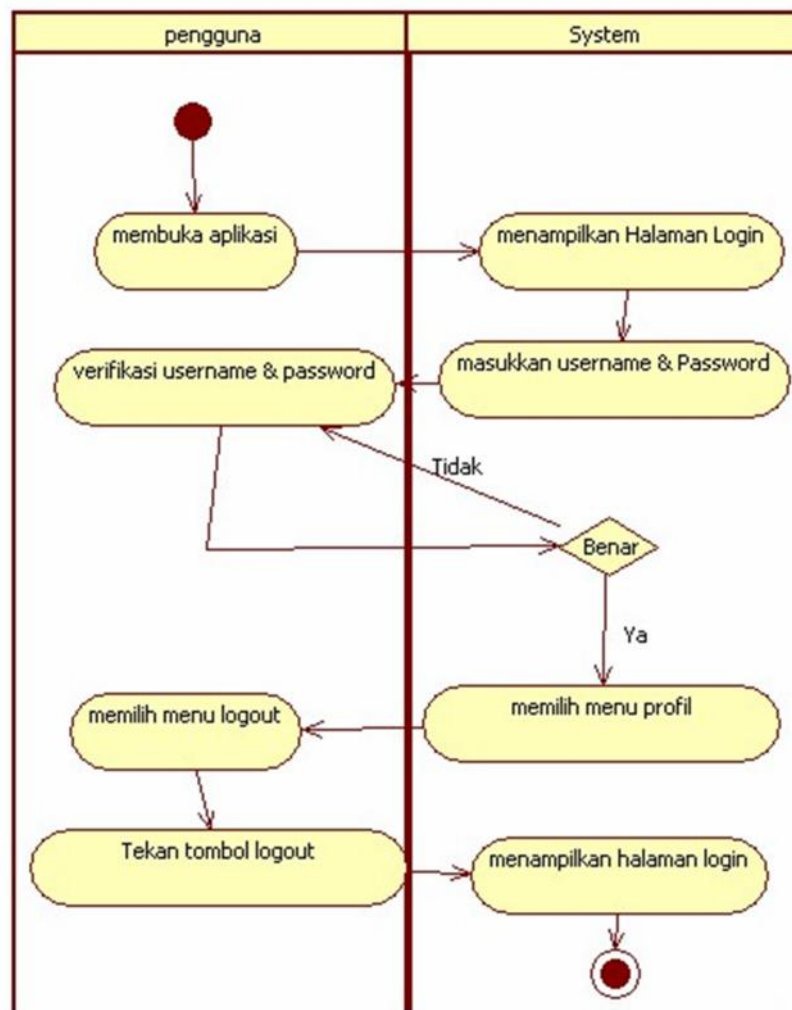
Ketika bengkel sudah Masuk ke halaman utama, bengkel menekan menu profil dan pilih *logout*, dan langsung menampilkan halaman *login*. *Sequence diagram logout* disajikan pada gambar 3.17.



Gambar 3.9 Sequence Diagram Logout

8. Activity Diagram Kelola Menu Logout

Activity diagram dimulai oleh membuka aplikasi lanjut ke *login*, masukan *username* dan *password* bila benar akan langsung menuju halaman profil kemudian menekan menu *logout* pilih “ya” langsung menampilkan halaman *login*. *Activity diagram* kelola data bengkel disajikan pada Gambar 3.9.



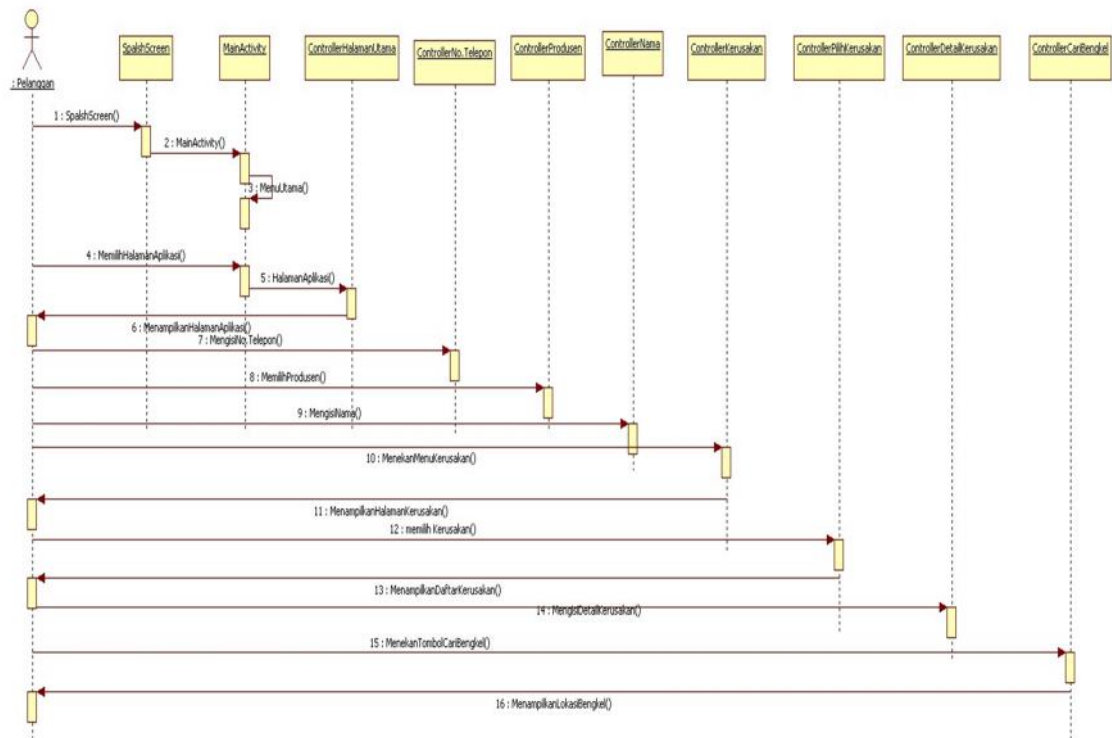
Gambar 3.10 Activity Diagram Kelola Data Logout.

4. Sequence Diagram

Menggambarkan interaksi antara sejumlah objek dalam urutan waktu. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antar objek yang terjadi pada titik tertentu dalam eksekusi sistem. Pada aplikasi Bengkel mobil terdapat 8 (tujuh) *sequence* diagram, yaitu sebagai berikut:

2. *Sequence Diagram Cari Bengkel*

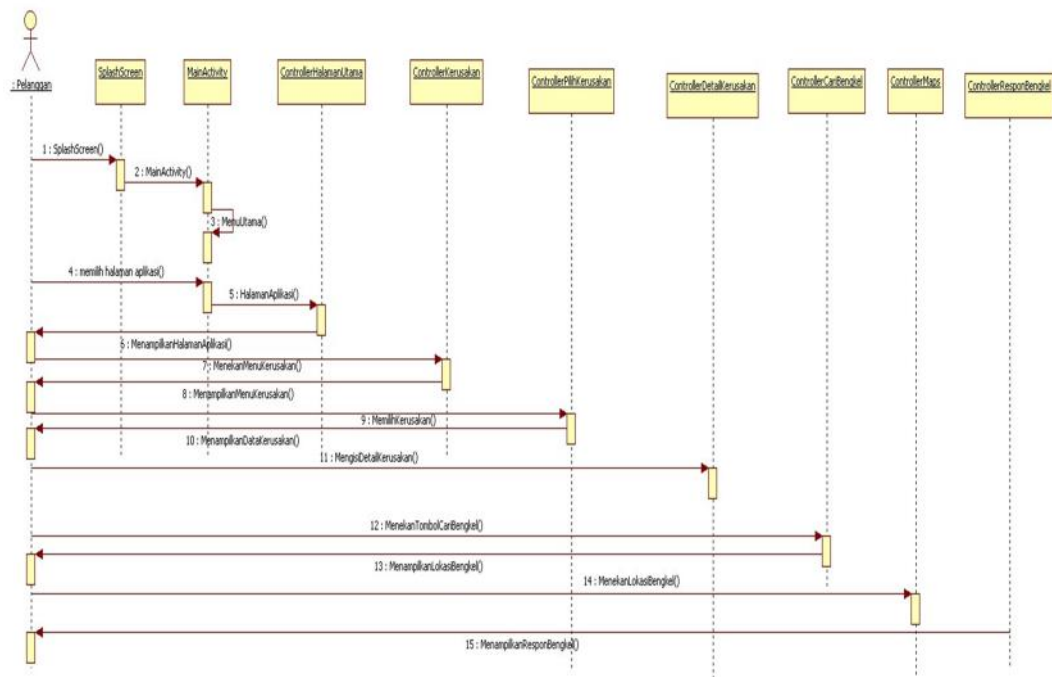
Untuk dapat memilih menu “Cari Bengkel” pelanggan pilih menu kerusakan pilih kerusakan, isi detail kerusakan, pilih bengkel *Sequence diagram Cari Bengkel* disajikan pada Gambar 3.11.



Gambar 3.11 *Sequence Diagram Cari Bengkel*

3. *Sequence Diagram Input Data Kerusakan Mobil*

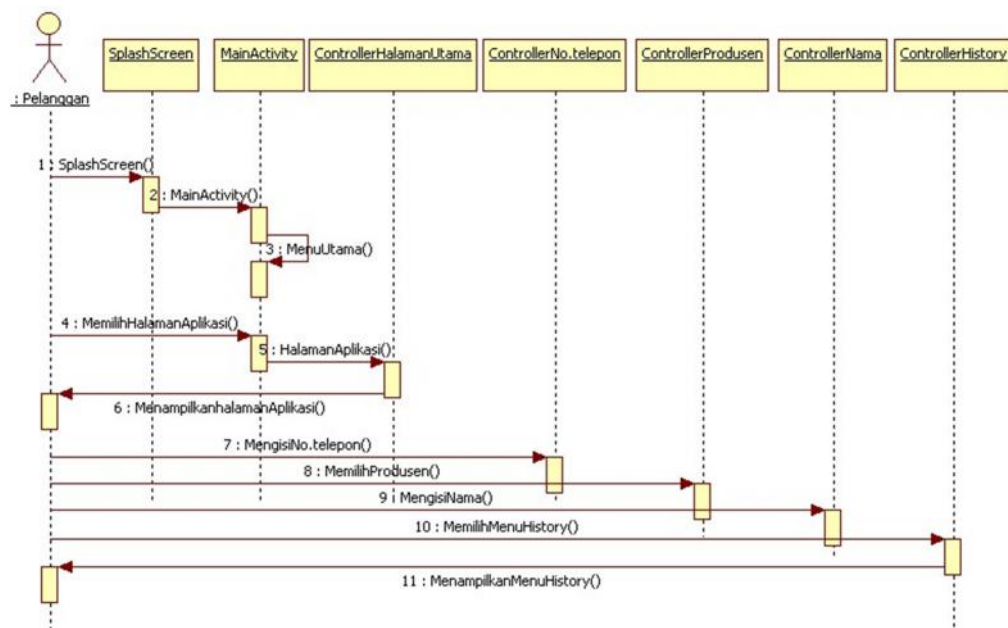
Untuk dapat *input* data kerusakan mobil pelanggan memilih menu kerusakan, pilih kerusakan, isi detail kerusakan mobil, pilih bengkel berdasarkan maps dan kirim kerusakan, lalu menunggu respon dari bengkel. *Sequence diagram input data kerusakan mobil* disajikan pada Gambar 3.12.



Gambar 3.12 *Sequence* Diagram Input Data Kerusakan Mobil.

4. *Sequence* Diagram History

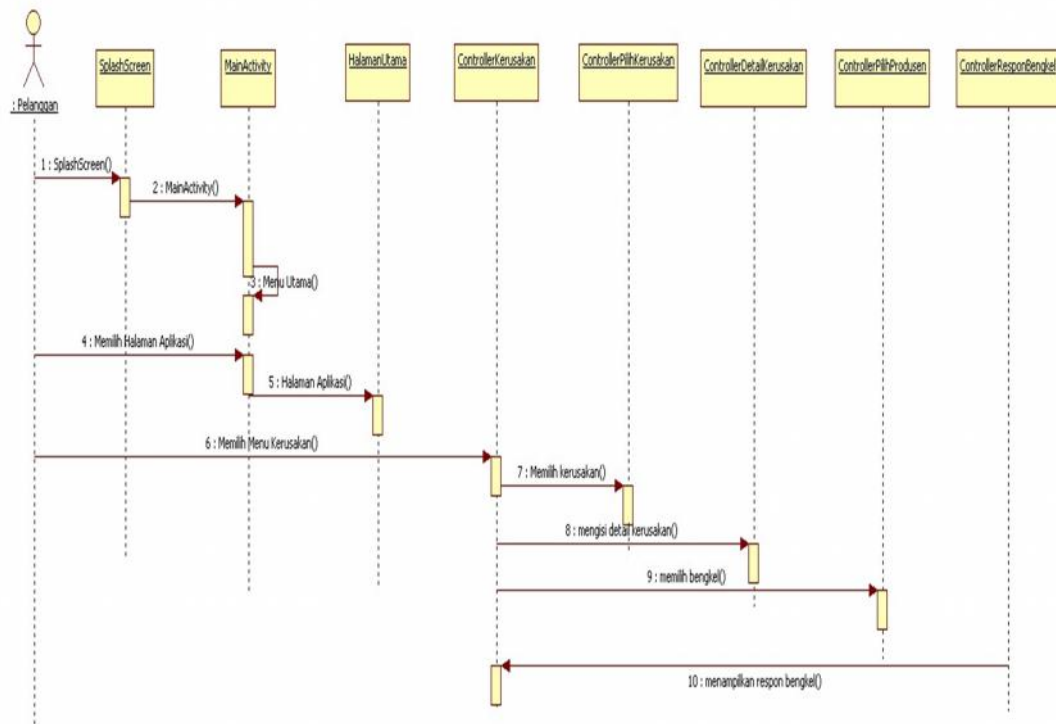
Ketika pelanggan berada di menu utama aplikasi, pelanggan dapat memilih menu “Profil”, pilih menu history, isi nama yang ingin di ubah dan no.telepon,dan klik tombol simpan. *Sequence* diagram history disajikan pada Gambar 3.13.



Gambar 3.13 *Sequence Diagram History*

5. *Sequence Diagram Akses Kerusakan Mobil Pelanggan*

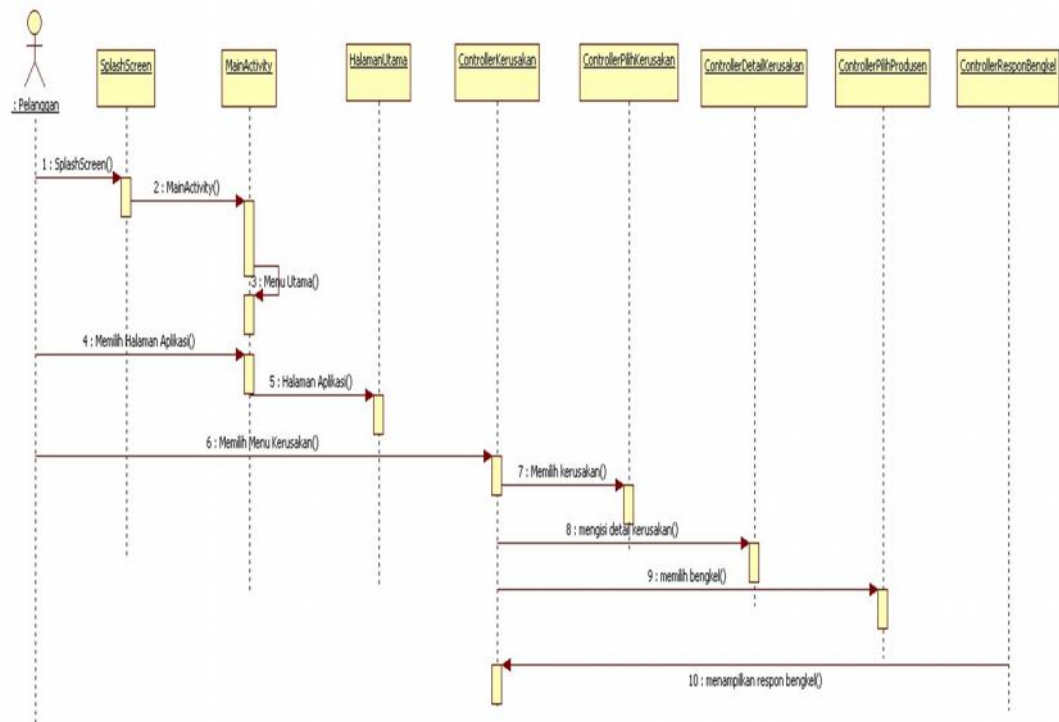
Ketika bengkel sudah *login* berada di menu utama aplikasi, bila mendapat *notifikasi* dari pelanggan pihak bengkel langsung melihat isi dari *notifikasi* tersebut dan menampilkan isi kerusakan mobil, kemudian memilih montir yang akan berangkat menuju lokasi kerusakan selanjutnya menekan tombol letak pelanggan lalu menampilkan letak pelanggan dan konfirmasi dari bengkel bahwa montir telah berangkat. *Sequence diagram* kelola mobil pelanggan disajikan pada gambar 3.14.



Gambar 3.14 *Sequence Diagram* Kerusakan Mobil Pelanggan

6. *Sequence Diagram* Akses Kerusakan Mobil Pelanggan

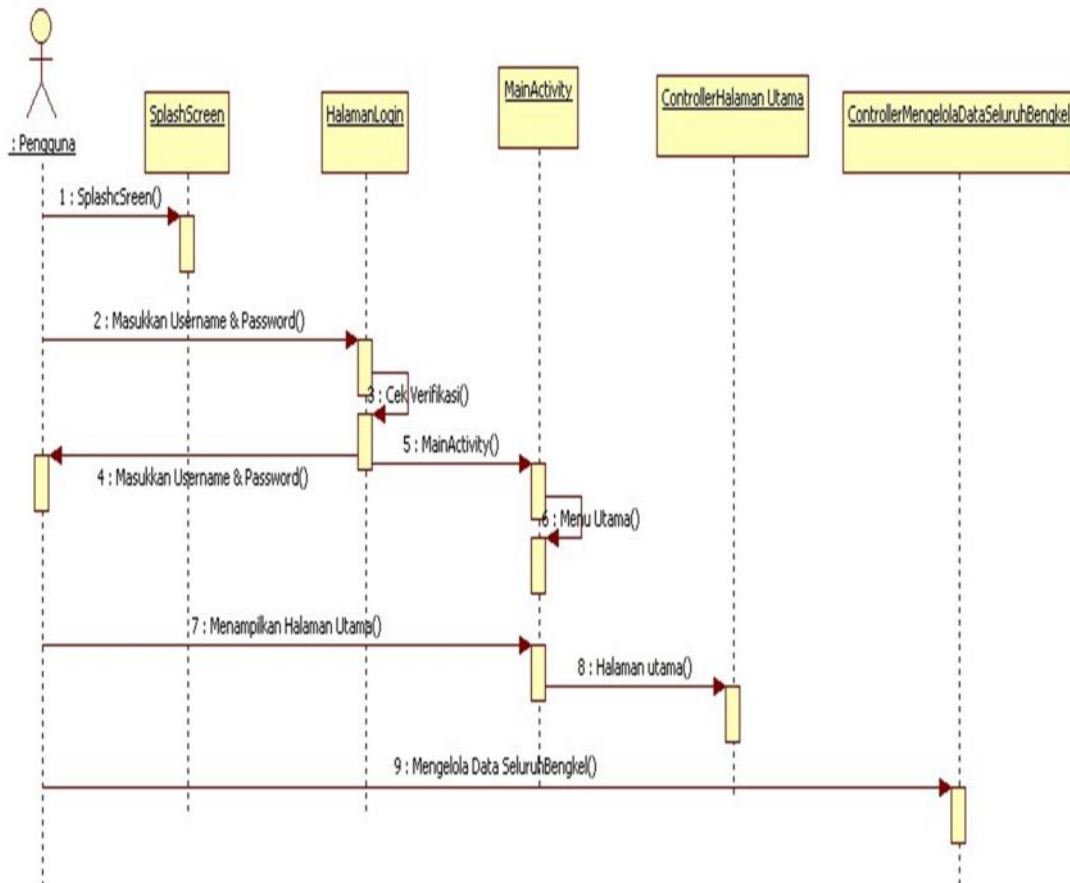
Ketika bengkel sudah *login* berada di menu utama aplikasi, bila mendapat *notifikasi* dari pelanggan pihak bengkel langsung melihat isi dari *notifikasi* tersebut dan menampilkan isi kerusakan mobil, kemudian memilih montir yang akan berangkat menuju lokasi kerusakan selanjutnya menekan tombol letak pelanggan lalu menampilkan letak pelanggan dan konfirmasi dari bengkel bahwa montir telah berangkat. *Sequence diagram* kelola mobil pelanggan disajikan pada gambar 3.15.



Gambar 3.15 *Sequence Diagram* Kerusakan Mobil Pelanggan

7. *Sequence Diagram* Kelola Data Seluruh Bengkel

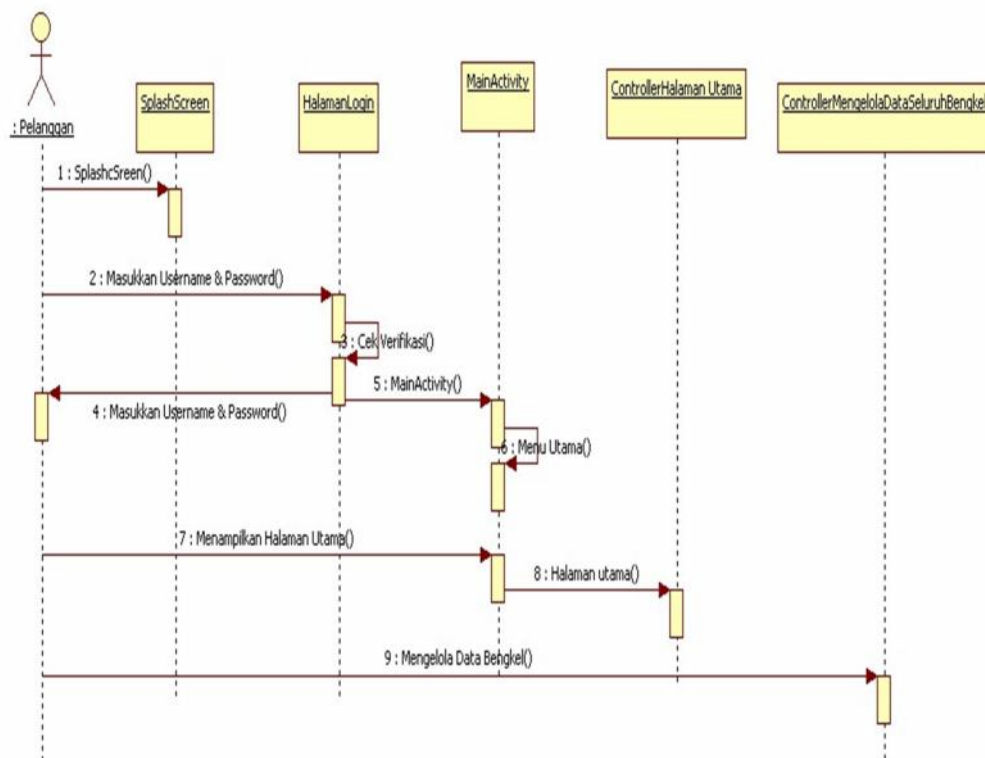
Ketika super admin sudah *login* langsung menampilkan halaman utama dari admin kemudian super admin menambah ,edit, hapus data bengkel yang sudah ada dan langsung tersimpan di *database*. *Sequence diagram* kelola data seluruh bengkel disajikan pada Gambar 3.16.



Gambar 3.16 *Sequence Diagram* Kelola Data Bengkel

8 *Sequence Diagram* Kelola Data Bengkel

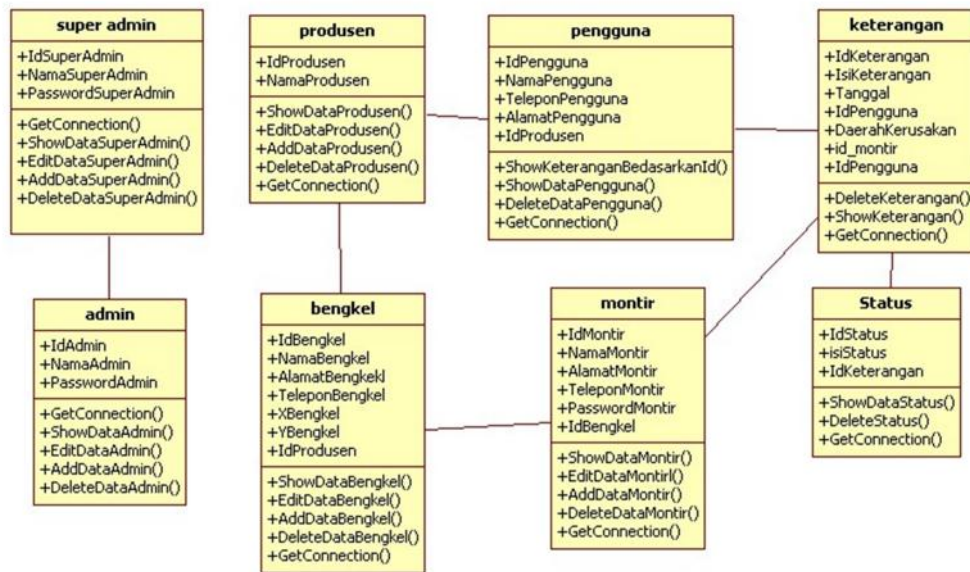
Ketika admin sudah *login* langsung menampilkan halaman utama dari admin kemudian admin menambah ,edit, hapus, mengganti profil bengkel yang sudah ada dan langsung tersimpan di *database*. *Sequence diagram* kelola data bengkel disajikan pada Gambar 3.17.



Gambar 3.17 *Sequence Diagram* Kelola Data Bengkel.

3. Class Diagram

Class diagram mendeskripsikan jenis – jenis obyek dalam sistem dan berbagai macam hubungan statis yang terjadi. Pada aplikasi bengkel mobil terdapat 8 kelas utama yang dibangun diantaranya superadmin, admin, produsen, bengkel, montir, dan keterangan, pelanggan, status. *Class* Diagram Bengkel mobil disajikan pada Gambar 3.18.



Gambar 3.18 Class Diagram Bengkel mobil.

3.4.4.2 Perancangan Antarmuka

Perancangan antarmuka merupakan proses penggambaran bagaimana sebuah tampilan (*interface*) sistem dibentuk. Aplikasi bengkel mobil dirancang dengan tampilan yang *user friendly*, sehingga diharapkan dapat mempermudah pelanggan dalam menggunakan aplikasi ini. Dalam Aplikasi ini terdapat beberapa *layout* atau *form* antara lain.

1. *Layout Splash Screen*

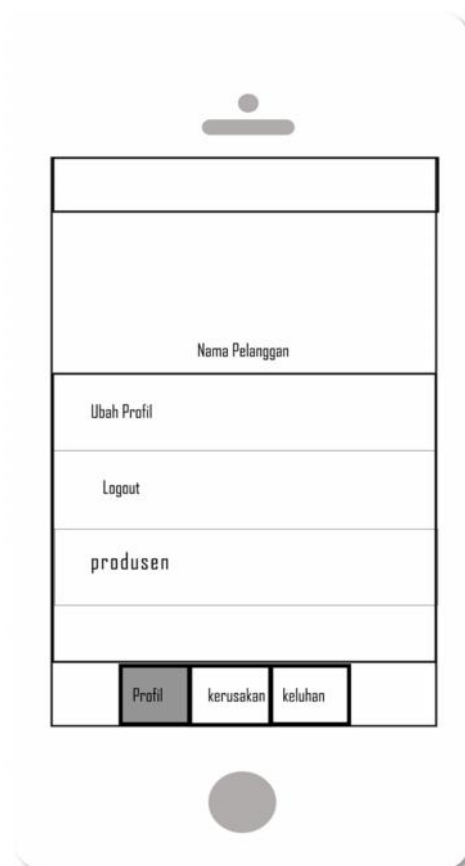
Splash Screen adalah *form* yang ditampilkan diawal ketika aplikasi/program dijalankan. aplikasi bengkel mobil menggunakan *splash screen* yang muncul sepersekian detik pada saat pertama membuka aplikasi bengkel mobil. *Splash screen* di sini dimaksudkan sebagai estetika untuk menunjukkan identitas aplikasi saja, tanpa fungsi lainnya. Perancangan *layout splash screen* aplikasi bengkel mobil disajikan pada Gambar 3.19.



Gambar 3.19 *Design Layout Splash Screen.*

2. *Layout Menu Profil*

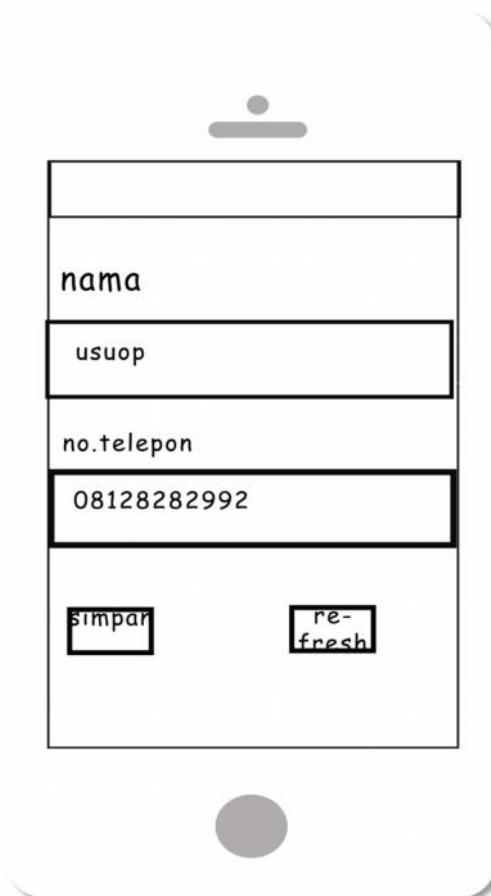
Menu utama berisikan menu-menu Ubah profil, produsen dan logout dan terdapat nama pelanggan pada bagian atas. Pada bagian bawah terdapat menu profil, kerusakan, *history* Perancangan *layout* menu profil aplikasi bengkel mobil disajikan pada Gambar 3.20.



Gambar 3.20 *Design Layout Menu Profil*

1. *Layout Menu Ubah Profil*

Pada *layout* menu ubah profil terdapat 2 kolom yang berisi nama dan no.telepon. sedangkan pada bagian bawah terdapat 2 *button* yaitu *button simpan* dan *button refresh*. Perancangan *layout* menu ubah profil disajikan pada Gambar 3.21.

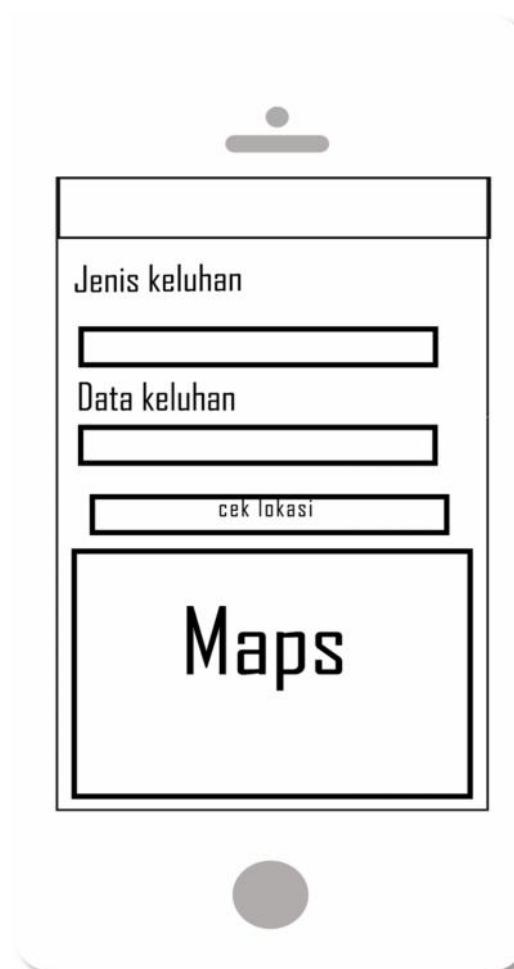


The image shows a mobile application interface for updating a profile. It features a central form with several input fields and two buttons. At the top of the form is an empty rectangular box. Below it is a label 'nama' followed by a text input field containing the text 'usuop'. Underneath is a label 'no.telepon' followed by a text input field containing the number '08128282992'. At the bottom of the form are two buttons: 'simpan' on the left and 're-fresh' on the right. The entire form is set against a light gray background that resembles a smartphone screen, complete with a home indicator bar at the top and a circular home button at the bottom.

Gambar 3.21 *Design Layout Ubah profil*

2. *Layout* Menu kerusakan

Ketika pelanggan memilih menu ini, pelanggan dapat memilih kerusakan pada mobil anda dan dapat mengisi detail kerusakan lalu akan menampilkan maps bengkel tersebut dan pada lokasi bengkel dapat diklik untuk lebih lanjut ke bengkel. Perancangan *layout* menu kerusakan berdasarkan kerusakan disajikan pada Gambar 3.22.



Gambar 3.22 *Design Layout* kerusakan

3. *Layout Menu respon bengkel*

Pada layout respon bengkel menu terakhir pada aplikasi pelanggan dan menu ini pelanggan hanya dapat menunggu respon dari bengkel dan di bawah terdapat button batal yang dimana pelanggan dapat mencari bengkel lain yang merespon.. Perancangan *layout* “Input Kerusakan” disajikan pada Gambar 3.24.



Gambar 3.24 *Design Layout Respon Bengkel*

4. *Layout Menu Registrasi No.Telepon*

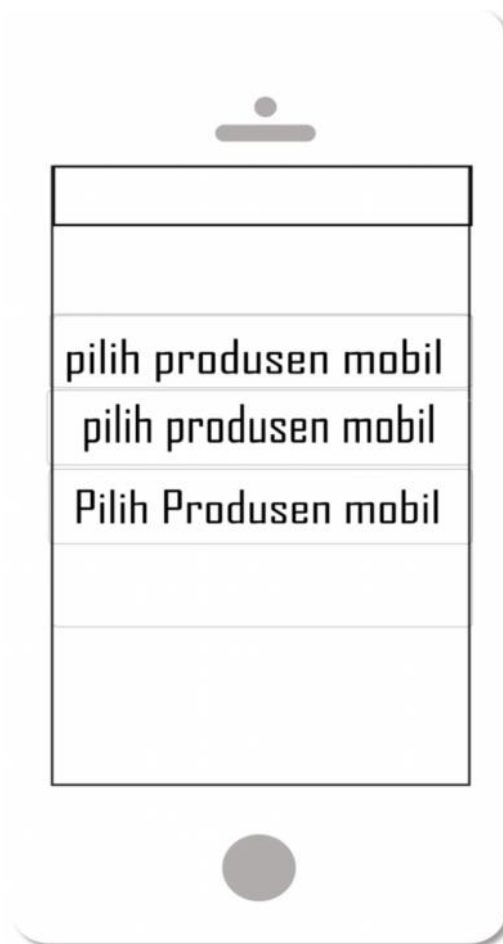
Pada menu registrasi no. telepon pelanggan yang belum terdaftar sebelumnya mengisi no.telepon dan terdapat *button* cek. Bila pelanggan sudah pernah memakai aplikasi ini dan mengisi no telepon yang sudah terdaftar akan langsung masuk ke menu utama pelanggan.. Perancangan *layout* "registrasi no.telepon" disajikan pada Gambar 3.24.



Gambar 3.24 *Design Layout Cek no.telepon*

5. *Layout* Menu Pilih Produsen

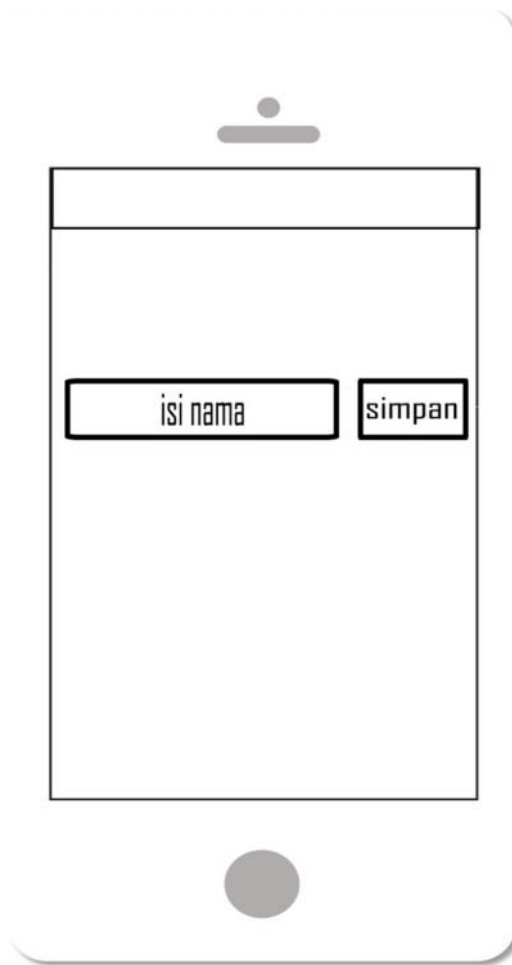
Pada menu pilih produsen pelanggan hanya dapat memilih produsen yang terdaftar pada aplikasi saja dan tidak bisa menambahkan bengkel yang tidak ada. Pelanggan tidak dapat memilih dua produsen, bila sudah memilih produsen dan me *klik* produsen tersebut akan pindah ke menu isi nama. Perancangan *layout* pilih produsen disajikan pada Gambar 3.25.



Gambar 3.25 *Design Layout* Pilih Produsen.

6. Layout Menu Isi Nama

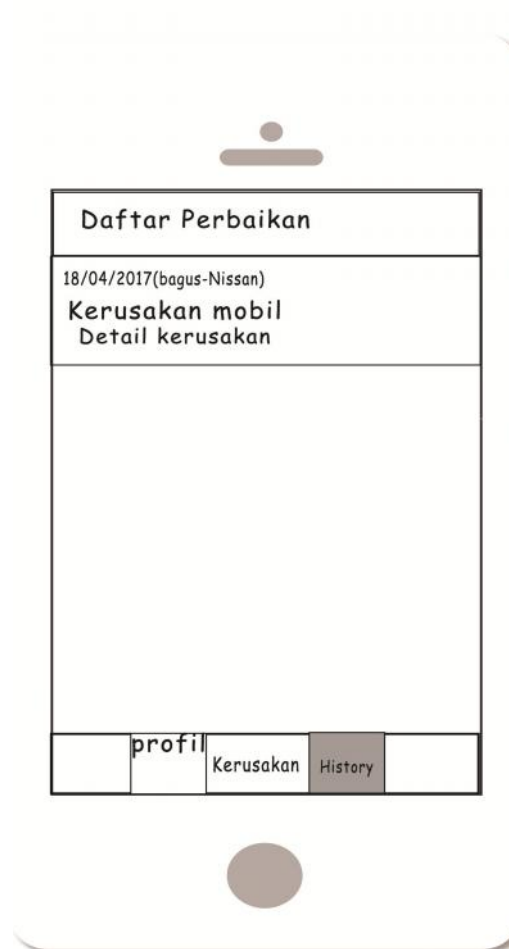
Pada menu isi nama pelanggan yang belum terdaftar akan melewati pada menu ini karena mengisi nama wajib di isi bila tidak pelanggan tidak bisa ke menu utama karena ini menu terakhir sebelum ke menu utama untuk mencari bengkel.. Perancangan *layout* isi nama disajikan pada Gambar 3.26.



Gambar 3.26 *Design Layout* Isi Nama.

7. *Layout Menu History*

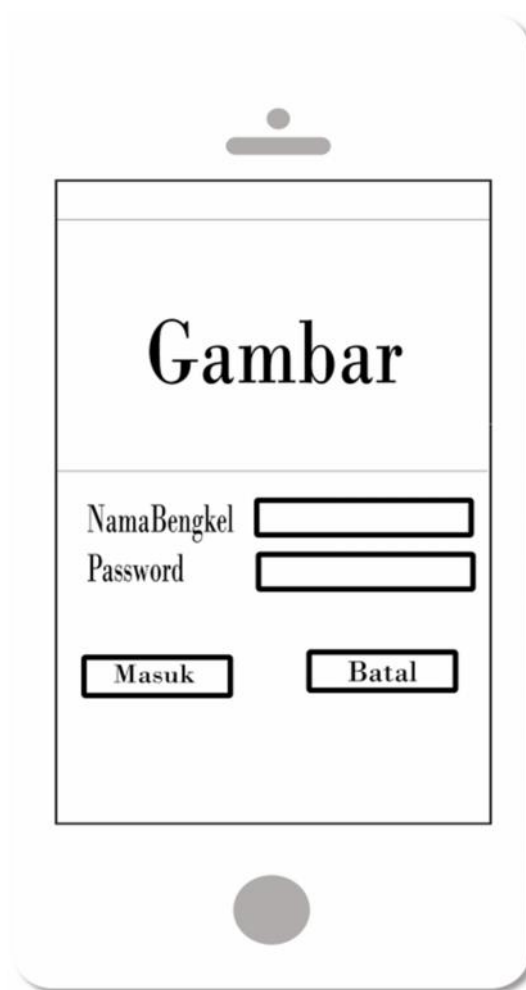
Pada menu *history* pelanggan hanya dapat melihat transaksi yang sudah dilakukan, menu history d terdapat tanggal kerusakan, nama, produsen mobil, kerusakan pada mobil, detail kerusakan pada mobil. dapat dilihat bila pelanggan sudah melakukan transaksi dengan pihak bengkel. Perancangan *layout* menu history disajikan pada Gambar 3.27.



Gambar 3.27 *Design Layout Menu History.*

8. Layout Menu Login Bengkel

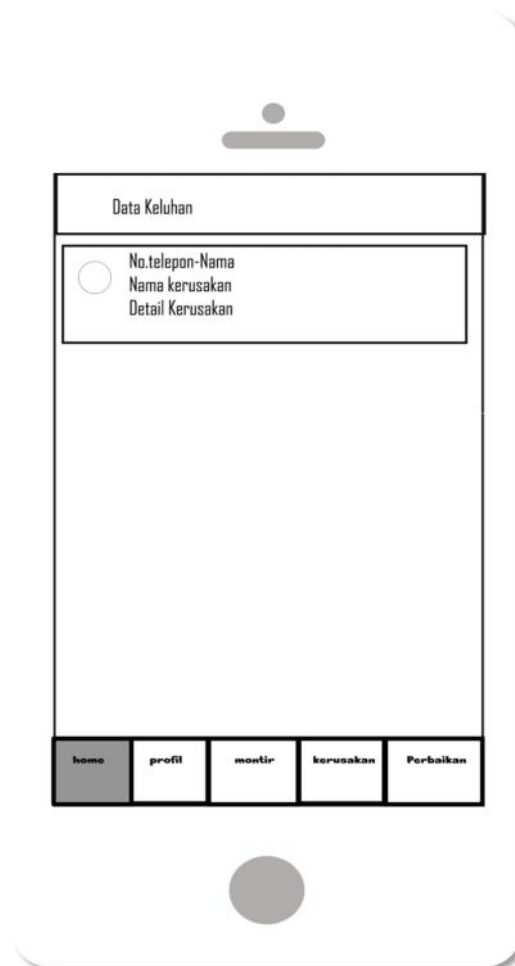
Pada menu login bengkel bila sudah tampil menu login sebagai bengkel yang terdapat form *username* dan *password* kedua form itu wajib, lalu tekan tombol masuk bila. Perancangan *layout login* admin disajikan pada Gambar 3.27.



Gambar 3.28 *Design Layout* Login Bengkel.

9. *Layout Menu List Kerusakan*

Ketika muncul *notifikasi* dari *smartphone* pihak bengkel bisa langsung membukanya dan akan tampil menu *list* kerusakan dari pelanggan mobil, bila ada tiga kerusakan secara langsung bisa membuka satu persatu dengan menekan tombol tekan. Perancangan *layout* menu *list* Kerusakan disajikan pada Gambar 3.29.



Gambar 3.29 *Design Layout* List Kerusakan.

10. *Layout* Menu Kerusakan dari Pelanggan

Ketika sudah menerima kerusakan dari pelanggan, pihak bengkel bisa langsung melihat pelanggan dan mengirim montir ke lokasi pelanggan mengalami kerusakan. Perancangan *layout* “Input Kerusakan” di sajikan pada Gambar 3.30.

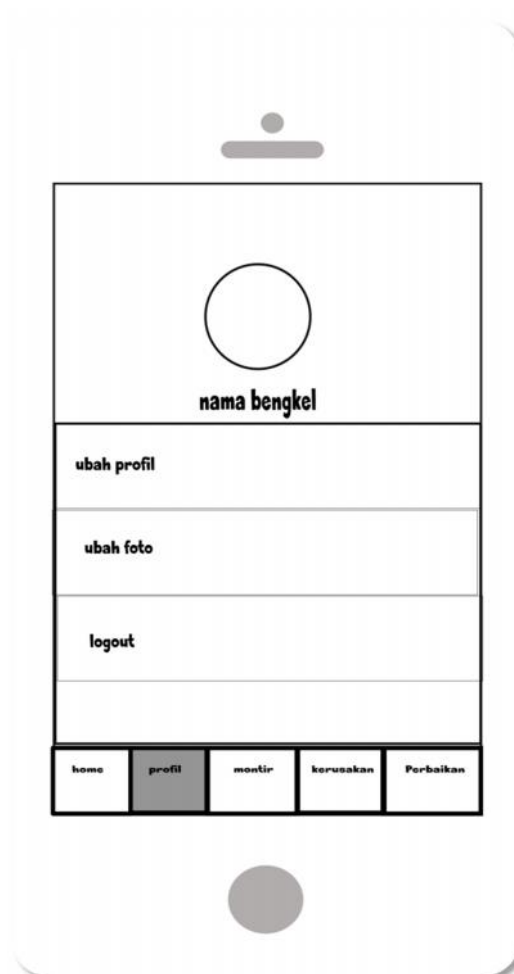


Detail keluhan
pengguna
(notelepon)-NamaPelangan
Jeniskerusakan/datakeluhan
Cek lokasi
maps

Gambar 3.30 *Design Layout* Input Kerusakan Mobil.

11. *Layout* Menu Profil

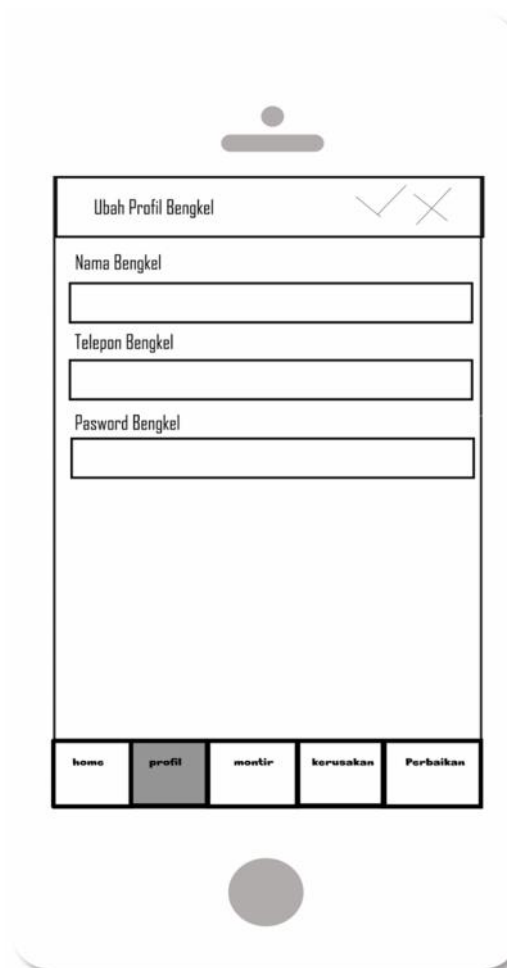
Pada menu Profil Bengkel terdapat foto bengkel dan *button* history, ubah foto, logout. Perancangan *layout* menu profil disajikan pada Gambar 3.31.



Gambar 3.31 *Design Layout* Profil

12. *Layout* Menu Ubah Profil

Pada Menu Kerusakan terdapat menu tambah kerusakan yang bisa langsung diisi, tak perlu lewat web. Menu ini di setiap bengkel memiliki kerusakan yang berbeda beda. Perancangan *layout* ubah profil disajikan pada Gambar 3.32.

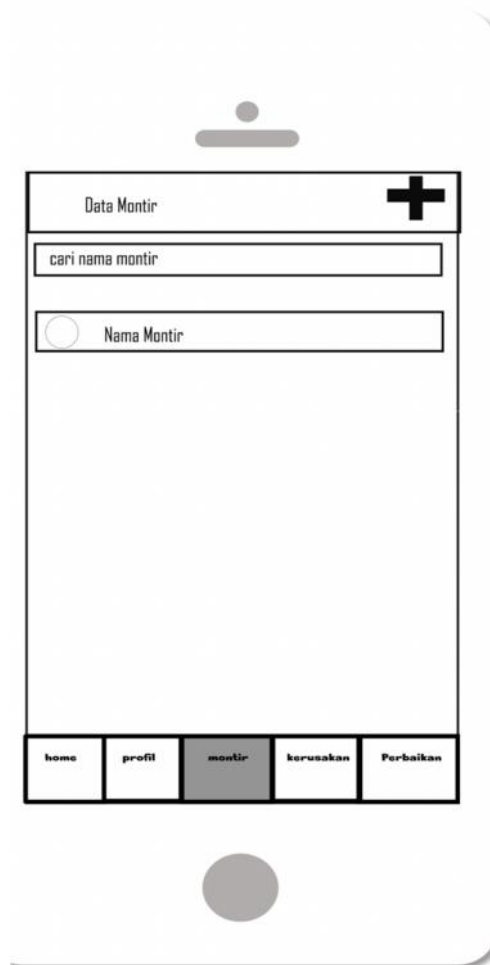


The image shows a mobile application screen for editing a workshop profile. At the top, there is a header bar with the title "Ubah Profil Bengkel" and two icons: a checkmark and a close (X) button. Below the header, there are three input fields labeled "Nama Bengkel", "Telepon Bengkel", and "Pasword Bengkel". Below these fields is a large empty rectangular area. At the bottom of the screen, there is a navigation bar with five buttons: "home", "profil", "montir", "kerusakan", and "Perbaikan". The "profil" button is currently selected and highlighted in grey. Above the navigation bar, there is a small grey circle and a horizontal bar, likely representing a camera or sensor. Below the navigation bar, there is a large grey circle, likely representing a home button or a large touch target.

Gambar 3.32 *Design Layout* Ubah Profil Bengkel

13. *Layout* Menu Montir

Pada Menu Montir terdapat. Menu ini di setiap bengkel memiliki montir yang berbeda beda dan terdapat button tambah di atas menu montir bila sudah menambahkan akan muncul di depan menu montir. Perancangan *layout* montir disajikan pada Gambar 3.33.



Gambar 3.33 *Design Layout* Montir

14. *Layout* Menu Tambah Montir

Pada Menu tambah data montir terdapat kolom nama montir dan telepon montir. Pada menu tambah montir terdapat button simpan dan button batal pada setiap kolom wajib diisi dan nama montir tidak diisi maka tidak dapat tersimpan. Perancangan *layout* tambah montir disajikan pada Gambar 3.34.



The image shows a mobile application interface for adding a mechanic's data. At the top, there is a header bar with the title "Tambah Data Montir" and two buttons: "simpan" (save) and "batal" (cancel). Below the header, there are two input fields: "Nama Montir" (Mechanic Name) and "Telepon Montir" (Mechanic Phone). The "Nama Montir" field is currently empty, while the "Telepon Montir" field contains the number "08123456789". At the bottom of the screen, there is a navigation bar with five tabs: "home", "profil", "montir", "kerusakan", and "Perbaikan". The "montir" tab is currently selected and highlighted in dark gray. The entire interface is displayed on a white background with rounded corners and a subtle shadow.

Gambar 3.34 *Design Layout* Tambah Data Montir

15. *Layout* Menu Ubah Data Montir

Pada Menu ubah data montir terdapat kolom nama montir dan telepon montir. Perancangan *layout* ubah data montir disajikan pada Gambar 3.35.

The image shows a mobile application interface for editing a technician's profile. The screen is titled "Ubah Profil Montir" with "oke" and "batal" buttons. It contains two input fields: "Nama Montir" and "Telepon Montir". The bottom navigation bar has five items: "home", "profil", "montir" (which is highlighted), "kerusakan", and "Perbaikan".

Gambar 3.35 *Design Layout* Tambah Data Montir

16. *Layout* Menu Kerusakan

Pada Menu kerusakan terdapat kolom cari nama kerusakan, nama kerusakan dan tambah kerusakan. Perancangan *layout* kerusakan disajikan pada Gambar 3.36.



Gambar 3.36 *Design Layout* Kerusakan

17. *Layout* Menu Tambah Kerusakan

Pada menu tambah kerusakan terdapat kolom cari nama kerusakan, nama kerusakan dan tambah kerusakan. Perancangan *layout* tambah kerusakan disajikan pada Gambar 3.37.

The image shows a mobile application interface for adding damage. The screen is titled "tambah kerusakan" with "Simpan" and "Batal" buttons. Below the title is a text input field labeled "Nama Kerusakan". At the bottom is a navigation bar with five items: "home", "profil", "montir", "kerusakan" (highlighted), and "Perbaikan".

Gambar 3.37 *Design Layout* Tambah Kerusakan

18. *Layout* Menu Ubah Kerusakan

Pada menu ubah kerusakan terdapat kolom nama kerusakan dan *button* simpan dan batal. Perancangan *layout* ubah kerusakan disajikan pada Gambar 3.38.

Ubah Data Kerusakan

simpan batal

Nama Kerusakan

home profil montir **kerusakan** Perbaikan

Gambar 3.38 *Design Layout* Tambah Kerusakan

19. *Layout* Menu Perbaikan

Pada menu ubah kerusakan terdapat kolom nama kerusakan dan *button* simpan dan batal. Perancangan *layout* ubah kerusakan disajikan pada Gambar 3.39.

The image shows a mobile application interface for a repair menu. The layout is as follows:

- Title Bar:** A white bar at the top with a grey dot in the center, containing the text "Data perbaikan".
- Status Buttons:** Two buttons below the title bar: a green button labeled "sudah selesa" and a pink button labeled "belum selesai".
- Input Fields:** Two stacked input fields labeled "periode awal" and "periode akhir".
- Action Button:** A button labeled "cek data" located below the input fields.
- List Area:** A large white box at the bottom containing the following labels:
 - tanggal/tahun (no.telepon) montir
 - Jenis kerusakan
 - Detail kersakan

Gambar 3.39 *Design Layout* Perbaikan

20. *Layout Menu Login Super Admin*

Bila sudah tampil menu login sebagai super admin yang terdapat form *username* dan *password* kedua form itu wajib, lalu tekan tombol masuk bila. Perancangan *layout login* admin disajikan pada Gambar 3.40.

```
graph TD; subgraph LOGIN; U[username]; P[password]; M[masuk]; B[batal]; end
```

The diagram illustrates the layout of the Admin Login menu. It consists of a main container divided into two sections. The top section is labeled 'HEADER'. The bottom section is labeled 'LOGIN' and contains a form with two input fields: 'username' and 'password'. Below these fields are two buttons: 'masuk' (login) and 'batal' (cancel).

Gambar 3.40 *Design Layout Login Admin*

21. *Layout Menu Home*

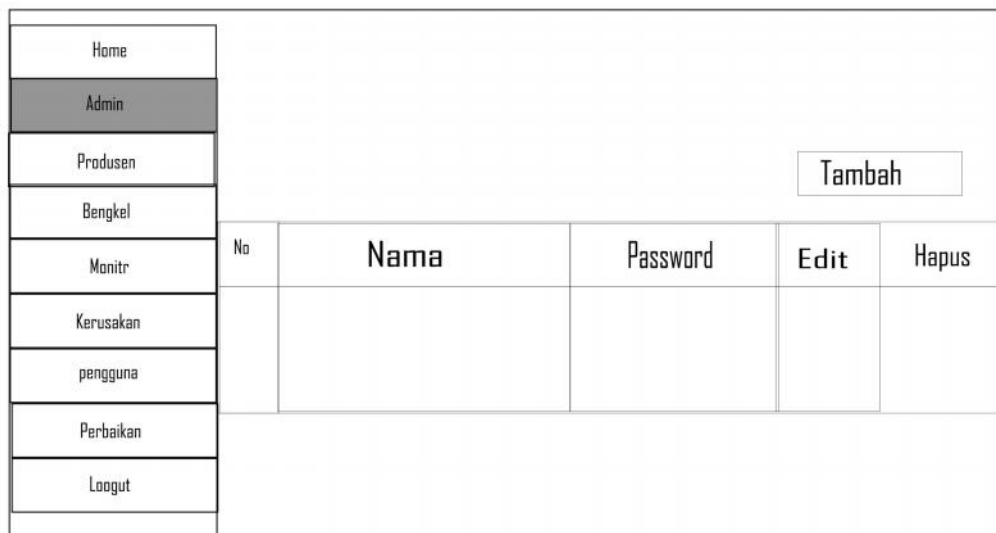
Ketika super admin sudah berhasil *login* akan tampil menu “*Home*” untuk memberi tahu bagaimana aplikasi ini berjalan. Perancangan *layout menu home* disajikan pada Gambar 3.40.



Gambar 3.41 *Design Layout Menu Home.*

22. Layout Menu Admin

Ketika super admin sudah berhasil *login* akan tampil menu “Admin” untuk menambahkan admin ada tambah data dan ada tombol edit dan hapus. Perancangan *layout* menu admin disajikan pada Gambar 3.42.



Gambar 3.42 *Design Layout Menu Admin.*

23. Layout Menu Kerusakan

Ketika super admin sudah berhasil *login* akan tampil menu “Kerusakan” pada menu kerusakan super admin bisa menambahkan kerusakan pada setiap bengkel, edit dan hapus. Perancangan *layout* menu kerusakan disajikan pada Gambar 3.43.

Home					
Admin					
Produsen					
Bengkel					
Monitr					
Kerusakan	No	Kerusakan	Bengkel	Edit	Hapus
pengguna					
Perbaikan					
Logout					

Gambar 3.43 Design Layout Menu Kerusakan.

24. Layout Menu Produsen

Ketika super admin sudah berhasil *login* akan tampil menu “Produsen” untuk melihat dari inputan produsen mobil yang ada di Bandar lampung serta ada tombol tambah data, edit dan hapus. Perancangan *layout* menu produsen disajikan pada Gambar 3.44.

Home					
Admin					
Produsen	Tambah				
Bengkel	No	Produsen	Logo	Edit	Hapus
Monitr					
Kerusakan					
pengguna					
Perbaikan					
Loogut					

Gambar 3.44 *Design Layout* Menu Produsen.

25. *Layout* Menu Bengkel

Menu bengkel melihat tampilan yang telah diisi oleh super admin dan memiliki tombol tambah data, edit, hapus. Perancangan *layout* menu bengkel disajikan pada Gambar 3.45.

Home							
Admin							
Produsen	Tambah						
Bengkel	No	Bengkel	Alamat	Koordinat	Foto	edit	Hapus
Monitr							
Kerusakan							
pengguna							
Perbaikan							
Loogut							

Gambar 3.45 *Designx Layout* Menu Bengkel.

26. Layout Menu Montir

Menu montir melihat tampilan yang telah diisi oleh super admin dan memiliki tombol tambah data, edit, hapus. Perancangan *layout* menu montir disajikan pada Gambar 3.46.

Home						
Admin						
Produsen						
Bengkel						
Montir						
Kerusakan						
pengguna						
Perbaikan						
Logout						

Gambar 3.46 *Design Layout* Menu Montir.

27. Layout Menu Pelanggan

Menu Pelanggan melihat tampilan yang telah diisi oleh super admin dan memiliki tombol hapus data saja. Perancangan *layout* menu pelanggan disajikan pada Gambar 3.47

Home						
Admin						
Produsen						
Bengkel						
Montir						
Kerusakan						
pengguna						
Perbaikan						
Logout						

Gambar 3.47 *Design Layout* Menu Pelanggan

28. *Layout* Menu Perbaikan

Menu perbaikan melihat tampilan yang telah diisi oleh pelanggan yang mengalami kerusakan dan memiliki tombol hapus. Perancangan *layout* menu keterangan disajikan pada Gambar 3.48.

Home							
Admin							
Produsen							
Bengkel							
Monitr	No	Tanggal	Pengguna	Telepon	keluhan	Montir	Hapus
Kerusakan							
pengguna							
Perbaikan							
Logout							

Gambar 3.48 *Design Layout* Menu Perbaikan

29. *Layout* Menu Tambah Data

Menu tambah data hampir disetiap menu yang ada di super admin, produsen, bengkel, montir, pelanggan. Perancangan *layout* menu tambah data disajikan pada Gambar 3.49.

Home
Admin
Produsen
Bengkel
Monitr
Kerusakan
pengguna
Perbaikan
Loogut

lihat data

nama Admin
 Password Admin

Simpan

Batal

Gambar 3.49 *Design Layout* Menu Tambah Data.

30. *Layout* Menu Edit Data

Menu edit data hampir disetiap menu yang ada di super admin, produsen, bengkel, montir, pelanggan. Perancangan *layout* menu tambah data disajikan pada Gambar 3.50.

Home
Admin
Produsen
Bengkel
Monitr
Kerusakan
pengguna
Perbaikan
Loogut

lihat data

nama Admin
 Password Admin

edit

Batal

Gambar 3.50 *Design Layout* Menu Edit Data

31. *Layout Menu Login Admin*

Bila sudah tampil menu login sebagai admin yang terdapat form *username* dan *password* kedua form itu wajib, lalu tekan tombol masuk bila. Perancangan *layout login* admin disajikan pada Gambar 3.51.

```

graph TD
    subgraph LOGIN
        direction TB
        U[username]
        P[password]
        M[masuk]
        B[batal]
    end
    subgraph HEADER
        H[HEADER]
    end
    LOGIN --- HEADER
  
```

Gambar 3.51 *Design Layout Login Admin*

32. *Layout Menu home*

Ketika admin sudah berhasil *login* akan tampil menu “*Home*” untuk memberi tahu bagaimana aplikasi ini berjalan. Perancangan *layout menu home* disajikan pada Gambar 3.52.

```

graph LR
    subgraph Menu
        direction TB
        H[Home]
        A[Admin]
        P[Produsen]
        B[Bengkel]
        M[Monitr]
        K[Kerusakan]
        U[pengguna]
        Per[Perbaikan]
        L[Logout]
    end
    subgraph Content
        C[logo Aplikasi]
    end
    Menu --- Content
  
```

Gambar 3.52 *Design Layout Menu Home.*

33. *Layout* Menu Profil

Ketika admin sudah berhasil *login* akan tampil menu “Profil” untuk menambahkan ubah data bengkel dan tersedia tombol simpan dan batal.

Perancangan *layout* menu profil disajikan pada Gambar 3.53.

The image shows a web application layout for an admin profile. On the left is a vertical sidebar menu with the following items: Home, Montir, Kerusakan, Perbaikan, Profil (highlighted in grey), Logout, and an empty box at the bottom. The main content area on the right contains a form for editing the profile. The form has the following fields: Nama Bengkel, Alamat Bengkel, Telepon Bengkel, Koordinat, Password Bengkel, and Foto. Each field is represented by a text label followed by a rectangular input box. At the bottom of the form are two buttons: 'Simpan' (Save) and 'Batal' (Cancel).

	Nama Bengkel	<input type="text"/>
Home	Alamat Bengkel	<input type="text"/>
Montir	Telepon Bengkel	<input type="text"/>
Kerusakan	Koordinat	<input type="text"/>
Perbaikan	Password Bengkel	<input type="text"/>
Profil	Foto	<input type="text"/>
Logout		
	<input type="button" value="Simpan"/>	<input type="button" value="Batal"/>

Gambar 3.53 *Design Layout* Menu Admin.

34. *Layout Menu Montir*

Ketika admin sudah berhasil *login* akan tampil menu “Montir” pada menu montir admin bisa menambahkan montir pada bengkel tersebut.

Perancangan *layout* menu montir disajikan pada Gambar 3.54.

Home					
Montir					
Kerusakan					
Perbaikan	No	Montir	Foto	Edit	Hapus
Profil					
Logout					

Gambar 3.54 *Design Layout Menu Montir*

35. *Layout Menu Kerusakan*

Pada menu kerusakan admin dapat menambahkan, ubah dan hapus kerusakan. Perancangan *layout* menu kerusakan disajikan pada Gambar 3.55.

Home				
Montir				
Kerusakan				
Perbaikan	No	Kerusakan	Edit	Hapus
Profil				
Logout				

Gambar 3.55 *Design Layout* Menu Kerusakan

36. *Layout* Menu Perbaikan

Menu perbaikan melihat tampilan yang telah diisi oleh pelanggan yang mengalami kerusakan dan memiliki tombol hapus. Perancangan *layout* menu perbaikan disajikan pada Gambar 3.56.

Home						
Montir						
Kerusakan						
Perbaikan	No	Tanggal	Pengguna	Telepon	Keluhan	Montir Hapus
Profil						
Logout						

Gambar 3.56 *Design Layout* Menu Perbaikan

3.5 Metode Pengujian Sistem

Metode pengujian sistem dalam penelitian ini adalah pengujian *Black Box* dengan metode *Equivalence Partitioning* (EP). Pengujian ini berguna untuk membuktikan semua fungsi-fungsi pada aplikasi berjalan dengan baik.

Pada pengujian ini diyakinkan bahwa masukan dan respon yang diterima sama sehingga terjadi kecocokan antara aplikasi dan pengguna. Metode ini dipilih karena metode ini dapat mencari kesalahan pada aplikasi, *interface* aplikasi dan kesalahan pada struktur data aplikasi. Rancangan daftar pengujian disajikan pada beberapa tabel dibawah ini:

Tabel 3.1 Pengujian Versi Android

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Versi Android	Pengujian kompatibilitas versi operating system Android	Pengujian pada Android Versi 5.0 (Lollipop)	Kompatibel dengan Android Versi 5.0 (Lollipop)
			Pengujian pada Android Versi 6.0 (Marshmallow)	Kompatibel dengan Android Versi 6.0 (Marshmallow)

Tabel 3.2 Pengujian *user interface*

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	<i>User Interface</i>	Pengujian Pada <i>Icon Bemo/pemo</i>	Klik <i>icon Bemo/pemo</i> pada perangkat Android pengguna	Menampilkan layout <i>splash screen</i>

Lanjutan tabel 3.2 pengujian interface

No	Kelas uji	Daftar pengujian	Scenario uji	Hasil yang diharapkan
2		Pengujian Pada Menu Utama Bemo/pemo	Klik <i>button</i> Menu “Profil”	Menampilkan <i>Layout</i> profil
			Klik <i>button</i> Menu “Montir”	Menampilkan <i>Layout</i> montir
			Klik <i>button</i> Menu “Kerusakan”	Menampilkan <i>Layout</i> kerusakan
			Klik <i>button</i> Menu “perbaikan”	Menampilkan <i>Layout</i> perbaikan
			Klik <i>button</i> Menu “data keluhan	Menampilkan <i>Layout</i> data keluhan

Tabel 3.3 Pengujian Fungsi Menu Aplikasi

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Fungsi Menu Profil	Pengujian Pada Menu profil	Klik <i>button</i> Menu “ubah profil”	Menampilkan <i>Layout</i> ubah profil
			Klik <i>button</i> Menu “ubah foto”	Menampilkan <i>Layout</i> ubah foto
			Klik <i>button</i> Menu “logout”	Menampilkan pemberitahuan <i>logout</i>
2	Fungsi Menu Montir	Pengujian Pada Menu Montir	Klik <i>button</i> Menu “hapus montir”	Menampilkan <i>Layout</i> hapusmontir
			Klik <i>button</i> Menu “ubah foto”	Menampilkan <i>Layout</i> ubah foto
			Klik <i>button</i> Menu “tambah montir”	Menampilkan <i>Layout</i> tambah montir
3	Fungsi Menu Kerusakan	Pengujian Pada Menu kerusakan	Klik <i>button</i> Menu “ubah kerusakan”	Menampilkan <i>Layout</i> ubah kerusakan
			Klik <i>button</i>	Menampilkan

			Menu “tambah kerusakan”	Layout tambah kerusakan
			Klik <i>button</i> Menu “hapus kerusakan”	Menampilkan Layout hapus kerusakan
4	Fungsi Menu perbaikan	Pengujian Pada Menu perbaikan	Klik <i>button</i> Menu “perbaikan”	Menampilkan Layout oerbaikan
			Klik <i>button</i> Menu “detail perbaikan”	Menampilkan Layout tdetail perbaikan
			Klik <i>button</i> Menu “ pilih tanggal perbaikan”	Menampilkan Layout tanggal perbaikan
			Klik <i>button</i> Menu “telepon”	Menampilkan Layout telepon
			Klik <i>button</i> Menu “Selesai di perbaiki”	Menampilkan Pemberitahuan selesai di perbaiki
5	Fungsi Menu data keluhan	Pengujian Pada Menu data keluhan	Klik <i>button</i> Menu “data keluhan pelanggan”	Menampilkan data keluhan
			Klik <i>button</i> Menu “pilih montir”	Menampilkan layout pilih montir
			Klik <i>button</i> Menu “cek lokasi”	Menampilkan layout maps
			Klik <i>button</i> Menu “data keluhan pelanggan”	Menampilkan data keluhan
			Klik <i>button</i> Refresh	Memuat Ulang rute menuju Wisata bila tidak tampil

Tabel 3.4 Pengujian *user interface*

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	<i>User Interface</i>	Pengujian Pada <i>Icon</i> pemo	Klik <i>icon</i> Bemo/pemo pada perangkat Android pelanggan	Menampilkan layout <i>splash screen</i>
		Pengujian Pada Menu Utama pemo	Klik <i>button</i> Menu “Profil”	Menampilkan <i>Layout</i> profil
			Klik <i>button</i> Menu “kerusakan”	Menampilkan <i>Layout</i> kerusakan
			Klik <i>button</i> Menu “perbaikan”	Menampilkan <i>Layout</i> perbaikan

Tabel 3.7 Pengujian Fungsi Menu Aplikasi

No	Kelas Uji	Daftar Pengujian	Skenario Uji	Hasil yang Diharapkan
1	Fungsi Menu profil	Pengujian Pada Menu profil	Klik <i>button</i> Ubah profil	Menampilkan <i>layout</i> ubah profil
			Klik <i>button</i> Menu “ubah produsen”	Menampilkan <i>layout</i> ubah produsen
			Klik <i>button</i> Menu “logout”	Menampilkan pemberitahuan logout

2	Fungsi Menu kerusakan	Pengujian Pada Menu kerusakan	Klik <i>button</i> pilih kerusakan	Menampilkan <i>layout</i> pilih kerusakan
			Klik <i>button</i> Menu “ detail kerusakan”	Menampilkan kolom detail kerusakan
			Klik <i>button</i> Menu “cek lokasi bengkel	Menampilkan lokasi bengkel
			Klik <i>button</i> bengkel	Menampilkan pemberitahuan kirim kerusakan
3	Fungsi Menu perbaikan	Pengujian Pada Menu perbaikan	Klik <i>button</i> Menu “perbaikan”	Menampilkan Bantuan
			Klik <i>button</i> Menu “detail perbaikan”	Menampilkan Detail perbaikan

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari penelitian yang telah dilakukan, dapat diambil simpulan sebagai berikut :

1. Telah berhasil dibangun aplikasi pencarian informasi bengkel mobil di Bandar Lampung berbasis *android* dan aplikasi berjalan *online* .
2. Aplikasi terbagi menjadi dua yaitu dalam bentuk *mobile* android untuk pengguna dan bengkel serta dalam bentuk *web* untuk Admin dan Super Admin.
3. Aplikasi pada bengkel terdapat lima menu utama home, profil, montir, kerusakan, *history*. Pada menu home bengkel dapat melakukan ubah foto, ubah nama bengkel dan ganti password. Menu montir dapat melakukan ubah nama montir, tambah montir dan hapus, sedangkan pada menu kerusakan bengkel dapat menambahkan kerusakan mobil dan menghapus kerusakan. Menu terakhir menu *history* dapat melakukan *view* dari transaksi dari bengkel.
4. Aplikasi pada pelanggan terdapat tiga menu utama profil, kerusakan, *history*. Pada menu profil pelanggan hanya dapat melakukan ubah nama, ubah telepon dan ubah produsen. Pada menu kerusakan adalah menu utama tersebut pihak pelanggan dapat melakukan laporan kerusakan pada pihak bengkel. pada menu *history* hampir sama dengan menu *history* pihak bengkel.

5. Hasil pengujian yang telah lakukan terhadap 20 responden mendapatkan hasil 93,2% dengan hasil sangat baik .dengan rata sangat setuju (SS) 13, setuju (S) 6,3 dan cukup setuju (CS) 0,7.

5.2 Saran

Berdasarkan perancangan dan hasil implementasi sistem yang dilakukan, maka saran yang perlu diperhatikan dalam mengembangkan aplikasi ini adalah sebagai berikut :

1. Aplikasi dapat dikembangkan kembali sehingga kompatibel dengan *platform* selain Android seperti iOS, *Windows Phone*.
2. Aplikasi dapat kompetibel dengan versi android di bawah 5.0(Lollipop)
3. Pada web super admin dan admin dapat di kembangkan kembali dan di tambahkan menu *back up* data.
4. Saat melakukan kirim kerusakan ke bengkel sebaiknya memberikan waktu tunggu.
5. Pada setiap aplikasi simbol nya dapat di perjelas dan bagus.

DAFTAR PUSTAKA

Andry. 2011. *Android A sampai Z*. PCplus, Jakarta.

Anhar. 2010. *Panduan Menguasai PHP dan Mysql*. Jakarta: Media Kita.

Clune, T.L., R.B. Rood. 2011. Software Testing and Verification In Climate Model Development. *IEEE Journal, Focus: Climate Change Software*. September-October, pp. 49-55.

Developers, Android. 2014. *Android Developers*. [Online]. Tersedia <http://developer.android.com/index.html>. Diakses pada tanggal 27Februari 2015.

Eclipse. 2014. *About the Eclipse Foundation*. [Online]. Tersedia :<https://www.eclipse.org/org/>. Diakses pada tanggal 2 Maret 2015.

Fowler, Martin. 2004. *UML Distilled Panduan Singkat Bahasa pemodelan Objek Standar, Edisi 3*. Andi Publishing, Yogyakarta.

Grady Booch. 1999. *Visual Modeling With Rational Rose 2000 And UML*. Halim.

J I., et al. 2011. Framework Pemetaan Data Berbasis Peta dengan Menggunakan Google Maps API (Skripsi). Universitas Bina Nusantara. Jakarta.

Jiang, F., Y. Lu. 2012. Software testing model selection research based on yin-yang testing theory. In: *IEEE Proceeding of International Conference on Computer Science and Information Processing (CISP)*, pp. 590-594.

Kroll, Per, Bruce MacIsaac. 2006. *Agility and Discipline Made Easy: Practices from OpenUP and RUP*. Pearson Education, Inc., Massachussets.

Larman, Craig. 2002. *Applying UML and Patterns: An Introduction to Object Oriented Analysis and Design and the Unified Process, 2nd Edition*. Prentice-Hall, Inc., New Jersey.

Lengkong, H.N., Alicia A.E Sinsuw., Arie S.M Lumenta. 2015. *Perancangan Penunjuk Rute Pada Kendaraan Pribadi Menggunakan Aplikasi Mobile GIS Berbasis Android yang Terintegrasi pada Google Maps*. E-Journal ISSN : 2301-8402. Unsrat. Manado

Lee, W. M. 2011. *Beginning Android Application Development*. Wiley Publishing, Inc.

Nazruddin, Safaat H. 2012. *(Edisi Revisi) Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android*. Informatika, Bandung.

Nazaruddin Safaat (2012). *Pemrograman Aplikasi Mobile, Smartphone, Dasar*. Jakarta : Penerbit Gunadarma

Prabawa, Prana Sabda. 2013. *Perancangan Aplikasi Pencarian Lokasi Fasilitas Pariwisata di Kota Bandung dan Implementasi Augmented Reality pada Platform Android*. (Skripsi). Universitas Komputer Indonesia. Bandung.

Prasetyo, Didik, Dwi ,2004, ***Solusi Pemrograman Berbasis Web Menggunakan PHP5***, PT. Elex Media Komputindo, Jakarta.

Pressman, Roger S. 2002 *Rekayasa Perangkat Lunak Pendekatan Praktisi* (Buku 1). Edisi 2. Andi : Yogyakarta.

Pressman, R.S. 2010. *Software Engineering: A Practitioner's Approach, 7th Edition*. McGraw-Hill, New York.

Uml-diagrams.org. 2014. *The Unified Modeling Language*. [Online]
Tersedia:<http://www.uml-diagrams.org/>. Diakses pada tanggal 20 Maret 2015.

Steiniger, S., Neun, M. and Edwardes, A. 2005. *Foundations of Location Based Services*. [online]. Tersedia :
http://www.spatial.cs.umn.edu/Courses/Fall11/8715/papers/IM7_steiniger.pdf

Wicaksono, Yogi. 2008. *Membangun Bisnis Online dengan Mambo*. Jakarta: PT. Elex Media Komputindo.