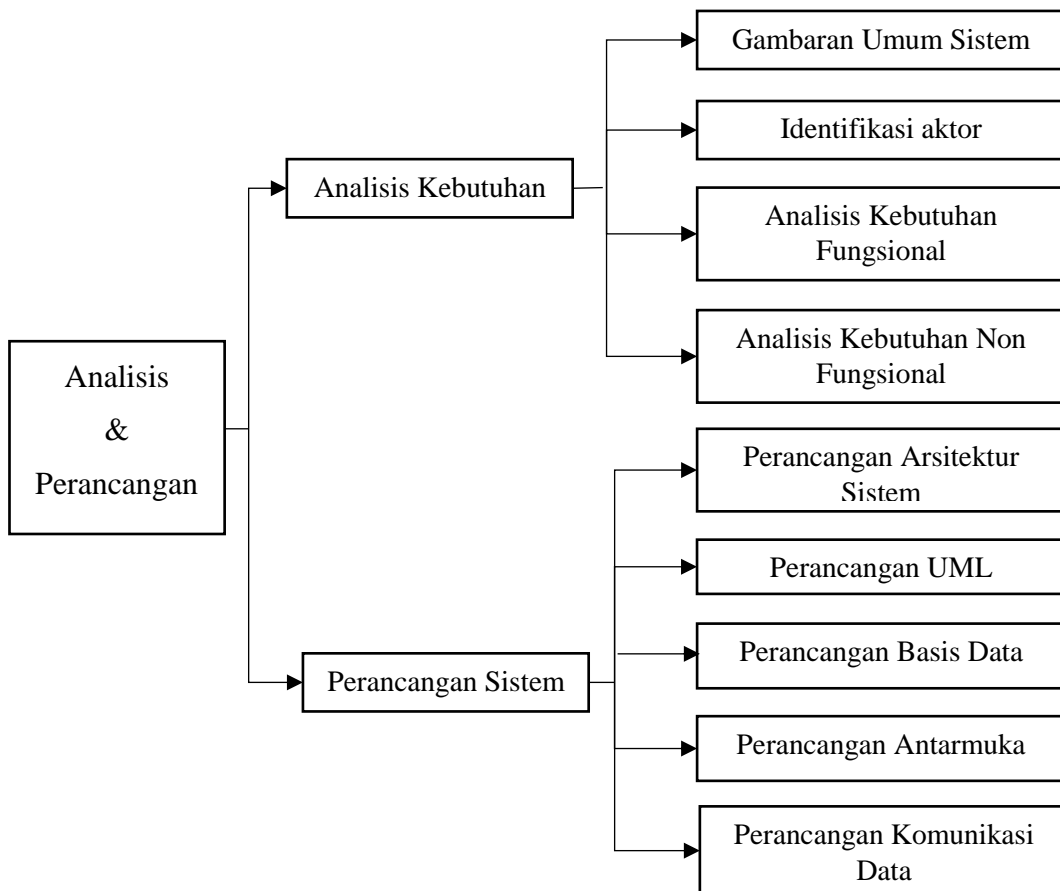


BAB 4 PERANCANGAN

Pada bab ini akan membahas mengenai perancangan aplikasi yang akan dibangun. Sistem yang akan menjalankan aplikasi ini dibangun dalam dua sisi dimana terdapat aplikasi mobile sebagai *front end* yang akan digunakan pengguna dan web base sistem informasi yang digunakan oleh admin. Perancangan yang dilakukan meliputi dua tahap. Tahap pertama adalah analisis sistem yang meliputi gambaran umum sistem dan kebutuhan dari sistem. Tahap kedua adalah perancangan aplikasi yang akan membahas lebih detail tentang perancangan aplikasi berdasarkan analisis kebutuhan yang telah dilakukan sebelumnya. Untuk lebih jelas mengenai perancangan yang dilakukan ditunjukkan dalam Gambar 4.1.



Gambar 4.1 Diagram Perancangan

4.1 Analisis kebutuhan

Proses analisis kebutuhan mengacu pada gambaran umum perangkat lunak pencarian dan pemberian bantuan pada permasalahan kendaraan serta hasil pengumpulan, pemahaman, dan penetapan kebutuhan – kebutuhan yang ingin didapatkan oleh pengguna. Proses analisis kebutuhan pertama akan diawali dengan penjabaran dari gambaran umum aplikasi, pengidentifikasian aktor yang terlibat, analisis data, penjelasan tentang daftar kebutuhan dan kemudian memodelkannya dalam diagram *use case*. Analisis kebutuhan ini ditujukan agar

dapat menggambarkan kebutuhan – kebutuhan yang harus disediakan oleh sistem agar dapat memenuhi kebutuhan pengguna.

4.1.1 Gambaran umum sistem

Gambaran umum dari sistem terdiri dari tiga bagian, yaitu deskripsi umum sistem, lingkungan sistem dan bisnis proses sistem.

4.1.1.1 Deskripsi sistem

Sistem pencarian dan pemberian bantuan pada permasalahan kendaraan ini merupakan sistem yang dirancang untuk perangkat bergerak pada sisi pengguna. Sistem ini dibuat untuk membantu pengendara kendaraan bermotor untuk mencari bantuan dalam mengatasi permasalahan kendaraan yang sedang menyimpannya. Sistem ini juga sebagai sarana pengguna untuk saling menolong pengguna atau pengendara lain yang membutuhkan pertolongan. Terdapat aktor administrator yang bertugas memverifikasi data yang ada pada sisi *server* yaitu data yang telah diinputkan pengguna melalui aplikasi pada perangkat bergerak atau *smartphone*.

Dengan menggunakan lokasi pencari bantuan dan lokasi pengguna lain yang dapat memberikan bantuan maka kecepatan bantuan tiba di lokasi atau keefektifan pemberian bantuan akan lebih baik dibandingkan mencari bantuan pada kerabat yang belum tentu lokasinya dekat dengan lokasi kejadian.

Bagi pengguna lain yang memberikan bantuan akan menjadi suatu kebanggaan tersendiri karena akan mendapatkan suatu kepuasan saat dapat memberikan bantuan kepada orang lain, serta akan diberikan suatu *badge* yang dapat ditunjukkan kepada orang lain yang dapat meningkatkan kebanggaan.

4.1.1.2 Lingkungan sistem

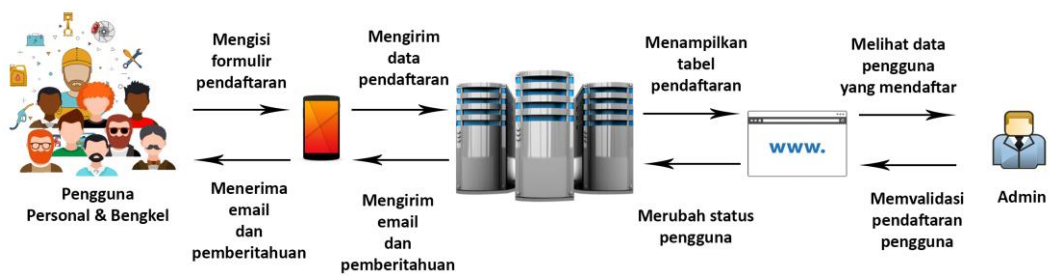
Aplikasi yang dibangun pada sistem ini terbagi menjadi dua bagian, yaitu aplikasi pada sisi *client* menggunakan perangkat *smartphone Android* dengan memanfaatkan fungsionalitas dari *smartphone* tersebut. Sedangkan aplikasi untuk Administrator berbasis *website* yang digunakan untuk manajemen data yang diinputkan oleh *client* serta berfungsi juga sebagai API server untuk penyedia layanan data dari aplikasi *client*.

4.1.1.3 Bisnis proses sistem

Bisnis proses aplikasi mobile untuk mencari dan memberikan bantuan pada permasalahan kendaraan dibagi menjadi 2 proses bisnis, yakni proses bisnis pendaftaran pengguna dan proses bisnis pencarian bantuan.

1. Bisnis proses pendaftaran pengguna

Gambaran umum proses pendaftaran pengguna secara garis besar digambarkan pada Gambar 4.2.



Gambar 4.2 Bisnis proses pendaftaran pengguna

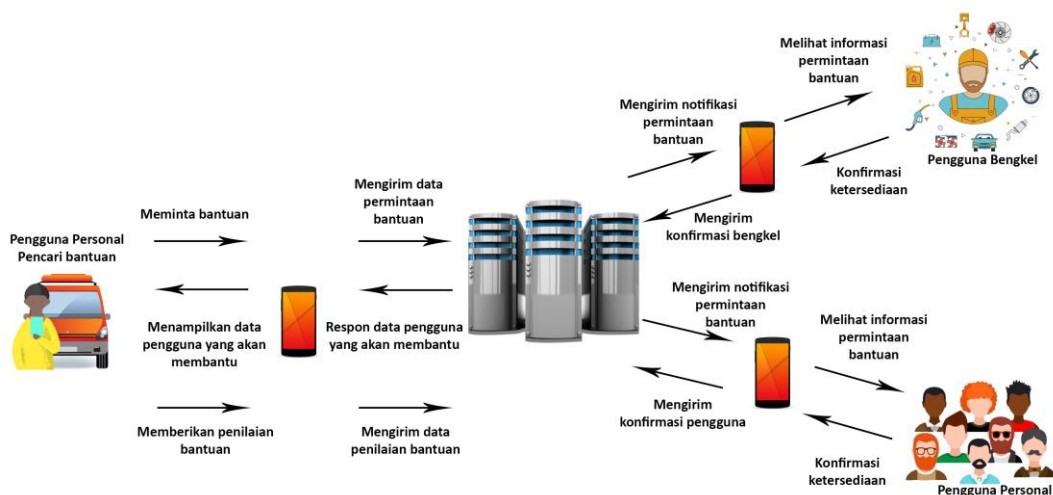
Untuk menggunakan aplikasi pengguna terlebih dahulu harus melakukan pendaftaran dengan mengisi lengkap formulir pada aplikasi dengan data data mengenai dirinya seperti nama, nomor telpon, ktp dan lainnya. Setelah itu data akan dikirim menuju *server* untuk disimpan.

Kemudian admin akan mendapatkan daftar pengguna yang melakukan pendaftaran pada web admin dan melihat data untuk dilakukan validasi. Apabila data dianggap telah valid maka admin menerima pendaftaran sehingga status pendaftaran berubah menjadi diterima dan sistem mengirimkan pemberitahuan kepada pengguna mengenai penerimaan pendaftarannya.

Namun apabila data dianggap tidak valid maka admin menolak pendaftaran dan merubah status pendaftar menjadi ditolak, sehingga pengguna akan mendapatkan pemberitahuan untuk melakukan perbaikan data pendaftaran.

2. Bisnis proses pencarian bantuan

Gambaran umum bisnis proses pencarian bantuan ditunjukkan pada gambar berikut.



Gambar 4.3 Bisnis proses pencarian bantuan

Dalam Gambar 4.3 digambarkan bahwa langkah pertama adalah mengajukan pencarian bantuan melalui aplikasi dengan mengirimkan data lokasi pencarian bantuan dan jenis bantuan menuju *server*.

Setelah itu *server* akan mencari pengguna lain yang berada pada sekitar lokasi dengan kriteria ketersediaan membantu jenis bantuan yang sesuai dan mengirimkan pemberitahuan kepada pengguna tersebut. Pengguna personal yang mendapatkan pemberitahuan akan dapat merespon mengenai ketersediaan membantu terhadap pencarian bantuan tersebut, sedangkan pengguna bengkel akan dapat memberikan respon bahwa dapat mendatangi lokasi atau tidak dan kemudian data dikirim menuju *server*.

Server kemudian akan mengirim data pengguna personal yang bersedia membantu dan pengguna bengkel dengan keterangan dapat mendatangi lokasi atau tidak kepada pencari bantuan. Setelah itu pencari bantuan akan mendapati daftar pemberi bantuan dan dapat memilih pemberi bantuan mana yang dikehendaki. Setelah memilih akan ditampilkan data pemberi bantuan sehingga pencari bantuan dapat menghubunginya.

Setelah proses pencarian bantuan dianggap selesai oleh pencari bantuan maka pencari bantuan dapat memberi nilai kepada pemberi bantuan yang nantinya akan diolah untuk menjadi *badge* milik pemberi bantuan.

4.1.2 Identifikasi aktor

Tahap ini adalah tahap untuk melakukan identifikasi terhadap aktor – aktor yang akan berinteraksi dengan sistem. Pada Tabel 4.1 menunjukkan aktor – aktor yang berhubungan dengan sistem.

Tabel 4.1 Identifikasi aktor

Aktor	Deskripsi
Pengguna Personal	<p>Pengguna personal adalah aktor yang akan menggunakan aplikasi <i>mobile</i>, dan tidak memiliki akses terhadap web admin pada <i>server</i>.</p> <p>Terdapat 2 jenis pengguna, yakni pengguna yang bertindak sebagai pencari bantuan dan pengguna yang bertindak sebagai pemberi bantuan. Pengguna yang sudah terdaftar dan telah terverifikasi dapat menggunakan aplikasi <i>mobile</i>.</p>
Pengguna Bengkel	<p>Pengguna bengkel adalah aktor yang akan menggunakan aplikasi <i>mobile</i>, dan tidak memiliki akses terhadap web admin pada <i>server</i>.</p> <p>Pengguna bengkel memiliki perbedaan fitur dari pengguna personal, dimana pengguna bengkel tidak dapat mencari bantuan dan memiliki kelebihan untuk melihat rekap kejadian pencarian bantuan disekitar bengkel miliknya.</p> <p>Pengguna bengkel yang sudah terdaftar dan telah terverifikasi dapat menggunakan aplikasi <i>mobile</i>.</p>

Tabel 4.1 Identifikasi aktor (lanjutan)

Administrator	Administrator atau admin merupakan aktor yang memiliki wewenang dalam sistem untuk mengelola data, baik untuk menambah, mengubah, dan menghapus data melalui web admin. Admin juga memiliki tugas untuk memverifikasi pendaftaran pengguna baru.
---------------	--

4.1.3 Analisis kebutuhan fungsional

Daftar kebutuhan terdiri dari kebutuhan fungsional dan non fungsional. Pada daftar kebutuhan fungsional akan dispesifikasikan yaitu spesifikasi kebutuhan fungsional dari pencari bantuan, pemberi bantuan, pengguna bengkel, dan admin. Daftar kebutuhan fungsional akan lebih dijabarkan menggunakan *use case diagram* dan skenario *use case*. Spesifikasi kebutuhan fungsional ditunjukkan pada Tabel 4.2.

Tabel 4.2 Daftar kebutuhan fungsional

Nomor	Kebutuhan	Use Case	Aktor
KF-001	Aplikasi dapat menjalankan mekanisme pendaftaran calon pengguna baru	Pendaftaran	Pengguna Personal, Pengguna Bengkel
KF-002	Aplikasi dapat mengajukan pencarian bantuan dengan visualisasi peta lokasi tempat pengguna saat ini berada	Pencarian Bantuan	Pengguna Personal
KF-003	Aplikasi dapat menampilkan daftar pengguna lain yang dapat membantu dan bengkel yang ada disekitarnya yang memungkinkan pencari bantuan memilih pengguna/bengkel yang diinginkan untuk membantunya	Pemilihan pemberi bantuan	Pengguna Personal

Tabel 4.2 Daftar kebutuhan fungsional (lanjutan)

Nomor	Kebutuhan	Use Case	Aktor
KF-004	Aplikasi mampu menampilkan profile pemberi bantuan	Lihat profile pemberi bantuan	Pengguna Personal
KF-005	Aplikasi dapat mengubah status pencarian bantuan dan memberikan rating terhadap pemberi bantuan	Akhiri pencarian bantuan	Pengguna Personal
KF-006	Aplikasi dapat menerima pemberitahuan saat terdapat pengguna lain mencari bantuan dan dapat melakukan konfirmasi terhadap pencarian bantuan tersebut	Pemberitahuan pencarian bantuan	Pengguna Personal, Pengguna Bengkel
KF-007	Aplikasi mampu menampilkan <i>badge</i> yang diperoleh dan memungkinkan pengguna menyebarkannya	Lihat badge	Pengguna Personal, Pengguna Bengkel
KF-008	Aplikasi dapat menampilkan statistik pencarian bantuan dan jenis permasalahan yang sering terjadi disekitar	Lihat statistik pencarian bantuan	Pengguna Bengkel
KF-009	Aplikasi memungkinkan pengguna untuk mengatur jenis pemberian bantuan yang dapat ditangani	Konfigurasi jenis bantuan	Pengguna Personal, Pengguna Bengkel

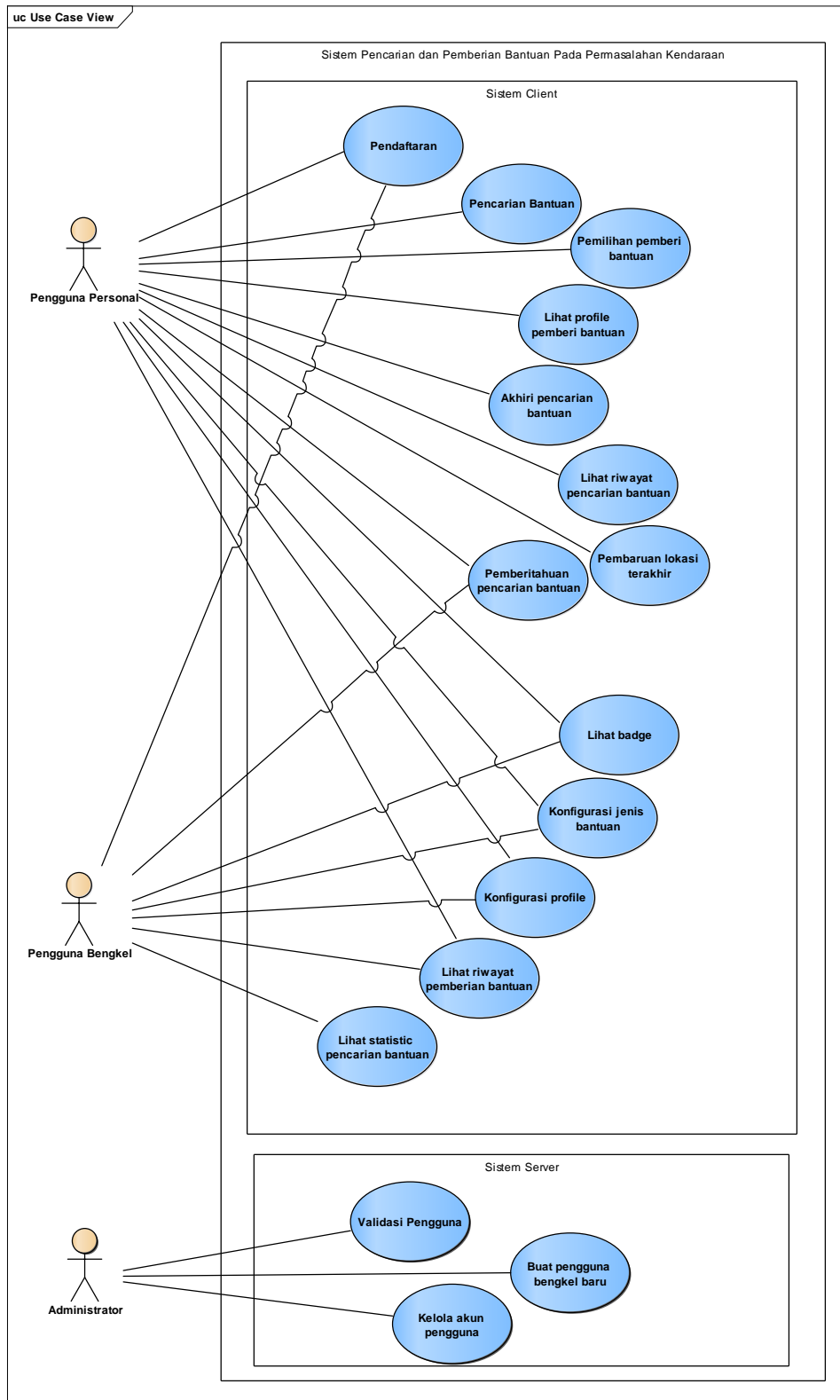
Tabel 4.2 Daftar kebutuhan fungsional (lanjutan)

Nomor	Kebutuhan	Use Case	Aktor
KF-010	Aplikasi memungkinkan menampilkan form yang dapat digunakan pengguna untuk merubah profilnya	Konfigurasi profile	Pengguna Personal, Pengguna Bengkel
KF-011	Aplikasi dapat menampilkan riwayat pencarian bantuan yang pernah dilakukan pengguna	Lihat riwayat pencarian bantuan	Pengguna Personal
KF-012	Aplikasi dapat menampilkan riwayat pemberian bantuan yang pernah dilakukan pengguna	Lihat riwayat pemberian bantuan	Pengguna Personal, Pengguna Bengkel
KF-013	Aplikasi dapat melakukan pembaruan lokasi pengguna secara berkala	Pembaruan lokasi terakhir	Pengguna Personal
KF-014	Sistem <i>server</i> dapat memvalidasi pendaftaran pengguna yang dilakukan melalui aplikasi <i>client</i>	Validasi Pengguna	Admin
KF-015	Sistem <i>server</i> dapat mendaftarkan pengguna bengkel baru	Buat pengguna bengkel baru	Admin
KF-016	Sistem <i>server</i> dapat melihat dan melakukan perubahan terhadap profile atau akun dari pengguna	Kelola akun pengguna	Admin

4.1.3.1 Use Case diagram

Pada sistem ini untuk memodelkan aspek perilaku dari sistem akan digunakan pemodelan dengan *use case* diagram. Terdapat 3 aktor dalam sistem ini yaitu Pengguna personal, Pengguna bengkel dan Admin. Aktor Pengguna personal dan Pengguna bengkel hanya dapat mengakses aplikasi *client* yang berbasis perangkat *smartphone*, sedangkan Admin hanya dapat mengakses aplikasi pada sistem

server yang berupa sebuah Web. *Use case* diagram pada sistem aplikasi ditunjukkan pada Gambar 4.4.



Gambar 4.4 Diagram *Use Case*

4.1.3.2 Skenario *Use Case*

Skenario *use case* digunakan untuk menjelaskan tiap *use case* yang terdapat pada gambar Gambar 4.4. Skenario akan menguraikan mengenai nama *use case*, deskripsi global *use case*, kondisi awal yang harus dipenuhi dan kondisi akhir yang diharapkan setelah berjalannya fungsional *use case* dan juga akan diberikan ulasan yang berkaitan dengan tanggapan dari sistem atas suatu aksi yang diberikan oleh actor. Berikut ini scenario *use case* dalam sistem aplikasi ini.

1. Skenario *Use Case* pendaftaran

Kebutuhan fungsional disediakan pada sistem adalah kebutuhan untuk melakukan pendaftaran. Kebutuhan tersebut direpresentasikan oleh *use case* pendaftaran, Tabel 4.3 merupakan scenario *use case* pendaftaran.

Tabel 4.3 Skenario *Use Case* pendaftaran

Nomor <i>Use Case</i>	KF-001
Nama <i>Use Case</i>	Pendaftaran
Tujuan	Melakukan pendaftaran pengguna
Deskripsi (<i>Brief Description</i>)	<i>Use case</i> ini memungkinkan pengguna untuk mengajukan pendaftaran pengguna untuk dapat menggunakan fitur dari aplikasi dengan mengisi lengkap data data yang diperlukan.
Aktor	Pengguna Personal, Pengguna Bengkel
Kondisi Awal (<i>Pre-Conditions</i>)	Aktor harus belum melakukan <i>login</i> . Aktor harus membuka aplikasi dan masuk halaman pendaftaran.
<i>Flow of Events</i>	
Alur Utama (<i>Basic Flow</i>)	
<ol style="list-style-type: none">1. Aktor mengisi data data lengkap pada formulir yang berisi foto diri, nama lengkap, tanggal lahir, nomor telpon, alamat, nomor identitas, foto kartu identitas, alamat tinggal, jenis kelamin.2. Aktor memilih jenis kategori pengguna.	

Tabel 4.3 Skenario *Use Case* pendaftaran (lanjutan)

<p>3. Apabila aktor memilih jenis bengkel maka diperlukan pengisian data lebih lanjut berupa alamat bengkel dan koordinat bengkel pada peta.</p> <p>4. Aktor menekan tombol simpan pendaftaran.</p> <p>5. Data pencarian bantuan dikirimkan menuju <i>server</i>.</p>	
Kondisi Akhir (Post-Conditions)	Aplikasi menampilkan tampilan menunggu respon proses validasi.

2. Skenario *Use Case* pencarian bantuan

Kebutuhan fungsional yang disediakan pada sistem adalah kebutuhan untuk melakukan pencarian bantuan. Kebutuhan tersebut direpresentasikan oleh *use case* pencarian bantuan, Tabel 4.4 merupakan skenario *use case* pencarian bantuan.

Tabel 4.4 Skenario *Use Case* pencarian bantuan

Nomor <i>Use Case</i>	KF-002
Nama <i>Use Case</i>	Pencarian bantuan
Tujuan	Mencari bantuan
Deskripsi (Brief Description)	<i>Use case</i> ini memungkinkan pengguna untuk mengajukan pencarian bantuan yang kemudian akan diajukan kepada pengguna lain yang ada disekitarnya.
Aktor	Pengguna Personal
Kondisi Awal (Pre-Conditions)	Aktor harus membuka dan melakukan <i>login</i> pada aplikasi terlebih dahulu sebelum <i>use case</i> dimulai. Aktor harus mengaktifkan fitur GPS pada <i>smartphone</i> miliknya. Aktor harus masuk pada halaman utama aplikasi.
Flow of Events	
Alur Utama (Basic Flow)	

Tabel 4.4 Skenario *Use Case* pencarian bantuan (lanjutan)

<ol style="list-style-type: none"> 1. Aktor mengatur lokasi keberadaannya pada map dengan memposisikan tanda sesuai lokasinya bila posisi pada awal ditunjukkan kurang tepat. 2. Aktor memilih jenis kendaraannya. 3. Aktor memilih jenis permasalahan pada kendaraaaannya yang hendak dicarikan bantuan. 4. Data pencarian bantuan dikirimkan menuju <i>server</i>. 	
Kondisi Akhir (<i>Post-Conditions</i>)	Aplikasi menampilkan tampilan menunggu respon pemberi bantuan.

3. Skenario *Use Case* pemilihan pemberi bantuan

Kebutuhan fungsional yang disediakan pada sistem adalah kebutuhan untuk melakukan pemilihan pemberi bantuan. Kebutuhan tersebut direpresentasikan oleh *use case* pemilihan pemberi bantuan. Tabel 4.5 merupakan skenario *use case* pemilihan pemberi bantuan.

Tabel 4.5 Skenario *Use Case* pemilihan pemberi bantuan

Nomor <i>Use Case</i>	KF-003
Nama <i>Use Case</i>	Pemilihan pemberi bantuan
Tujuan	Memilih pemberi bantuan
Deskripsi (<i>Brief Description</i>)	<i>Use case</i> ini memungkinkan pengguna memilih pemberi bantuan yang telah merespon permintaan pencarian bantuannya dan mendapatkan detail informasi pengguna pemberi bantuan yang dipilihnya.
Aktor	Pengguna Personal
Kondisi Awal (<i>Pre-Conditions</i>)	Aktor harus membuka dan melakukan <i>login</i> pada aplikasi terlebih dahulu sebelum <i>use case</i> dimulai. Aktor harus melakukan pencarian bantuan terlebih dahulu.

Tabel 4.5 Skenario *Use Case* pemilihan pemberi bantuan (lanjutan)

Flow of Events	
Alur Utama (<i>Basic Flow</i>)	
<ol style="list-style-type: none"> 1. Aktor mendapatkan daftar pengguna lain yang merespon pencarian bantuannya. 2. Aplikasi menampilkan daftar pemberi bantuan dengan mengategorikan pengguna personal dan bengkel. 3. Aktor memilih pemberi bantuan dengan memilih nama dari daftar tersebut. 4. Data pemberi bantuan yang dipilih diteruskan menuju <i>server</i>. 	
Kondisi Akhir (<i>Post-Conditions</i>)	<p>Sistem mengubah status pencarian bantuan menjadi sedang menerima bantuan.</p> <p>Aplikasi menampilkan data profile pemberi bantuan yang dipilihnya.</p>

4. Skenario *Use Case* akhiri pencarian bantuan

Kebutuhan fungsional yang disediakan pada sistem adalah kebutuhan untuk mengakhiri pencarian bantuan. Kebutuhan tersebut direpresentasikan oleh *use case* akhiri pencarian bantuan, Tabel 4.6 merupakan skenario *use case* akhiri pencarian bantuan.

Tabel 4.6 Skenario *Use Case* akhiri pencarian bantuan

Nomor <i>Use Case</i>	KF-005
Nama <i>Use Case</i>	Akhiri pencarian bantuan
Tujuan	Mengakhiri proses pencarian bantuannya.
Deskripsi (<i>Brief Description</i>)	<i>Use case</i> ini memungkinkan pengguna mengakhiri proses pencarian bantuan dan memberikan penilaian terhadap pemberi bantuannya.
Aktor	Pengguna Personal

Tabel 4.6 Skenario *Use Case* akhiri pencarian bantuan (lanjutan)

Kondisi Awal (Pre-Conditions)	Aktor harus membuka dan melakukan <i>login</i> pada aplikasi terlebih dahulu sebelum <i>use case</i> dimulai. Aktor harus melakukan pencarian bantuan terlebih dahulu.
Flow of Events	
Alur Utama (Basic Flow)	
<ol style="list-style-type: none"> 1. Aktor menekan tombol mengakhiri bantuan 2. Aktor memberi penilaian terhadap pemberi bantuan berupa bintang 3. Data penilaian dikirim menuju <i>server</i> 	
Kondisi Akhir (Post-Conditions)	Sistem mengubah status pencarian bantuan menjadi selesai. Aplikasi menampilkan pemberitahuan bahwa proses pencarian bantuan telah selesai.

5. Skenario *Use Case* pemberitahuan pencarian bantuan

Kebutuhan fungsional yang disediakan pada sistem adalah kebutuhan untuk mendapatkan pemberitahuan pencarian bantuan. Kebutuhan tersebut direpresentasikan oleh *use case* pemberitahuan pencarian bantuan, Tabel 4.7 merupakan skenario *use case* pemberitahuan pencarian bantuan.

Tabel 4.7 Skenario *Use Case* pemberitahuan pencarian bantuan

Nomor <i>Use Case</i>	KF-006
Nama <i>Use Case</i>	Pemberitahuan pencarian bantuan
Tujuan	Pengguna mendapatkan pemberitahuan dan merespon untuk memberi bantuan
Deskripsi (Brief Description)	<i>Use case</i> ini memungkinkan pengguna mendapatkan pemberitahuan saat terjadi pencarian bantuan disekitarnya, dan bagaimana pengguna merespon pencarian bantuan tersebut.

Tabel 4.7 Skenario *Use Case* pemberitahuan pencarian bantuan (lanjutan)

Aktor	Pengguna Personal, Pengguna Bengkel
Kondisi Awal (Pre-Conditions)	Aktor harus melakukan <i>login</i> pada aplikasi terlebih dahulu sebelum <i>use case</i> dimulai. Aktor yang berupa pengguna personal sedang tidak dalam proses pemberian bantuan lain.
Flow of Events	
Alur Utama (Basic Flow)	
<ol style="list-style-type: none"> 1. Aktor menekan notifikasi pada <i>smartphone</i> miliknya 2. Aplikasi membuka halaman respon bantuan 3. Aktor yang berupa pengguna personal memilih untuk membantu atau tidak, sedangkan actor pengguna bengkel memilih dapat datang kelokasi atau tidak. 4. Apabila dalam 1 menit notifikasi diabaikan maka dianggap actor memilih opsi tidak dalam merespon bantuan 5. Respon dikirimkan menuju <i>server</i> 	
Kondisi Akhir (Post-Conditions)	Sistem mengubah status respon pemberi bantuan. Aplikasi menampilkan pesan proses pengiriman data pengajuan pemberian bantuan dari <i>server</i> .

6. Skenario *Use Case* lihat statistik pencarian bantuan

Kebutuhan fungsional yang disediakan pada sistem adalah kebutuhan untuk melihat statistik pencarian bantuan. Kebutuhan tersebut direpresentasikan oleh *use case* lihat statistik pencarian bantuan, Tabel 4.8 merupakan skenario *use case* lihat statistik pencarian bantuan.

Tabel 4.8 Skenario *Use Case* lihat statistik pencarian bantuan

Nomor <i>Use Case</i>	KF-008
Nama <i>Use Case</i>	Lihat statistik pencarian bantuan

Tabel 4.8 Skenario Use Case lihat statistik pencarian bantuan (lanjutan)

Tujuan	Pengguna dapat mengetahui statistik pencarian bantuan disekitarnya
Deskripsi (Brief Description)	<i>Use case</i> ini memungkinkan pengguna mengetahui data mengenai pencarian bantuan yang pernah terjadi disekitarnya.
Aktor	Pengguna Bengkel
Kondisi Awal (Pre-Conditions)	Aktor harus membuka dan melakukan <i>login</i> pada aplikasi terlebih dahulu sebelum <i>use case</i> dimulai. Aktor harus berada pada halaman statistik.
Flow of Events	
Alur Utama (Basic Flow)	
<ol style="list-style-type: none"> 1. Aktor memilih range tanggal statistik 2. Aplikasi mengirim permintaan data statistik dengan range tanggal yang dipilih 3. Aktor mendapatkan data statistik dari <i>server</i> 4. Aplikasi menampilkan data statistik 	
Kondisi Akhir (Post-Conditions)	Aplikasi menampilkan data dalam bentuk lokasi pada map, grafik banyak pencarian bantuan perharinya, dan persentase jenis bantuan yang dicari.

7. Skenario Use Case validasi pengguna

Kebutuhan fungsional yang disediakan pada sistem adalah kebutuhan untuk melakukan validasi pendaftaran pengguna. Kebutuhan tersebut direpresentasikan oleh *use case* validasi pengguna, Tabel 4.9 merupakan skenario *use case* validasi pengguna.

Tabel 4.9 Skenario *Use Case* validasi pengguna

Nomor Use Case	KF-014
Nama Use Case	Validasi pengguna
Tujuan	Melakukan validasi pendaftaran pengguna
Deskripsi (Brief Description)	<i>Use case</i> ini memungkinkan untuk menerima atau menolak pendaftaran pengguna
Aktor	Admin
Kondisi Awal (Pre-Conditions)	Aktor harus membuka dan melakukan <i>login</i> pada <i>website</i> terlebih dahulu sebelum <i>use case</i> dimulai. Aktor daftar pengajuan pendaftaran.
Flow of Events	
Alur Utama (Basic Flow)	
<ol style="list-style-type: none"> 1. Aktor memilih pengguna yang mengajukan pendaftaran 2. Aktor melakukan pengecekan data pengguna 3. Jika data valid actor memilih menerima pendaftaran pengguna 4. Sistem mengirimkan email dan pemberitahuan 	
Kondisi Akhir (Post-Conditions)	Status pengguna menjadi aktif atau tidak aktif.

4.1.4 Analisis kebutuhan non fungsional

Analisis kebutuhan non fungsional adalah analisis untuk mengetahui spesifikasi yang dibutuhkan oleh sistem. Sistem pencarian dan pemberian bantuan ini lebih menekankan pada bagaimana rancangan dan implementasi aplikasi pencarian dan pemberian bantuan dapat memberikan kemudahan dan manfaat penggunaan dari sistem terhadap pencarian bantuan saat terjadi permasalahan pada kendaraannya. Pengujian yang digunakan untuk mengukur kepuasan tersebut ialah pengujian *usability* yang diharapkan dari hasil pengujian tersebut dapat melebihi target. Jadi analisis kebutuhan non fungsional dari sistem rekomendasi ini ialah *usability* dengan target penilaian lebih dari (\geq) 80, sebab syarat

diterimanya sebuah sistem dari hasil pengujian *usability* menggunakan skor SUS ialah rentang nilai antara 68-100.

4.2 Perancangan sistem

Proses perancangan sistem dibagi menjadi empat bagian, yaitu perancangan UML, perancangan arsitektur sistem, perancangan basis data, dan perancangan antar muka. Perancangan sistem pada penelitian ini menggunakan pendekatan berorientasi objek yang direpresentasikan dengan menggunakan UML.

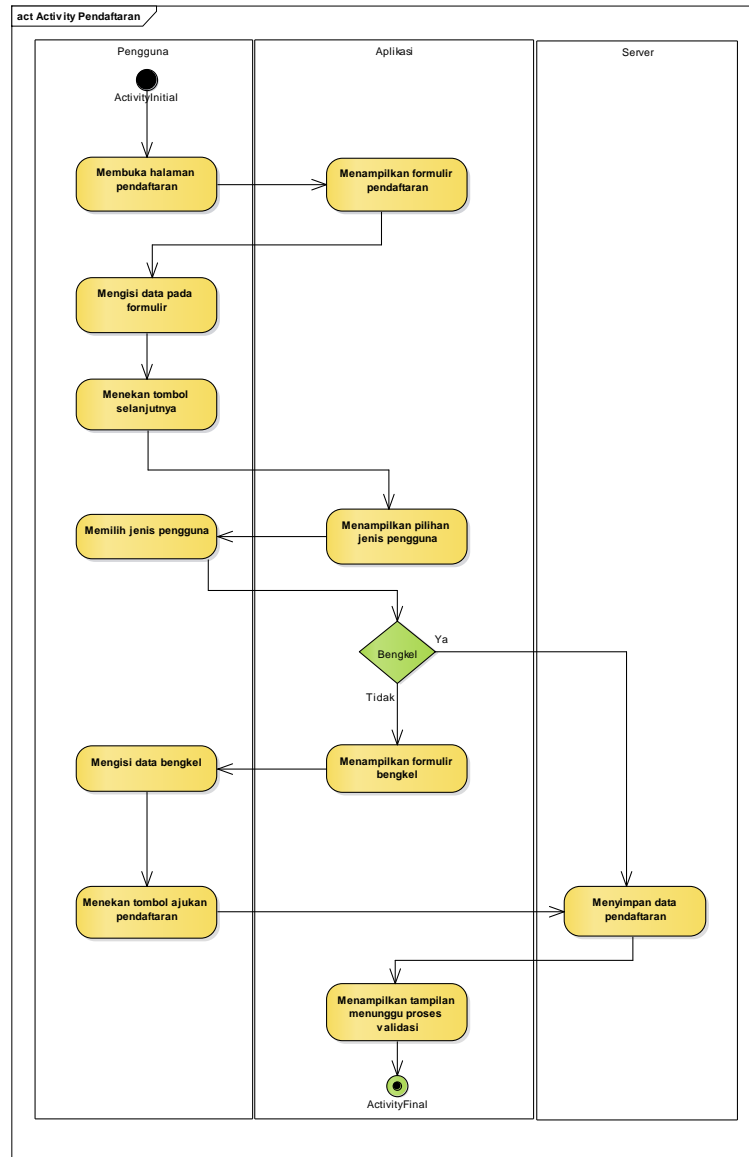
4.2.1 Perancangan UML

4.2.1.1 Perancangan *Activity Diagram*

Activity diagram digunakan untuk memodelkan aktifitas antara pengguna dan sistem yang berjalan berdasarkan pada skenario *use case* yang telah dibuat. Berikut ialah salah satu *activity diagram* pada sistem pencarian dan pemberian bantuan terhadap permasalahan kendaraan.

1. *Activity Diagram* proses pendaftaran

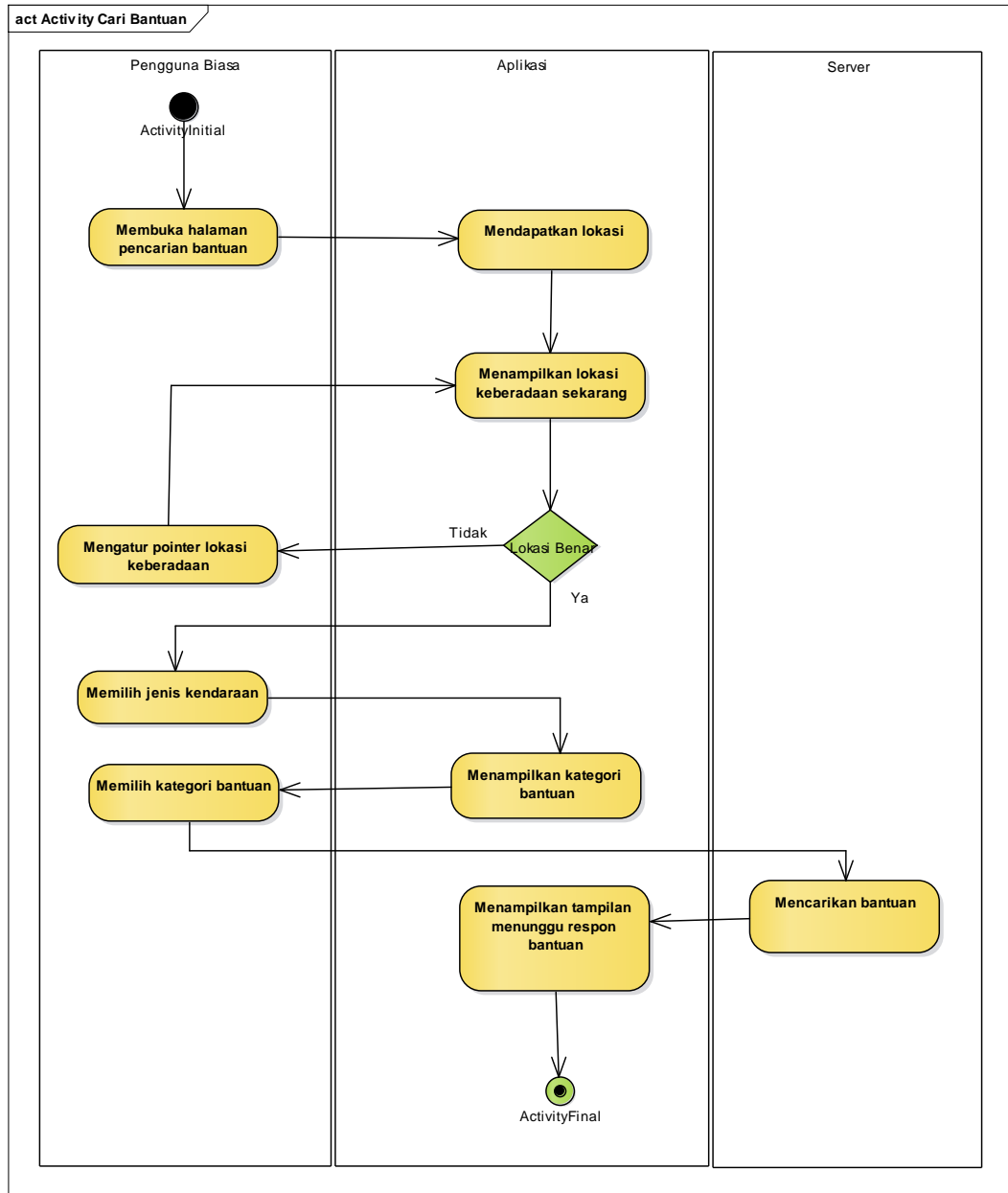
Pada Gambar 4.5 menjelaskan alur dalam mengajukan pendaftaran menggunakan aplikasi, pada saat membuka halaman pendaftaran pengguna dianggap belum melakukan *login*.



Gambar 4.5 Activity Diagram proses pendaftaran

2. Activity Diagram proses pencarian bantuan

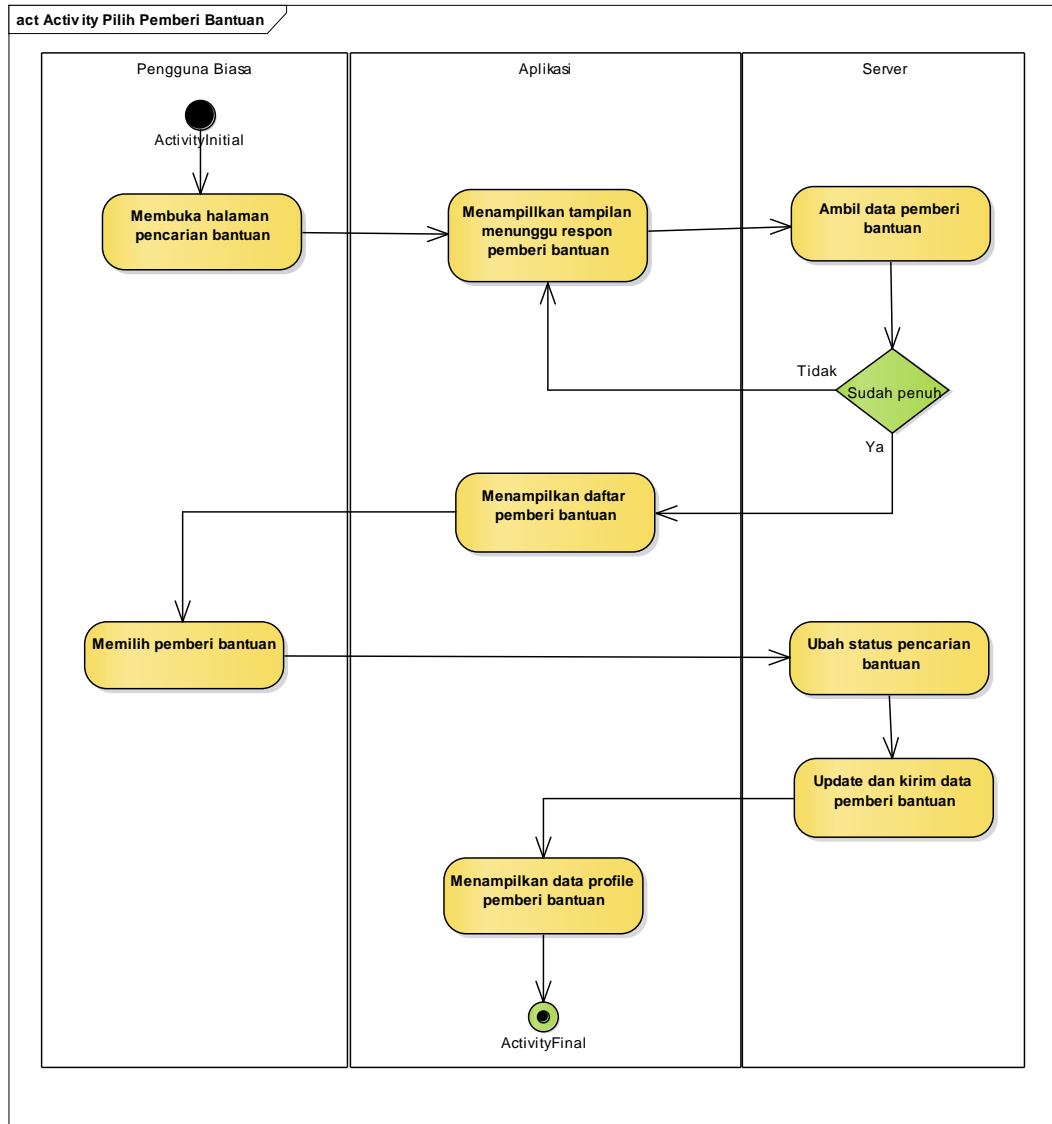
Pada Gambar 4.6 menjelaskan alur dalam membuat laporan pencarian bantuan menggunakan aplikasi, pada saat membuka halaman pencarian bantuan pengguna dianggap telah melakukan *login*.



Gambar 4.6 Activity Diagram prosen pencarian bantuan

3. Activity Diagram proses pemilihan pemberi bantuan

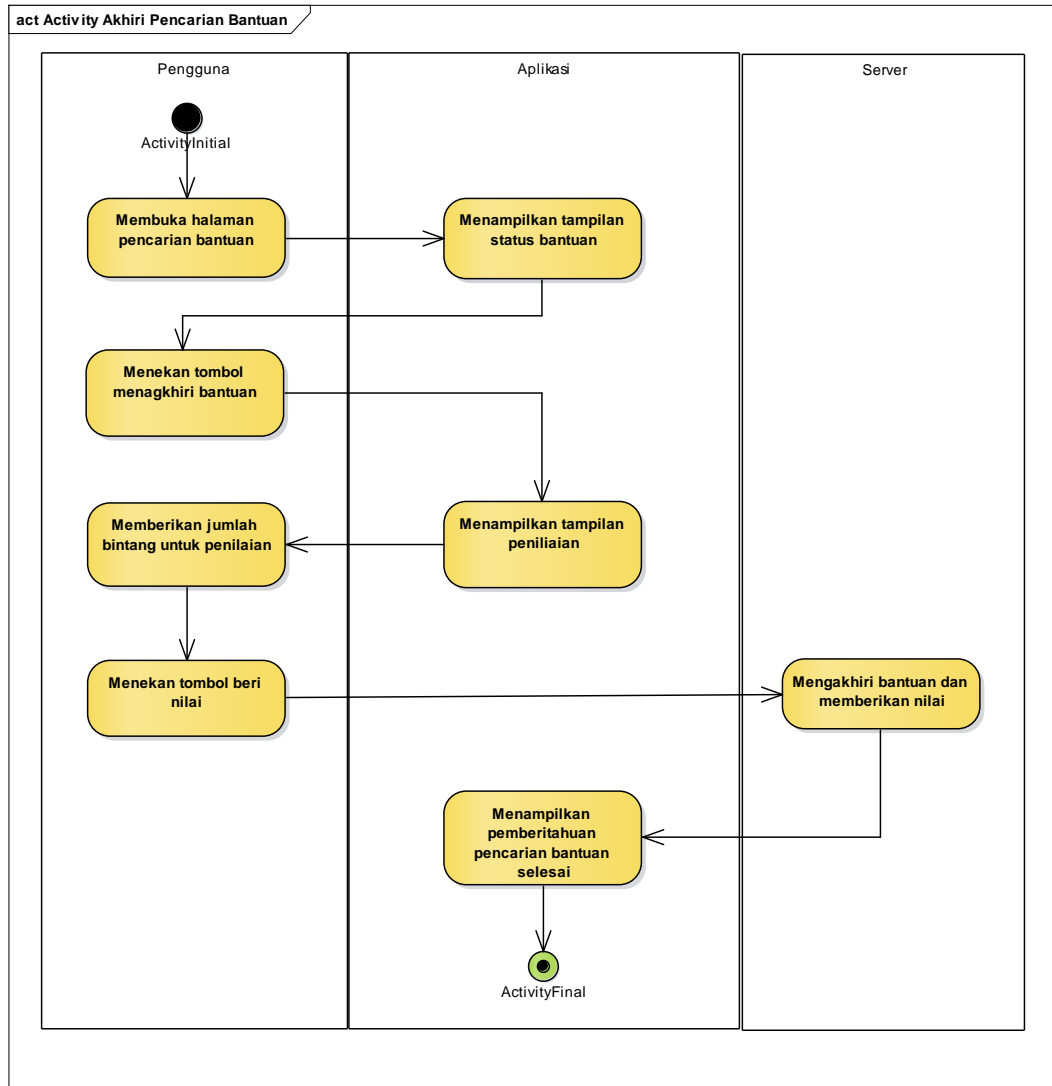
Pada Gambar 4.7 menjelaskan alur dalam memilih pemberi bantuan menggunakan aplikasi, pada saat membuka halaman pencarian bantuan pengguna dianggap telah melakukan *login* dan sedang dalam proses pencarian bantuan.



Gambar 4.7 Activity Diagram proses pemilihan pemberi bantuan

4. Activity Diagram proses akhiri pencarian bantuan

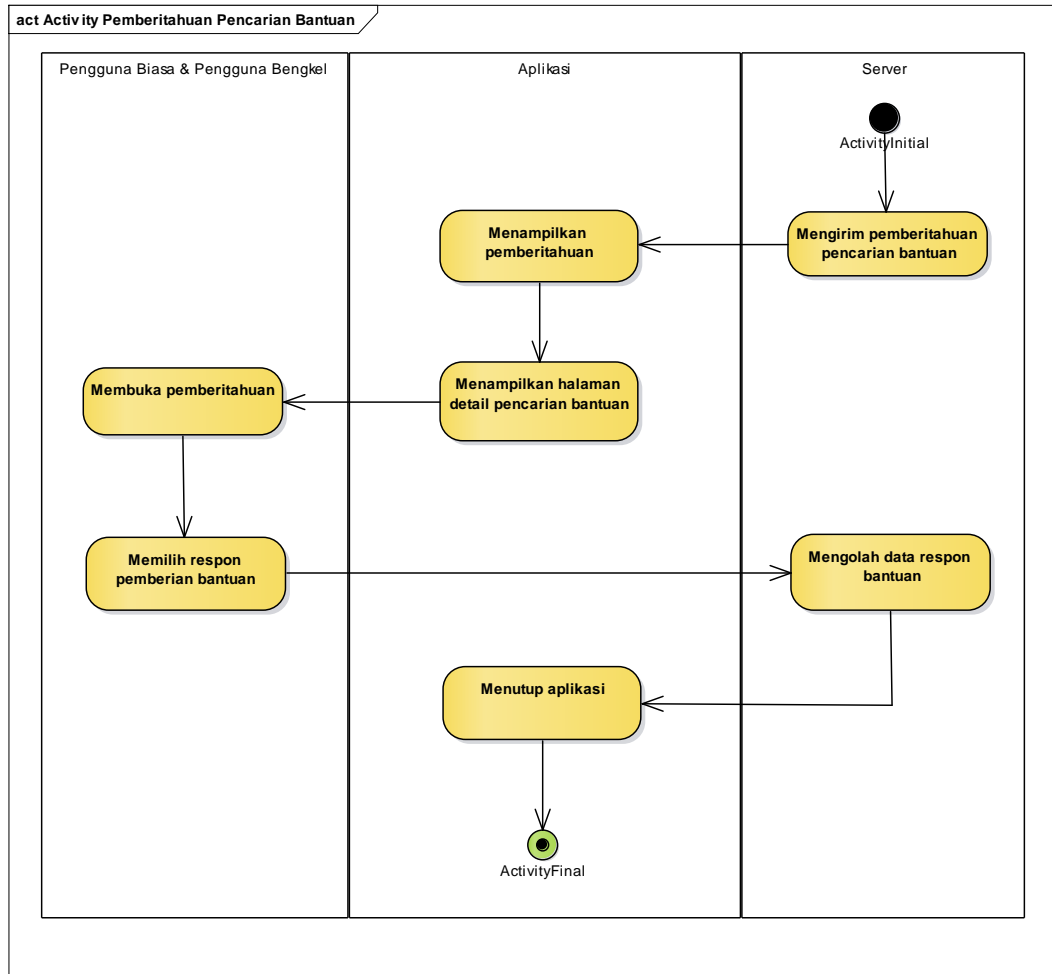
Pada Gambar 4.8 menjelaskan alur dalam mengakhiri proses pencarian bantuan menggunakan aplikasi, pada saat membuka halaman pencarian bantuan pengguna dianggap telah melakukan *login* dan sedang dalam proses pencarian bantuan.



Gambar 4.8 Activity Diagram proses akhiri pencarian bantuan

5. Activity Diagram proses pemberitahuan pencarian bantuan

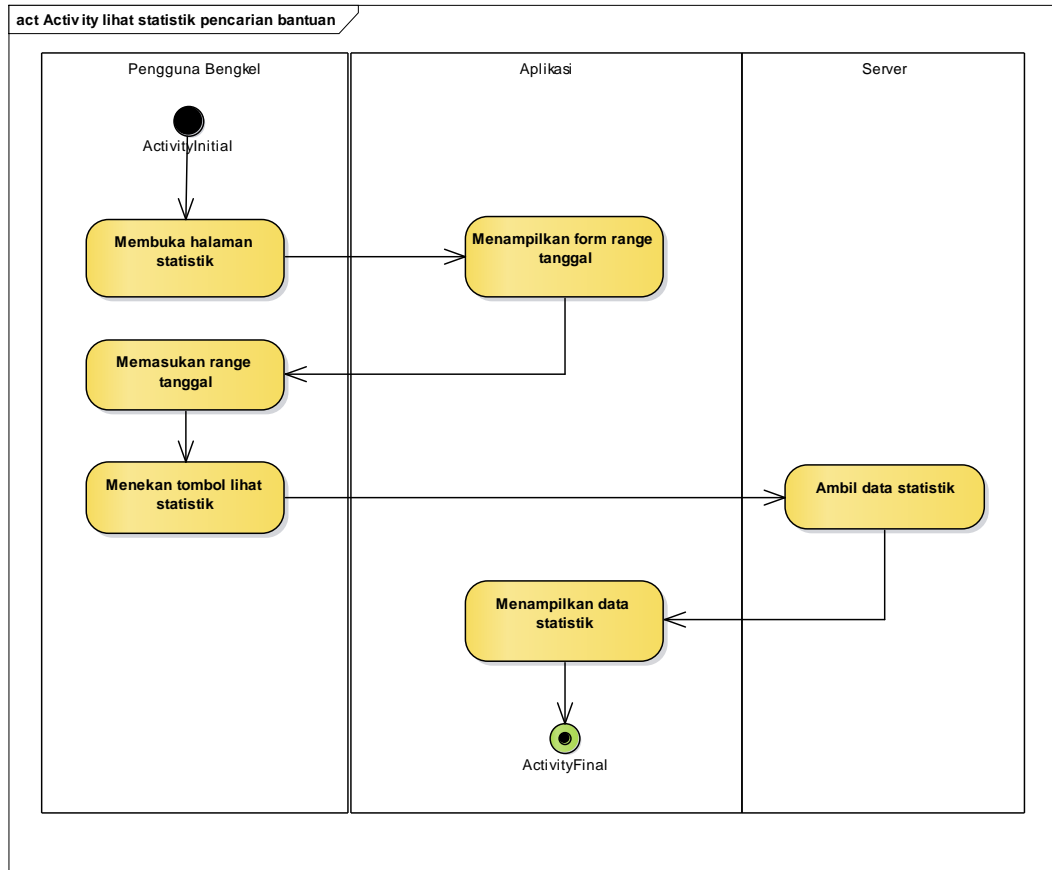
Pada Gambar 4.9 menjelaskan alur dalam menerima laporan pencarian bantuan menggunakan aplikasi, pada saat menampilkan pemberitahuan pengguna dianggap telah melakukan *login*.



Gambar 4.9 Activity Diagram proses pemberitahuan pencarian bantuan

6. Activity Diagram proses lihat statistik pencarian bantuan

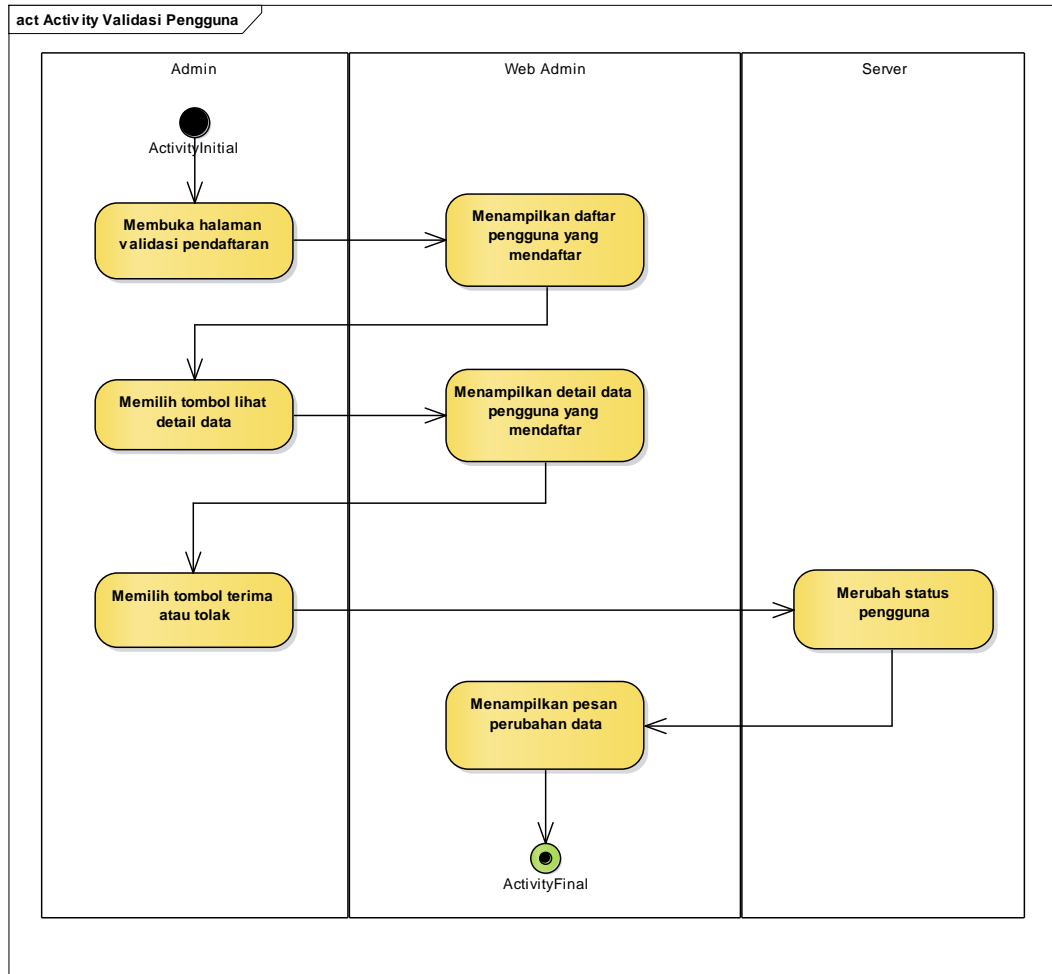
Pada Gambar 4.10 menjelaskan alur dalam melihat statistik pencarian bantuan yang terjadi di dekat bengkel menggunakan aplikasi, pada saat membuka halaman statistik pengguna dianggap telah melakukan *login*.



Gambar 4.10 Activity Diagram proses lihat statistik pencarian bantuan

7. Activity Diagram proses validasi pengguna

Pada Gambar 4.11 menjelaskan alur dalam memvalidasi pendaftaran pengguna menggunakan *website* admin, pada saat membuka halaman validasi pendaftaran admin dianggap telah melakukan *login*.



Gambar 4.11 Activity Diagram proses validasi pengguna

4.2.1.2 Perancangan Class Diagram

Perancangan *class diagram* digunakan untuk menampilkan sejumlah kelas dan package yang ada pada sistem perangkat lunak yang dikembangkan. *Class diagram* merupakan diagram yang menunjukkan hubungan antar kelas dalam sistem yang sedang dibangun dan membentuk suatu relasi. Perancangan ini akan menampilkan perancangan *class diagram* fitur fitur utama dari aplikasi client dan server.

Pada Gambar 4.12 ditunjukkan *class diagram* pada aplikasi yang merupakan kumpulan dari *Activity* utama pada aplikasi beserta *package* yang digunakan. Dimana terdapat *SplashScreenActivity*, *SignupActivity*, *MainActivityPersonal*, *MainActivityGarage*, dan *ResponseActivity*. *SplashScreenActivity* merupakan *class* yang akan melakukan pengecekan sebelum menjalankan aplikasi, dimana pengecekan yang dimaksud adalah pengecekan status pengguna telah masuk, maupun izin penggunaan fitur dari android yang akan digunakan. *SignupActivity* merupakan *class* yang berfungsi untuk menampilkan antarmuka dan melakukan proses pendaftaran pengguna baru. *MainActivityPersonal* merupakan *class* utama

untuk pengguna personal dimana akan terdapat proses pencarian bantuan. *MainActivityGarage* adalah *class* utama untuk pengguna bengkel yang akan terdapat proses penampilan statistik pencarian bantuan. *ResponseActivity* merupakan *class* yang digunakan untuk menampilkan dan memproses respon pemberian bantuan.

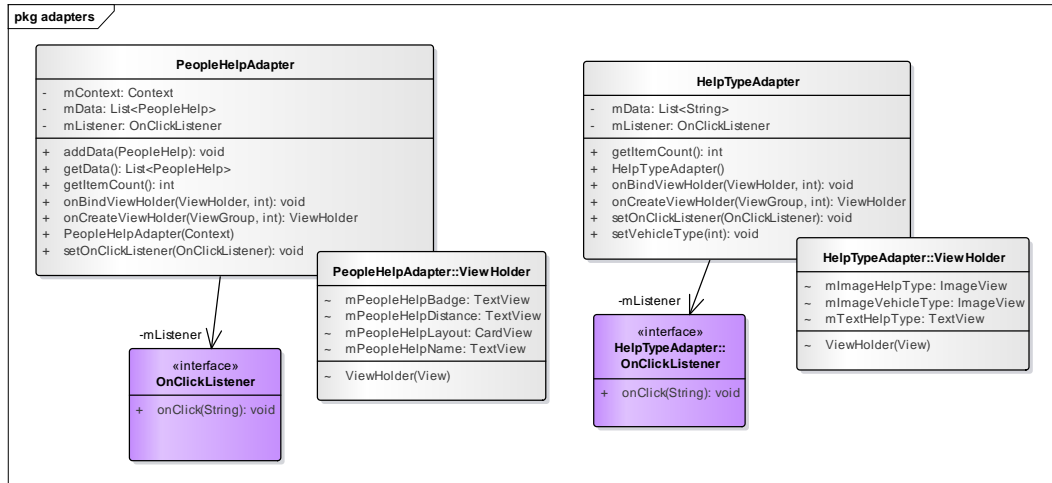
Gambar 4.13 menunjukkan *class diagram* yang terdapat pada *package fragment*. *Package* ini berisi kumpulan *class* yang digunakan untuk sebagai tampilan pada beberapa *activity* yang telah dijelaskan sebelumnya. Terdapat 3 *class* *PersonalRequestHelpFragment*, *PersonalProcessHelpFragment* dan *PersonalResponseHelpFragment* yang merupakan *fragment* untuk digunakan pada *MainActivityPersonal* sebagai tampilan dari proses pencarian bantuan, pemilihan pemberian bantuan, akhiri bantuan. Sedangkan *class* *SignupBasic*, *SignupType* dan *SignupGarage* merupakan *fragment* untuk proses pendaftaran pengguna baru yang digunakan pada *SignupActivity*. Terdapat juga *MapFragment* yang digunakan untuk menampilkan peta yang akan digunakan pada *activity* maupun *fragment* yang membutuhkan penampilan peta dan interaksi dengan peta seperti *PersonalRequestHelpFragment*.

Gambar 4.14 menunjukkan sebagian *class diagram* yang terdapat pada *package adapters*. *Package* ini berisi *class* yang berfungsi mengelola data untuk *RecyclerView* yang digunakan pada *fragment* maupun *activity*.

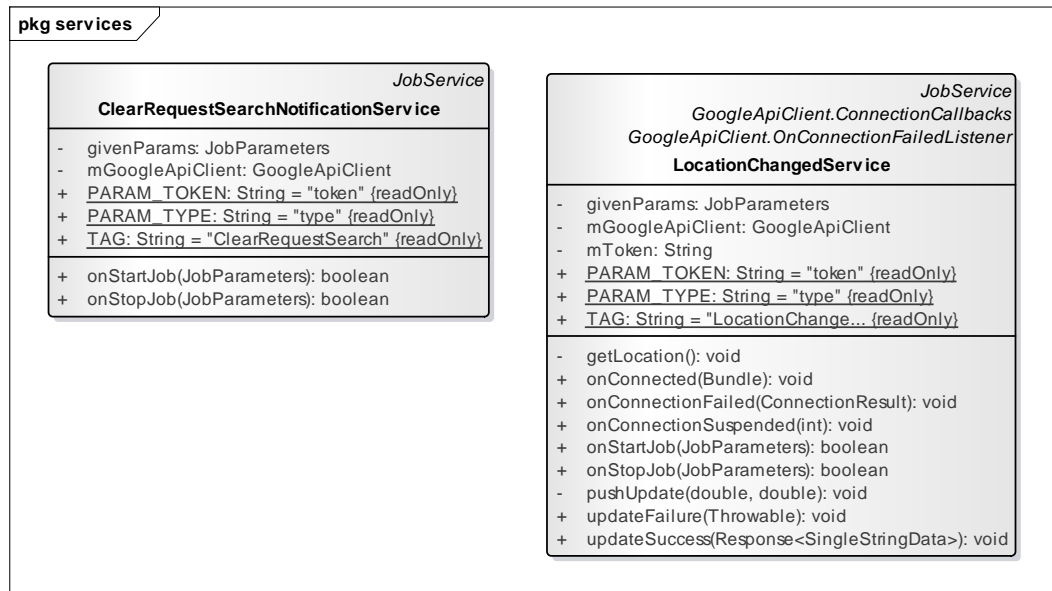
Gambar 4.15 menunjukkan *class diagram* yang terdapat pada *package services* yang berisi *class* untuk melakukan pekerjaan di *background*. Seperti *LocationChangedService* yang akan melakukan perubahan data lokasi terakhir pengguna secara *automatis* dan *ClearRequestSearchNotificationService* yang akan secara *automatis* memberikan respon pemberian bantuan sebagai abaikan maupun tidak dapat mendatangi saat terdapat pemberitahuan pencarian bantuan dan tidak meresponnya dalam waktu tertentu.

Gambar 4.16 menunjukkan *class diagram* yang terdapat pada *package notifications*. *Package* ini berisi *class* untuk menangani notifikasi atau pemberitahuan yang diberikan dari *server* yang ditangani oleh *class* *NotificationHandlerService* dan menangani perubahan *token* dari notifikasi untuk dikirimkan ke *server* yang ditangani oleh *class* *NotificationInstanceIDService*. Untuk menangani pemberitahuan yang tiba *NotificationHandlerService* membutuhkan *class* yang disediakan *package notifications* pada *package utils*.

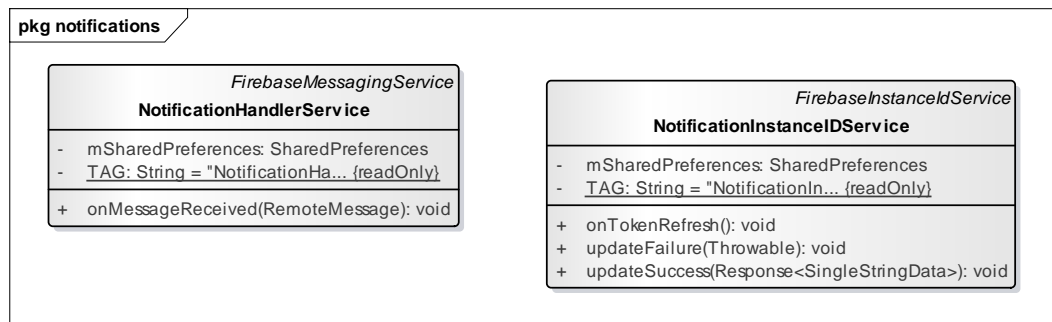
Gambar 4.17 dan Gambar 4.18 merupakan *class diagram* berisi *class* *class* yang berfungsi sebagai *class* pembantu *class* lainnya. Didalam *package* ini terdapat *package* lain seperti *notifications*, *playservices*, *networks* dan *customviews*. Dimana *packages notifications* berisi *class* yang akan membantu suatu *class* menangani notifikasi, *package playservices* berisi *class interface* yang akan membantu menangani respon ketersediaan *Google PlayServices*, *package networks* berisi *class* dan *interface* yang akan menangani hubungan aplikasi dengan *server*, *packages customviews* berisi *class* dan *interface* yang berfungsi menangani interaksi pada elemen elemen *view* pada *activity* atau *fragment*.



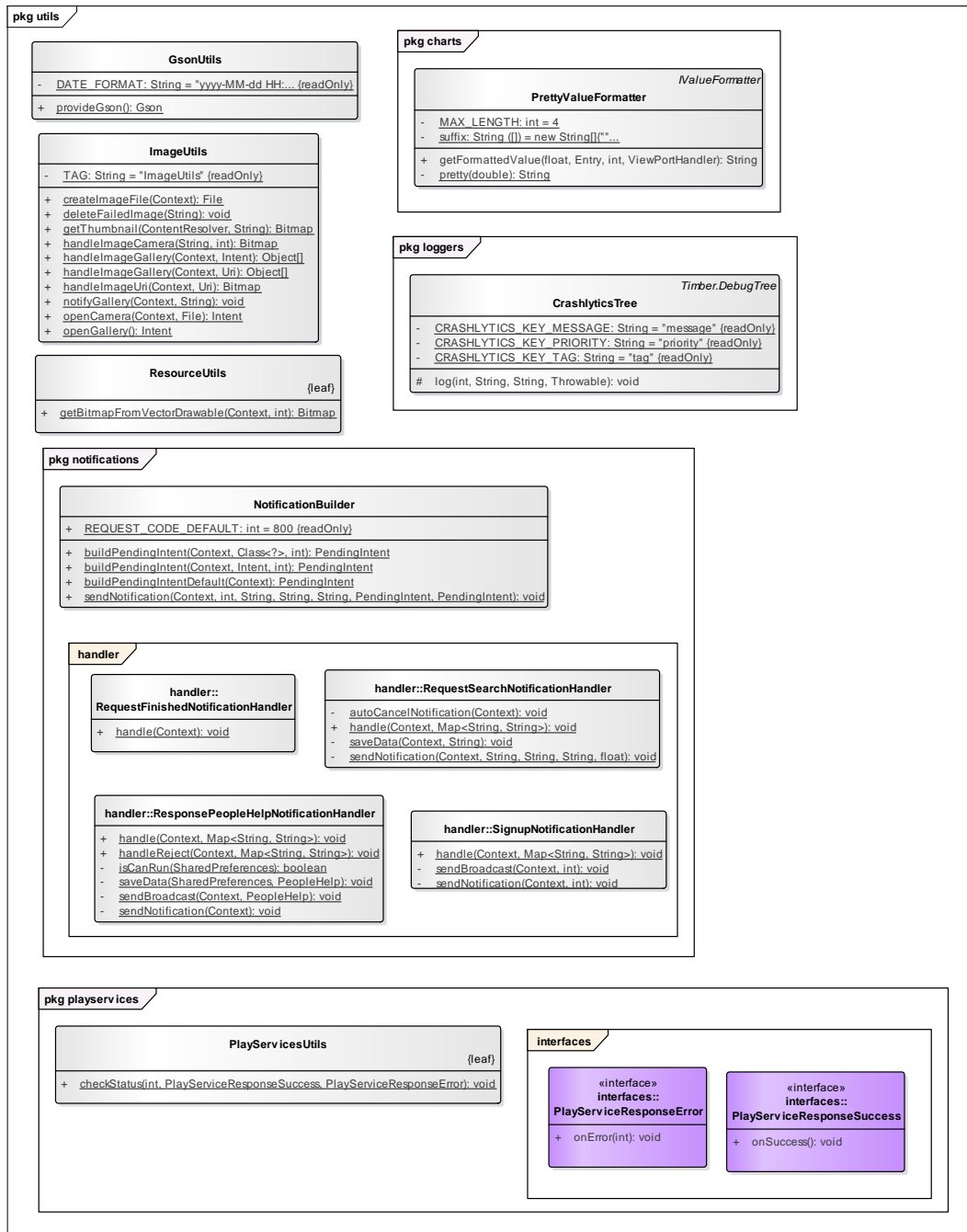
Gambar 4.14 Class Diagram Package Adapters



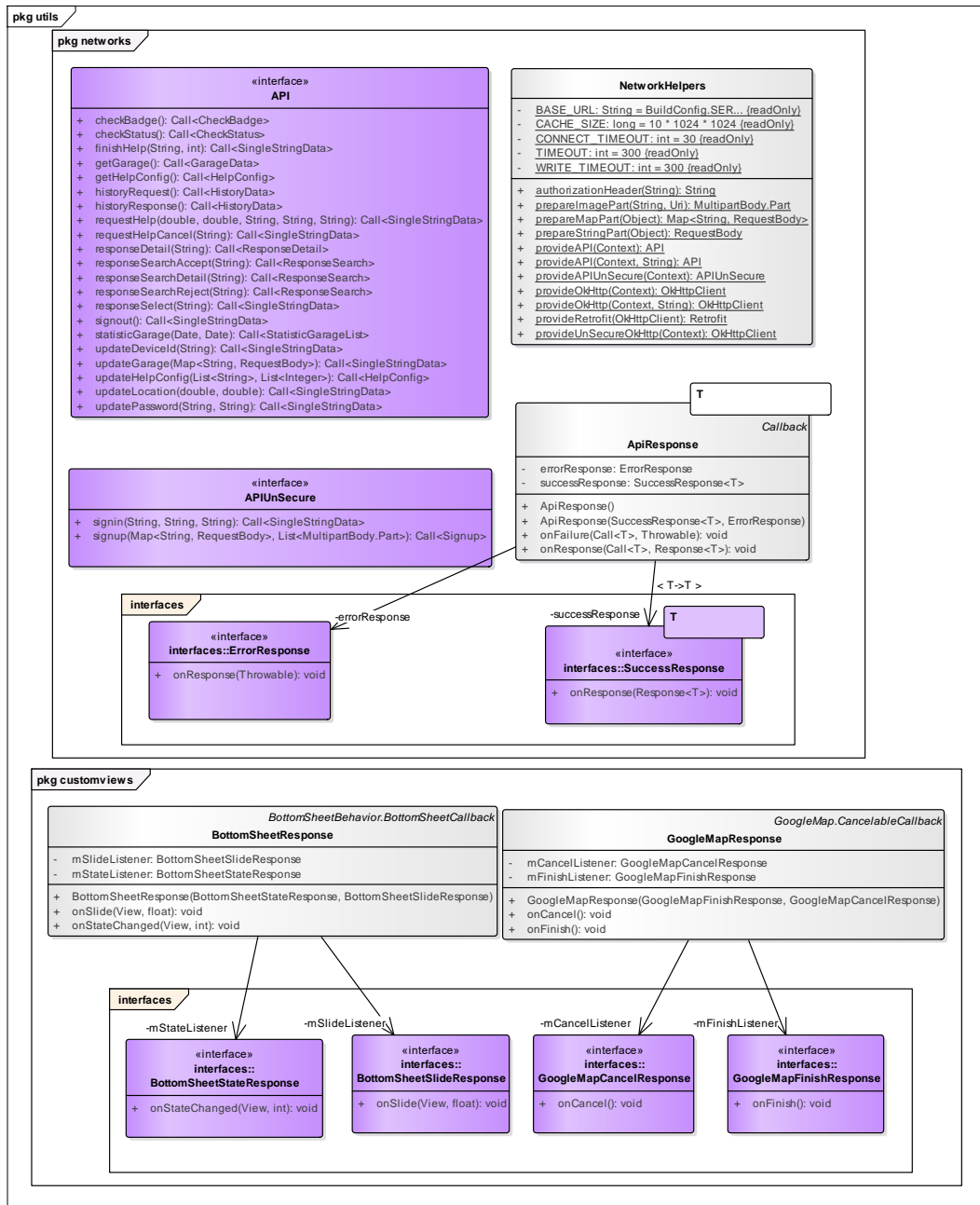
Gambar 4.15 Class Diagram Package Services



Gambar 4.16 Class Diagram Package Notifications



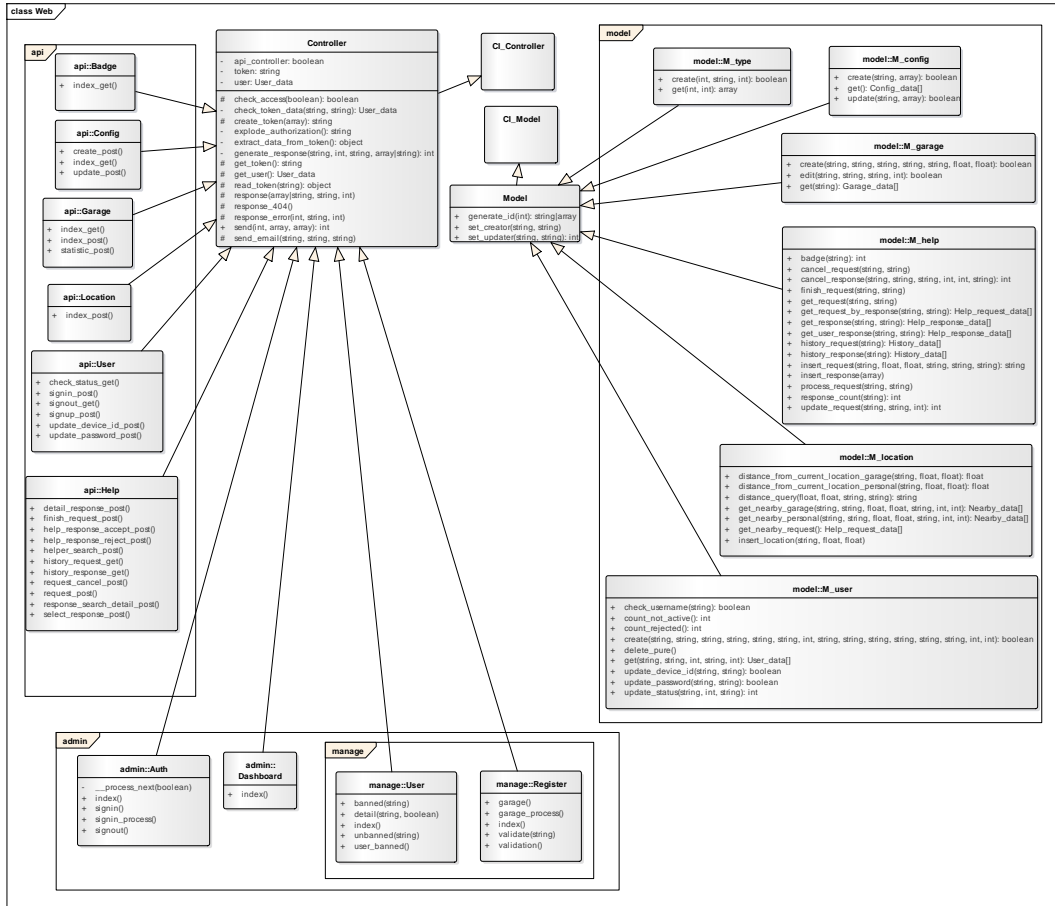
Gambar 4.17 Class Diagram package Utils



Gambar 4.18 Class Diagram Package Utils

Pada Gambar 4.19 menunjukkan class diagram pada sisi server, dimana menggunakan framework PHP berbasis MVC (*Model View Controller*). Controller terdapat 2 direktori utama, yaitu *api* dan *admin*, dimana di dalam direktori admin terdapat direktori lain bernama *manage*. Fungsi dari direktori *api* pada controller adalah sebagai penyedia *web service* dimana di dalamnya terdapat controller dengan class *Badge*, *Config*, *Garage*, *Location*, *User* dan *Help* yang tiap *method* didalamnya diakhiri dengan *_get* atau *_post* sebagai pembeda metode transaksi akses dari *client*. Sedangkan pada controller yang terdapat pada *admin* seperti class *Auth*, *Dashboard* dan *User* tidak terdapat akhirus seperti itu karena nantinya

ini diakses melalui browser sehingga tidak diperlukan pembeda metode transaksi seperti pada *web service*. Untuk *model* terdapat 6 class yaitu *M_type*, *M_config*, *M_Garage*, *M_Help*, *M_Location* dan *M_User* yang digunakan untuk transaksi menuju penyimpanan basis data, baik menyimpan data maupun mengambil data dari basis data.



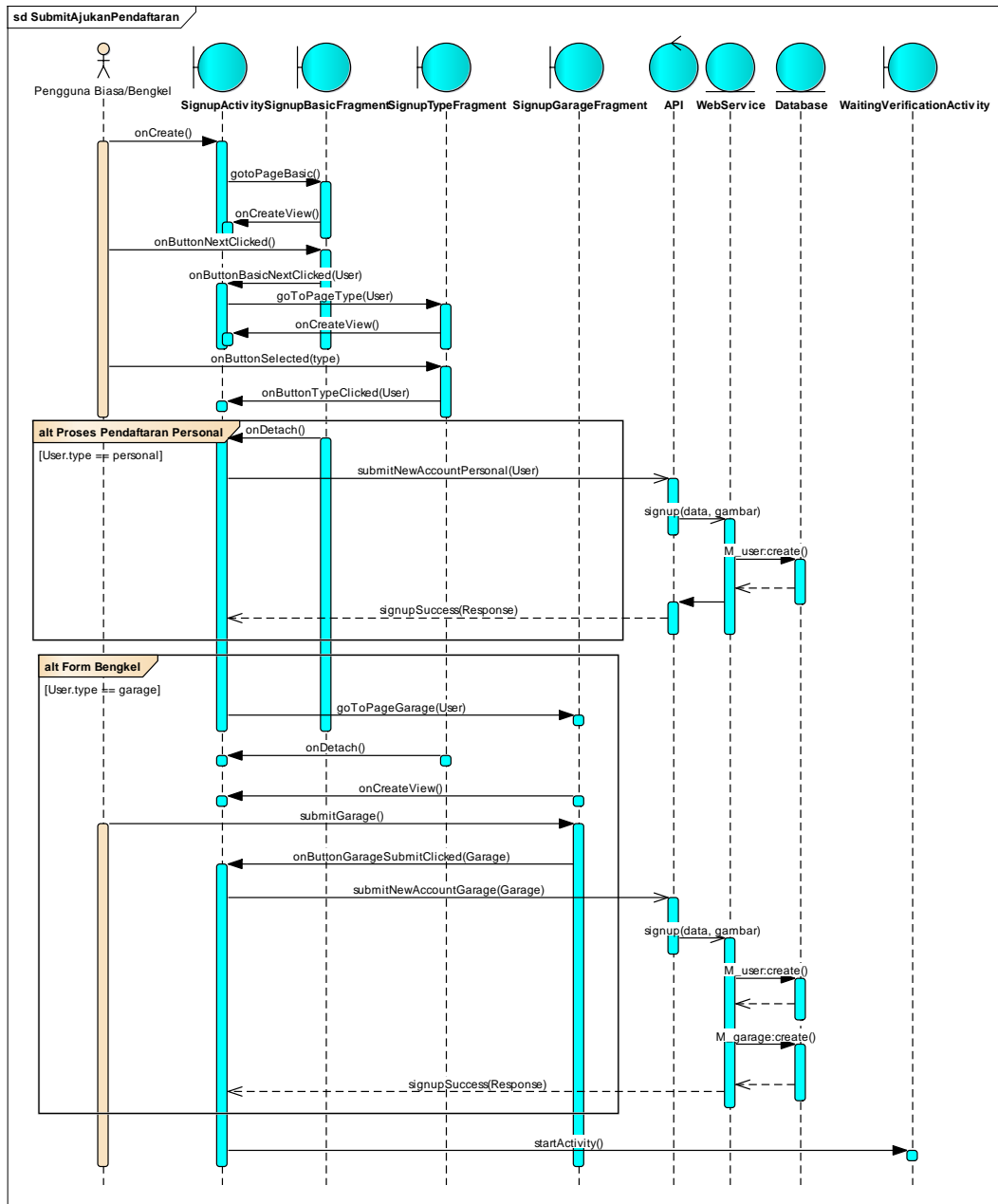
Gambar 4.19 Class Diagram Web

4.2.1.3 Perancangan Sequence Diagram

Pada diagram *sequence* akan memodelkan bagaimana perilaku yang dilakukan antara aktor didalam sistem berdasarkan waktu yang runtut. Berikut merupakan diagram-diagram *sequence* pada sistem pencarian dan pemberian bantuan pada permasalahan kendaraan berdasarkan dari skenario *use case* yang telah dibuat sebelumnya.

1. Sequence Diagram skenario pendaftaran

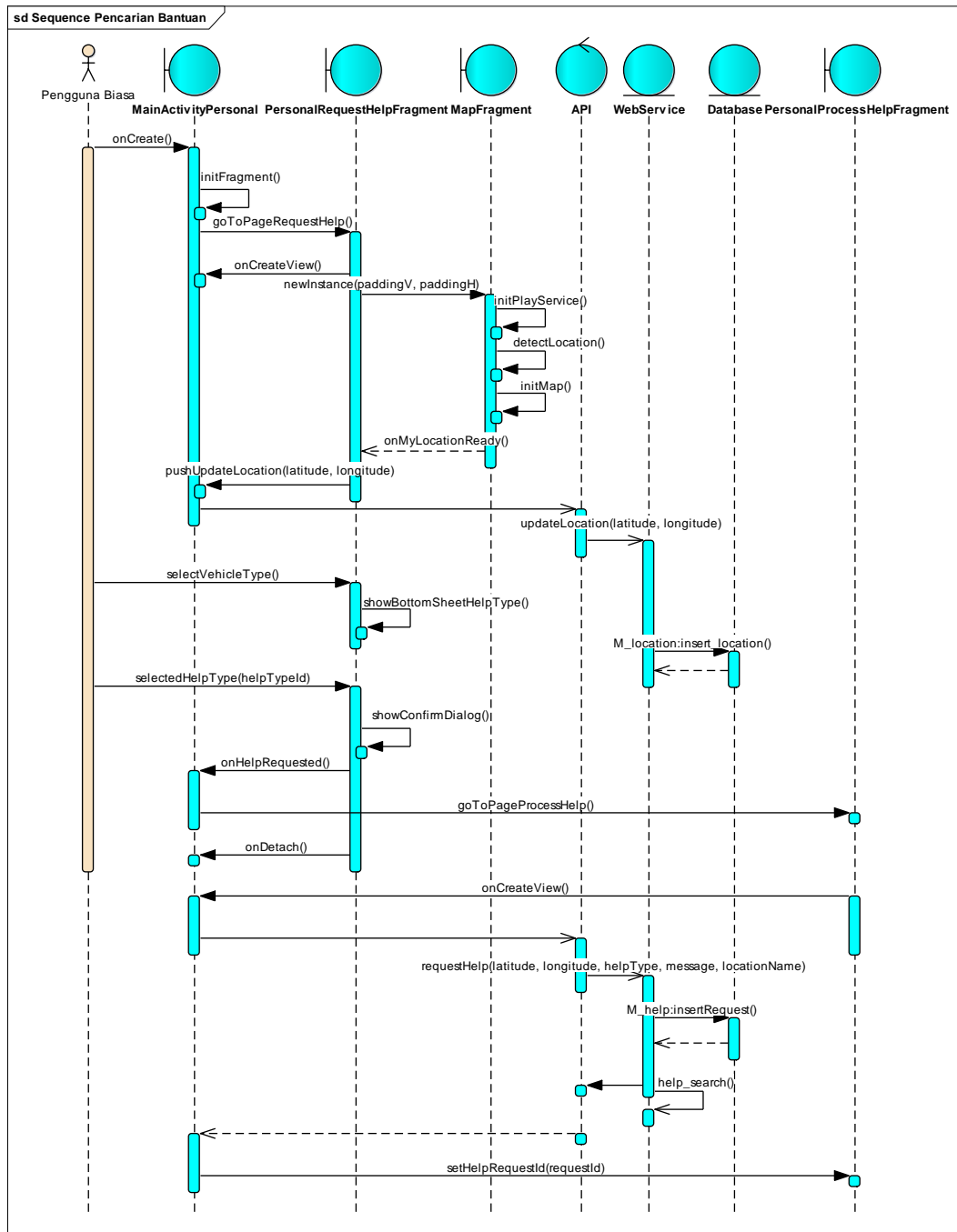
Gambar 4.20 merupakan diagram *sequence* untuk proses pendaftaran. Diagram ini menggambarkan interaksi ketika pengguna mengajukan pendaftaran melalui aplikasi dan selanjutnya aplikasi mengirim data ke *server* untuk disimpan.



Gambar 4.20 Sequence Diagram skenario pendaftaran

2. Sequence Diagram skenario pencarian bantuan

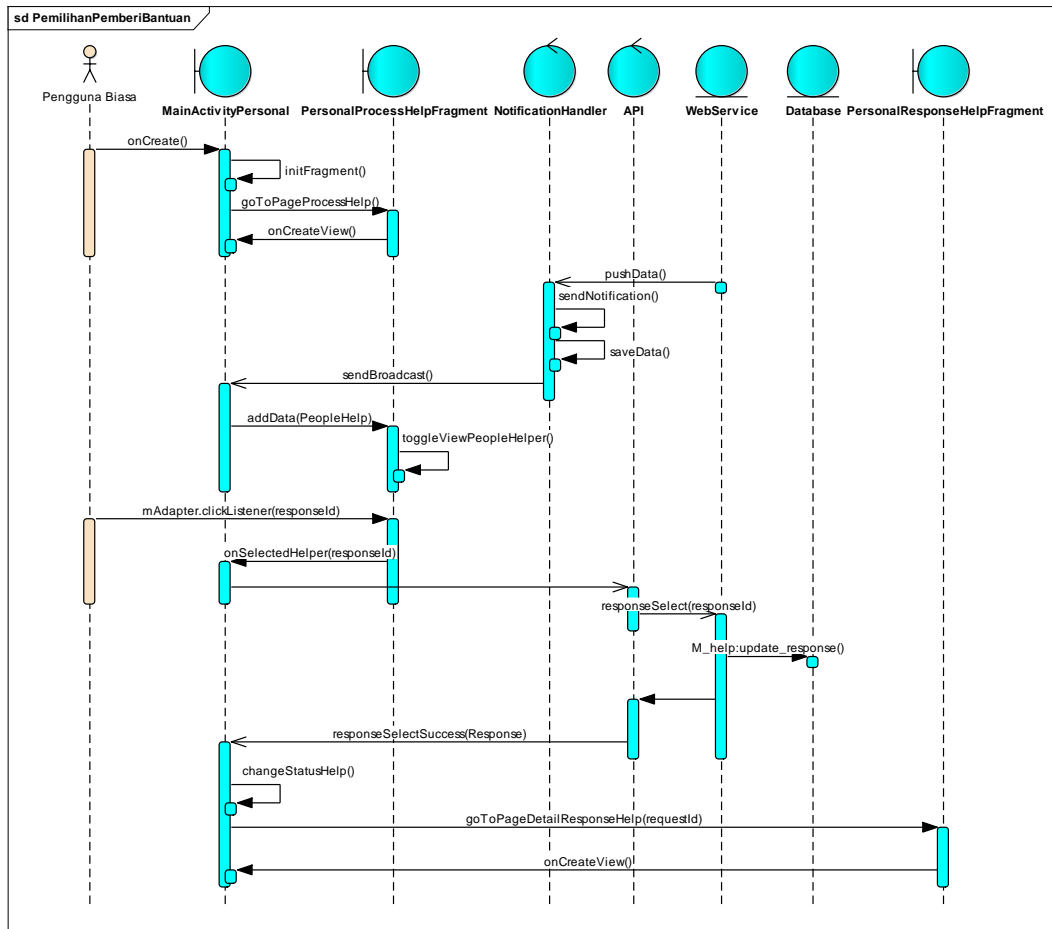
Gambar 4.21 merupakan diagram *sequence* untuk proses pencarian bantuan. Diagram ini menggambarkan interaksi ketika pengguna mencari bantuan melalui aplikasi dan selanjutnya aplikasi mengirim data ke *server* untuk disimpan.



Gambar 4.21 Sequence Diagram skenario pencarian bantuan

3. Sequence Diagram skenario pemilihan pemberi bantuan

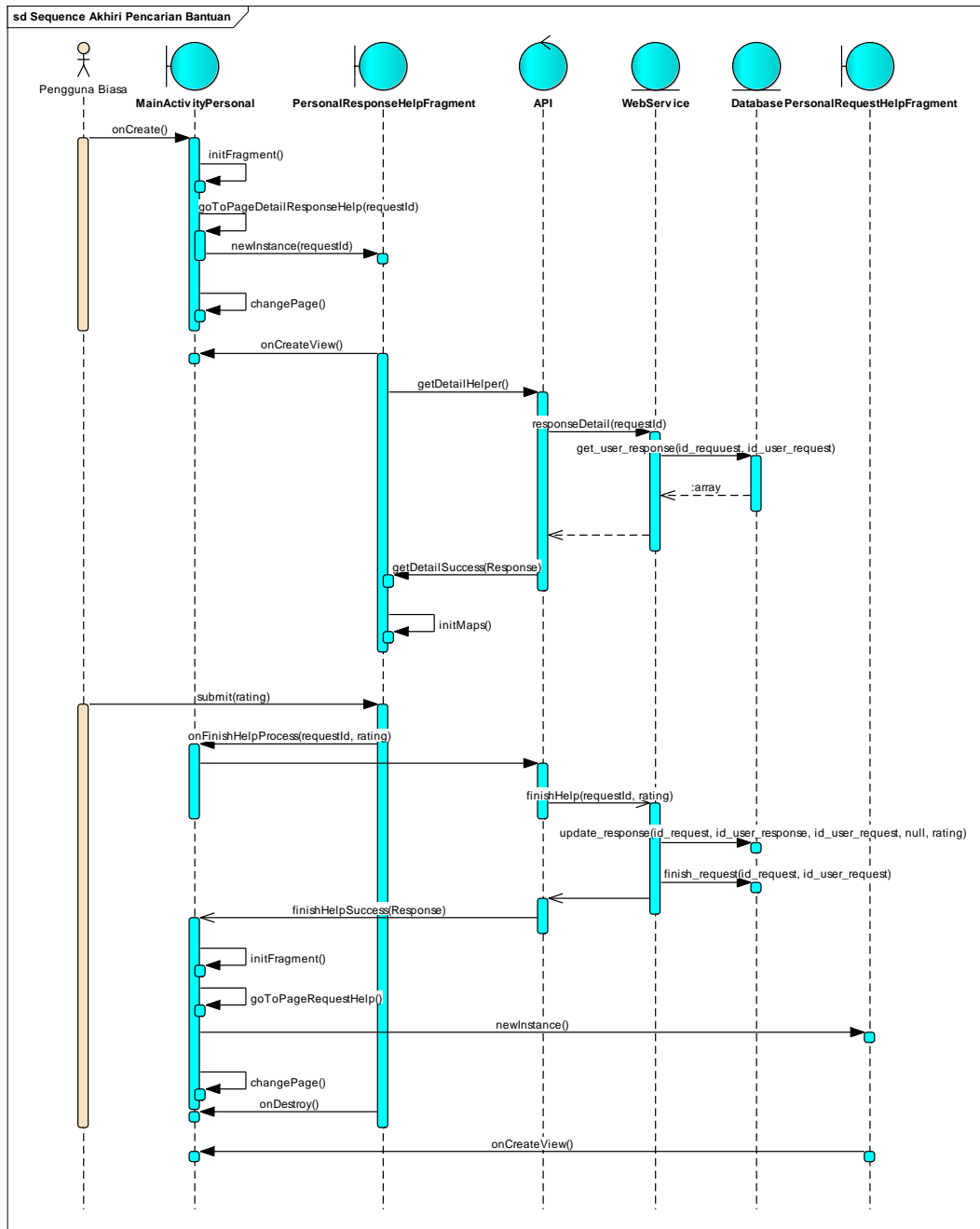
Gambar 4.22 merupakan diagram *sequence* untuk proses pemilihan pemberi bantuan. Diagram ini menggambarkan interaksi ketika pengguna memilih pemberi bantuan melalui aplikasi dan selanjutnya aplikasi mengirim data ke *server* untuk disimpan.



Gambar 4.22 Sequence Diagram skenario pemilihan pemberi bantuan

4. Sequence Diagram skenario akhiri pencarian bantuan

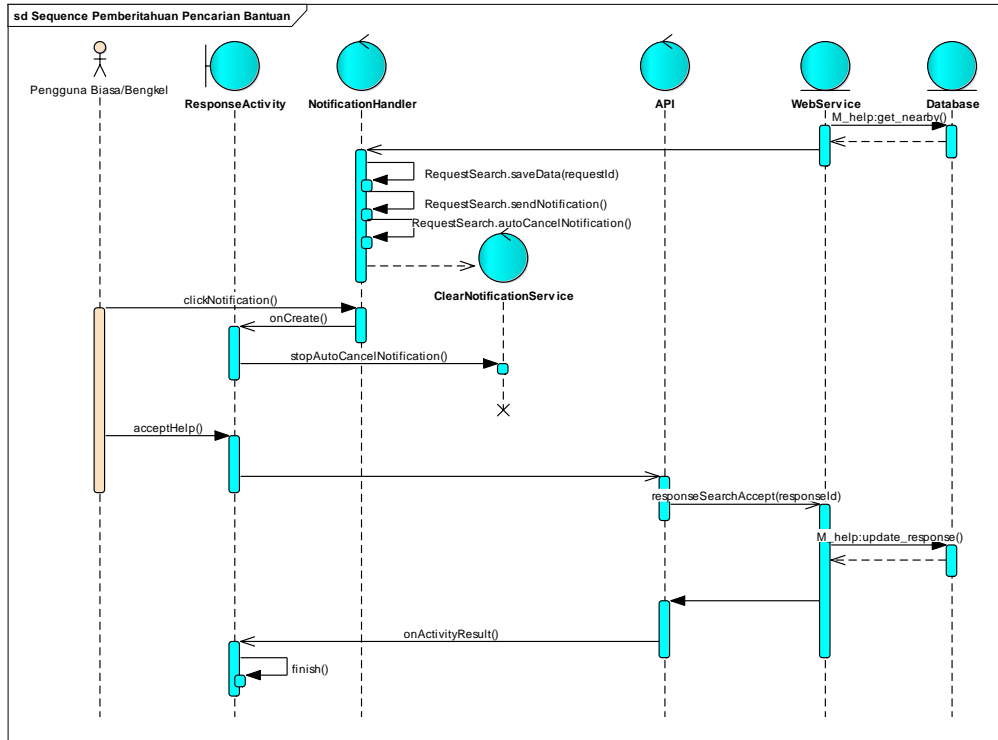
Gambar 4.23 merupakan diagram *sequence* untuk proses mengakhiri proses pencarian bantuan. Diagram ini menggambarkan interaksi ketika pengguna mengakhiri proses pencarian bantuan melalui aplikasi dan selanjutnya aplikasi mengirim data ke *server* untuk disimpan.



Gambar 4.23 Sequence Diagram skenario akhiri pencarian bantuan

5. Sequence Diagram skenario pemberitahuan pencarian bantuan

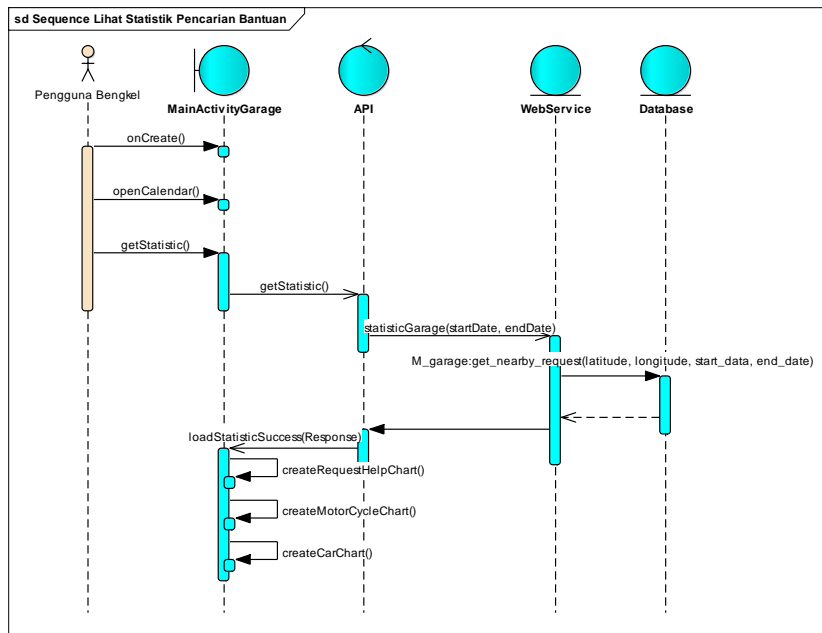
Gambar 4.24 merupakan diagram *sequence* untuk proses pemberitahuan saat sedang terjadi pencarian bantuan disekitarnya. Diagram ini menggambarkan interaksi ketika pengguna mendapatkan pemberitahuan pencarian bantuan melalui aplikasi dan selanjutnya aplikasi mengirim data ke *server* untuk disimpan.



Gambar 4.24 Sequence Diagram skenario pemberitahuan pencarian bantuan

6. Sequence Diagram skenario lihat statistik pencarian bantuan

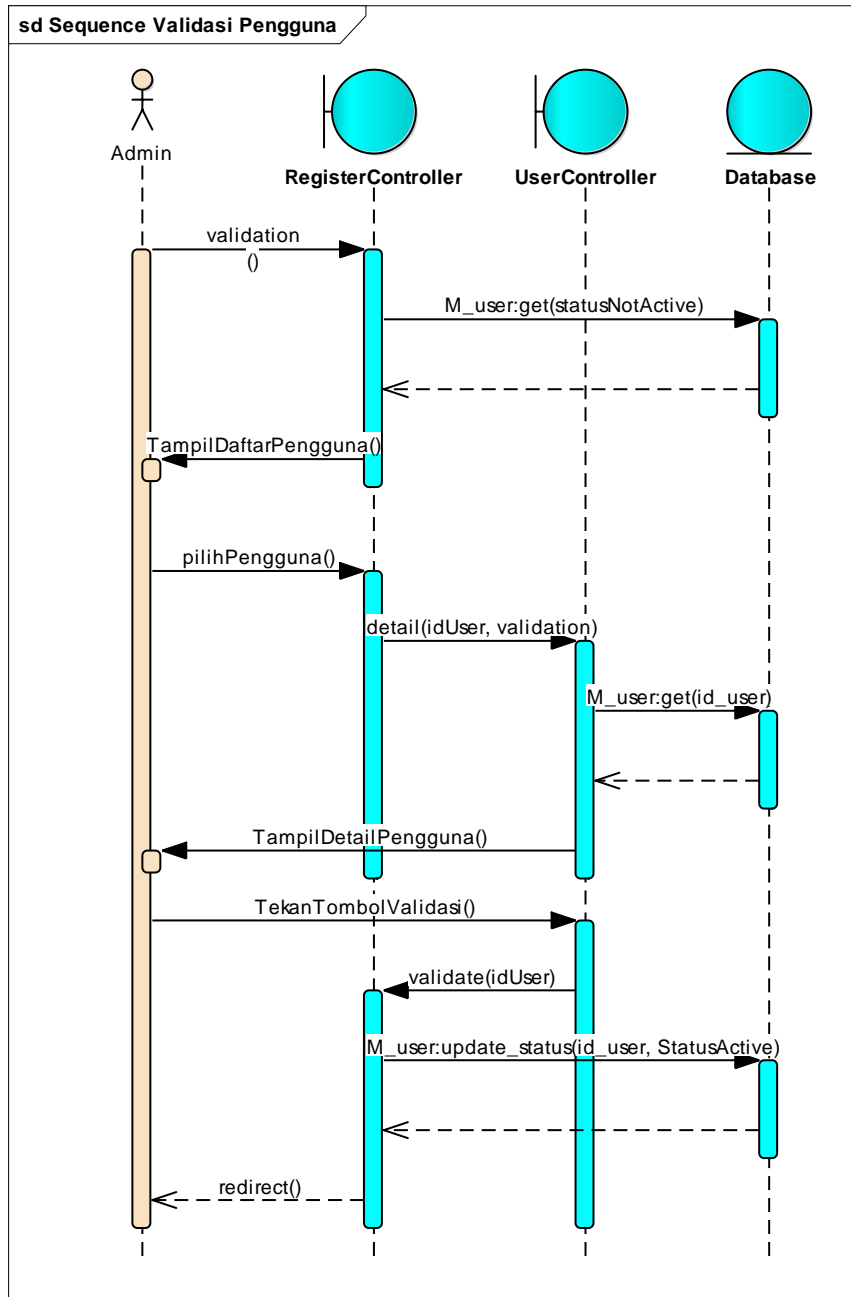
Gambar 4.25 merupakan diagram *sequence* untuk proses melihat statistik pencarian bantuan. Diagram ini menggambarkan interaksi ketika pengguna melihat data statistik pencarian bantuan melalui aplikasi.



Gambar 4.25 Sequence Diagram skenario lihat statistik pencarian bantuan

7. Sequence Diagram skenario validasi pengguna

Gambar 4.26 merupakan diagram *sequence* untuk proses melakukan validasi pendaftaran pengguna. Diagram ini menggambarkan interaksi ketika admin melakukan proses validasi pendaftaran pengguna pada web admin.



Gambar 4.26 Sequence Diagram skenario validasi pengguna

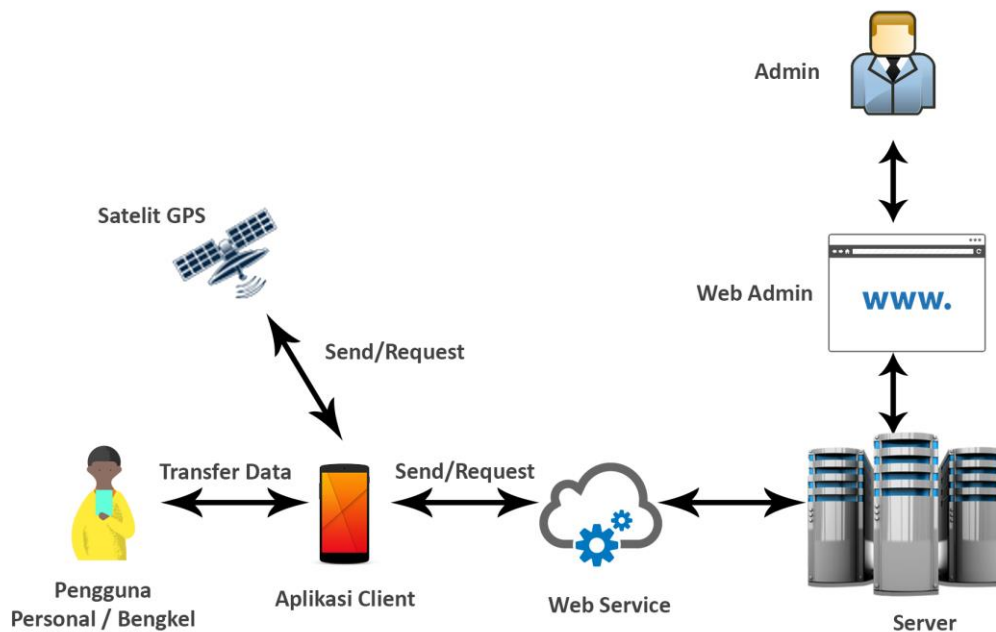
4.2.2 Perancangan arsitektur sistem

Sistem pencarian dan pemberian bantuan terhadap permasalahan pada kendaraan dibangun dengan menggunakan konsep *client* dan *server*. Perancangan

arsitektur sistem dibuat untuk menjelaskan bagaimana arsitektur sistem akan diimplementasikan. Sistem ini diberi nama “Todong Ban”. Dalam perancangan arsitektur sistem ini menggunakan pendekatan *Location Based Service* (LBS) seperti yang digambarkan pada Gambar 4.27.

Sistem *client* dibuat dengan bahasa pemrograman Java dan XML yang merupakan bagian dari pembuatan aplikasi *native* dari android dengan memanfaatkan beberapa sensor dari *smartphone* android tersebut. Sensor yang akan digunakan antara lain kamera, GPS, dan internet. Sedangkan untuk sistem *server* dibangun dengan menggunakan bahasa pemrograman PHP, HTML, CSS, Javascript dengan menggunakan *Framework Codeigniter*.

Pengguna sistem baik pengguna personal dan pengguna bengkel hanya dapat mengakses aplikasi *client*, sedangkan sistem *server* hanya dapat diakses oleh administrator. *Web Service* digunakan untuk menjembatani komunikasi antara *client* dan *server*, dimana proses pertukaran data yang terjadi antara *client* dan *server* dikirim menggunakan format JSON.



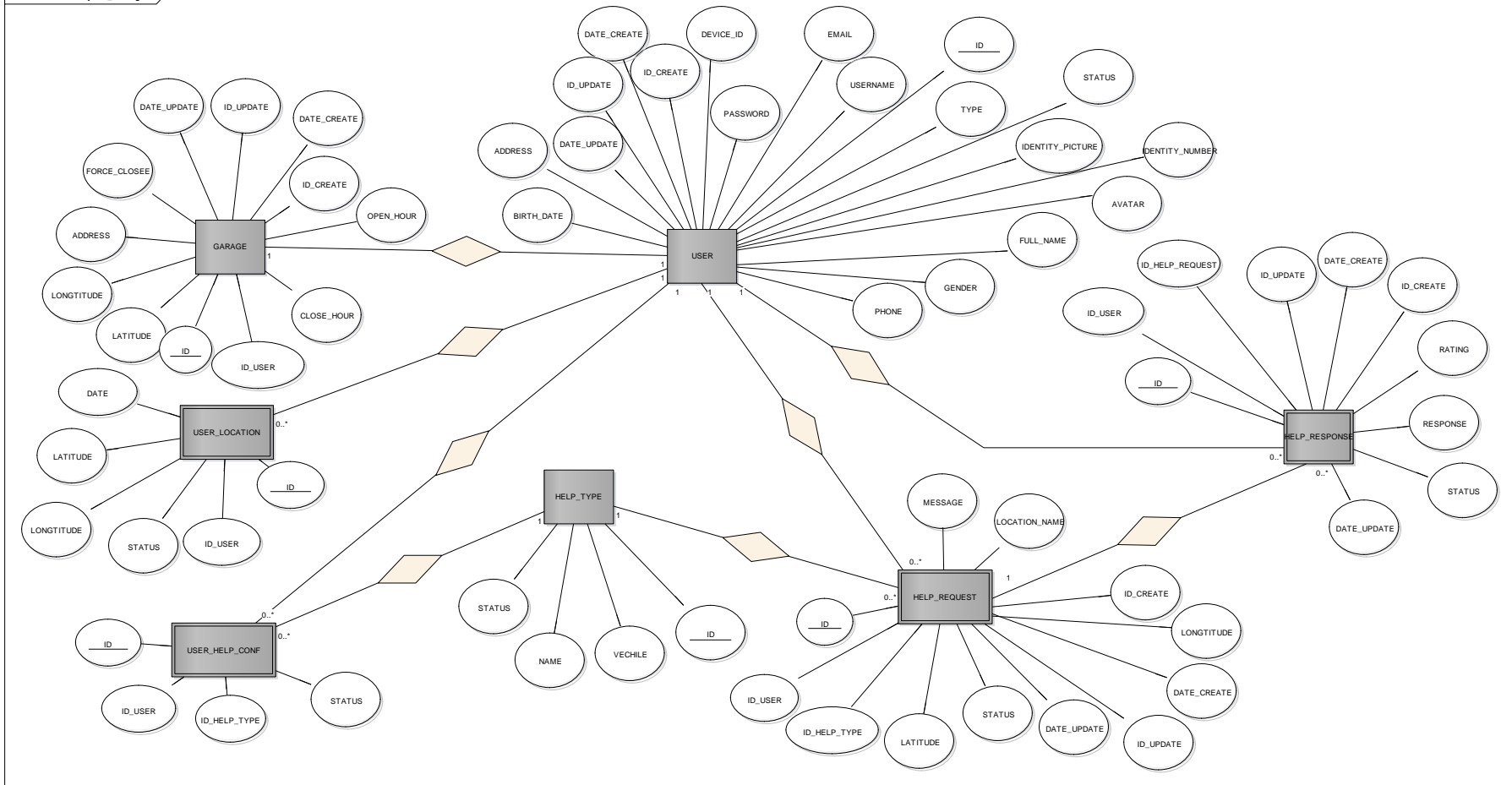
Gambar 4.27 Arsitektur sistem pencarian dan pemberian bantuan berdasarkan lokasi terdekat

4.2.3 Perancangan basis data

Perancangan basis data diperlukan untuk menggambarkan bagaimana data yang diperlukan sistem akan disimpan. Pada penelitian ini perancangan basis data direpresentasikan dalam bentuk ERD (*Entity Relationship Diagram*). ERD menunjukkan hubungan yang terjadi di antara objek (*entitas*) yang terlibat dalam suatu *database*. ERD berisi komponen-komponen himpunan entitas dan himpunan relasi yang masing-masing dilengkapi dengan beberapa atribut yang mempresentasikan seluruh fakta yang ditinjau dari keadaan yang nyata. Pada

perancangan basis data sistem ini terdapat 7 (tujuh) buah tabel yaitu Tabel user, Tabel garage, Tabel user_location, Tabel help_type, Tabel user_help_conf, Tabel help_request dan Tabel help_response. Tabel user digunakan untuk menyimpan data pengguna dan administrator. Tabel garage digunakan untuk menyimpan data detail mengenai bengkel milik pengguna. Tabel user_location merupakan table yang digunakan untuk mencatat lokasi terakhir dari pengguna personal. Tabel help_type digunakan untuk menyimpan data jenis bantuan yang akan digunakan dalam table help_request dan user_help_conf. Tabel user_help_conf sendiri digunakan untuk menyimpan data jenis bantuan apa yang bisa ditangani oleh pengguna. Tabel help_request digunakan untuk menyimpan data pencarian bantuan yang dilakukan oleh pengguna, dan table help_response digunakan untuk mencata pengguna lain yang meresponse dari permintaan bantuan yang terdapat pada table help_request.

ERD sistem pencarian dan pemberian bantuan pada permasalahan kendaraan dapat dilihat pada Gambar 4.28. Detail atribut tiap basis data dijelaskan pada Tabel 4.10.



Gambar 4.28 ERD Sistem

Tabel 4.10 Data atribut tiap basis data

Nama Tabel	Atribut	Tipe, Length
USER	ID (PK)	Varchar, 36
	USERNAME	Char, 16
	EMAIL	Varchar, 256
	PASSWORD	Varchar, 128
	FULL_NAME	Varchar, 256
	PHONE	Varchar, 50
	GENDER	Tinyint
	AVATAR	Varchar, 40
	ADDRESS	Text
	IDENTITY_NUMBER	Varchar, 20
	IDENTITY_PICTURE	Varchar, 40
	DEVICE_ID	Varchar, 256
	TYPE	Tinyint
	STATUS	Tinyint
	ID_CREATE	Varchar, 36
	DATE_CREATE	Datetime
	ID_UPDATE	Varchar, 36
	DATE_UPDATE	Datetime
USER_LOCATION	ID (PK)	Varchar, 36
	ID_USER (FK)	Varchar, 36
	LATITUDE	Decimal, 10,7
	LONGTITUDE	Decimal, 10,7
	DATE	Datetime
	STATUS	Tinyint
HELP_TYPE	ID (PK)	Varchar, 36
	VECHILE	Tinyint
	NAME	Varchar, 20
	STATUS	Tinyint

Tabel 4.10 Data atribut tiap basis data (lanjutan)

Nama Tabel	Atribut	Tipe, Length
GARAGE	ID (PK)	Varchar, 36
	ID_USER (FK)	Varchar, 36
	NAME	Varchar, 125
	OPEN_HOUR	Time
	CLOSE_HOUR	Time
	ADDRESS	Text
	LATITUDE	Decimal, 10,7
	LONGTITUDE	Decimal, 10,7
	FORCE_CLOSE	Tinyint
	STATUS	Tinyint
	ID_CREATE	Varchar, 36
	DATE_CREATE	Datetime
	ID_UPDATE	Varchar, 36
	DATE_UPDATE	Datetime
HELP_RESPONSE	ID (PK)	Varchar, 36
	ID_USER (FK)	Varchar, 36
	ID_HELP_REQUEST (FK)	Varchar, 36
	RESPONSE	Tinyint
	RATING	Tinyint
	STATUS	Tinyint
	ID_CREATE	Varchar, 36
	DATE_CREATE	Datetime
	ID_UPDATE	Varchar, 36
	DATE_UPDATE	Datetime
USER_HELP_CONF	ID (PK)	Varchar, 36
	ID_USER (FK)	Varchar, 36
	ID_HELP_TYPE (FK)	Varchar, 26
	STATUS	Tinyint

Tabel 4.10 Data atribut tiap basis data (lanjutan)

Nama Tabel	Atribut	Tipe, Length
HELP_REQUEST	ID (PK)	Varchar, 36
	ID_USER (FK)	Varchar, 36
	ID_HELP_TYPE (FK)	Varchar, 36
	LATITUDE	Decimal, 10,7
	LONGTITUDE	Decimal, 10,7
	STATUS	Tinyint
	ID_CREATE	Varchar, 36
	DATE_CREATE	Datetime
	ID_UPDATE	Varchar, 36
	DATE_UPDATE	Datetime

4.2.4 Perancangan navigasi dan antarmuka

Perancangan navigasi dan antarmuka menjelaskan bagaimana rancangan antarmuka dan alur perpindahan dari setiap halaman aplikasi. Perancangan ini diperlukan untuk mempermudah pembuatan aplikasi yang dapat digunakan dengan mudah oleh pengguna dalam pengoperasian aplikasi.

4.2.4.1 Perancangan Screen Flow

Perancangan *screen flow* ini digunakan untuk mempermudah menjelaskan alur proses antarmuka dari aplikasi pada sisi *client*. Antarmuka ini pada awal akan membuka halaman *splash screen* dimana pada halaman ini akan dilakukan beberapa pengecekan sebelum menjalankan fitur-fitur utama aplikasi, seperti pengecekan apakah pengguna telah *login*, status token pengguna, dan perizinan penggunaan fitur dari *smartphone* seperti GPS dan sebagainya. Jika pengguna belum *login* maka akan ditampilkan halaman pendaftaran yang terdapat menu untuk berpindah ke halaman *login/masuk*.

Pada tampilan pendaftaran pertama kali pengguna akan disuguhkan formulir data diri pengguna beserta tombol selanjutnya, yang apabila ditekan akan memberikan pilihan tombol daftar sebagai pengguna personal atau daftar sebagai bengkel, apabila daftar sebagai bengkel dipilih maka akan di tampilkan formulir data bengkel dan tombol daftar yang berfungsi untuk menjalankan fitur pendaftaran seperti halnya saat memilih tombol daftar sebagai pengguna personal sebelumnya dan akan tampil halaman menunggu verifikasi. Pada halaman *login* akan di tampilkan formulir untuk masuk dan tombol masuk beserta menu daftar, apabila menekan tombol masuk dan berhasil *login* maka akan diarahkan menuju *splash screen* yang kemudian diarahkan menuju halaman utama sesuai jenis pengguna.

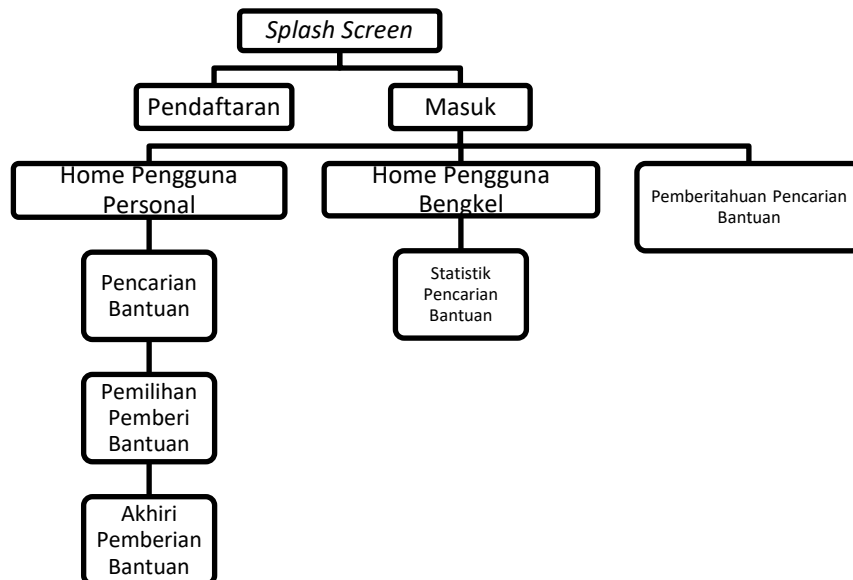
Pada tampilan halaman utama/home pengguna personal terdapat navigasi menu yang dapat dibuka di sebelah kiri, peta pada keseluruhan layar dan tombol pilihan jenis kendaraan, dimana saat peta digerakan maka akan memunculkan kursor untuk merubah lokasi permintaan bantuan, dan jika tombol jenis kendaraan ditekan maka akan ditampilkan jenis permasalahan yang dapat diminta sesuai kendaraan yang dipilih dan jika dipilih akan tampil halaman pemilihan pemberi bantuan.

Pada halaman pemilihan pemberi bantuan terdapat daftar pengguna yang bersedia membantu dan daftar bengkel di sekitar, dimana jika salah satu bagian dari daftar tersebut dipilih maka akan ditampilkan halaman untuk akhiri bantuan.

Halaman akhiri bantuan menampilkan detail mengenai pemberi bantuan beserta lokasi dari pengguna pemberi bantuan dalam bentuk peta, selain itu terdapat tombol untuk akhiri bantuan yang jika ditekan maka akan menampilkan tampilan penilaian dengan berupa pemberian *rating* beserta tombol beri nilai yang akan mengakhiri bantuan dan akan mengembalikan tampilan menjadi tampilan awal.

Untuk pengguna bengkel halaman utama/home adalah halaman statistik pencarian bantuan, dimana pada awalnya selain navigasi menu terdapat formulir pemilihan tanggal mulai dan akhir untuk data statistik dan tombol untuk mengolah statistik tersebut, saat tombol ditekan maka aplikasi akan menampilkan data dalam berupa grafik di bawah tanggal yang dipilih tadi.

Untuk pengguna personal dan pengguna bengkel terdapat tampilan saat terdapat pemberitahuan pencarian bantuan, dimana halaman tampilan tersebut menampilkan data pencari bantuan beserta tombol untuk merespon pencarian bantuan tersebut. Perancangan *screen flow* dapat dilihat pada Gambar 4.29.

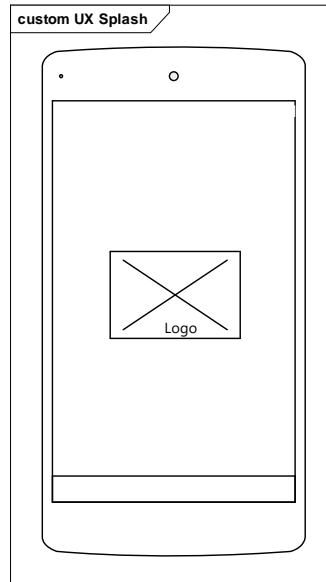


Gambar 4.29 Screen Flow sistem client

4.2.4.2 Perancangan antarmuka

1. Halaman *Splash Screen*

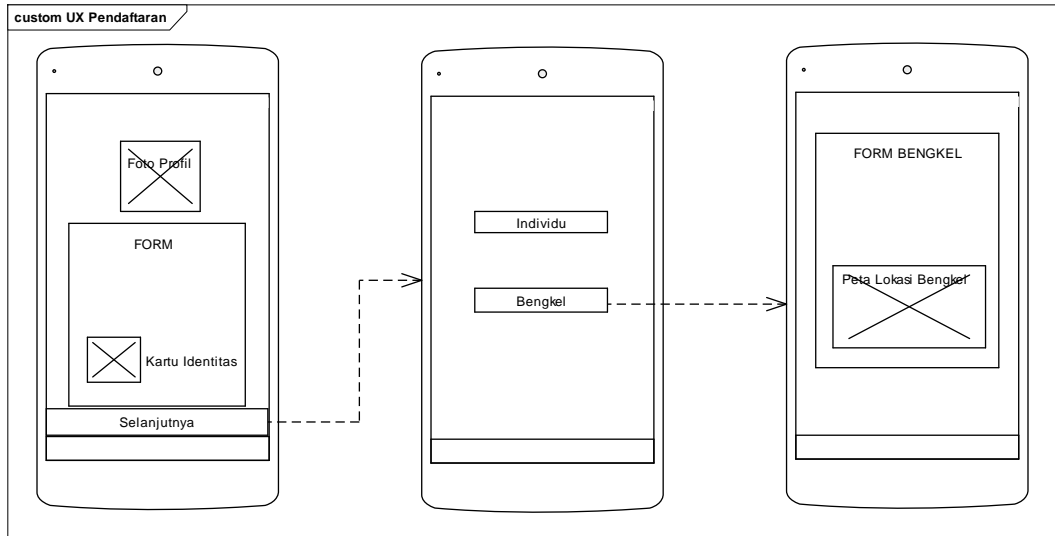
Halaman *splash screen* merupakan halaman yang tampil pertama kali saat membuka aplikasi. Pada halaman ini menampilkan logo dari aplikasi. Rancangan antarmuka *splash screen* dapat dilihat pada Gambar 4.30.



Gambar 4.30 Rancangan antarmuka Splash Screen

2. Halaman pendaftaran

Halaman pendaftaran merupakan halaman yang berfungsi menampilkan formulir pengajuan pendaftaran. Pada halaman ini menampilkan formulir pendaftaran diri berupa foto diri (profil), nama lengkap, tanggal lahir, nomor telpon, alamat, nomor identitas, foto kartu identitas, alamat tinggal dan jenis kelamin. Lalu terdapat pilihan jenis pengguna (biasa/individu dan bengkel). Terdapat juga formulir untuk data bengkel dengan data nama bengkel, alamat bengkel dan lokasi bengkel. Rancangan antarmuka pendaftaran dapat dilihat pada Gambar 4.31.

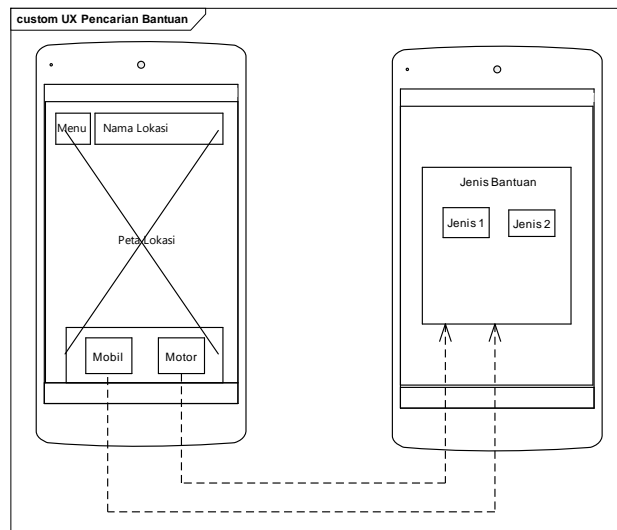


Gambar 4.31 Rancangan antarmuka pendaftaran

Pada Gambar 4.31 terdapat alur dimana saat pengguna menekan tombol selanjutnya pada halaman awal maka akan tampil sebuah halaman baru dengan dua buah tombol yakni tombol tipe pengguna. Saat tombol tipe pengguna bengkel dipilih maka akan tampil sebuah halaman baru yaitu formulir bengkel.

3. Halaman pencarian bantuan

Halaman pencarian bantuan merupakan halaman yang berfungsi untuk melakukan pencarian bantuan. Pada halaman ini menampilkan peta lokasi keberadaan dan tombol jenis kendaraan yang akan menampilkan jenis bantuan setelah ditekan. Rancangan antarmuka pencarian bantuan dapat dilihat pada Gambar 4.32.

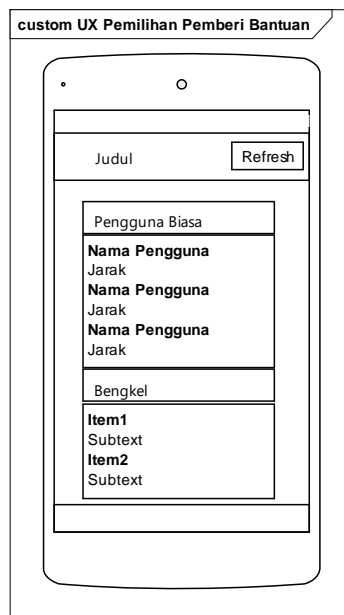


Gambar 4.32 Rancangan antarmuka pencarian bantuan

Pada Gambar 4.32 terdapat alur dimana setelah pengguna menekan tombol jenis kendaraan maka akan muncul sebuah dialog atau tampilan baru diatas tampilan awal untuk memilih jenis bantuan yang diinginkan pengguna.

4. Halaman pemilihan pemberi bantuan

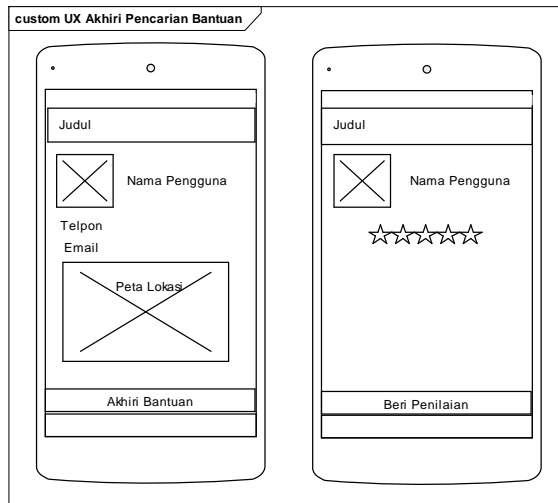
Halaman pemmilihan pemberi bantuan merupakan halaman yang berfungsi untuk memilih pemberi bantuan setelah dilakukan pencarian bantuan. Pada halaman ini menampilkan daftar pengguna personal dan bengkel yang ada disekitar untuk membantu. Rancangan antarmuka pemilihan pemberi bantuan dapat dilihat pada Gambar 4.33.



Gambar 4.33 Rancangan antarmuka pemilihan pemberi bantuan

5. Halaman akhiri pencarian bantuan

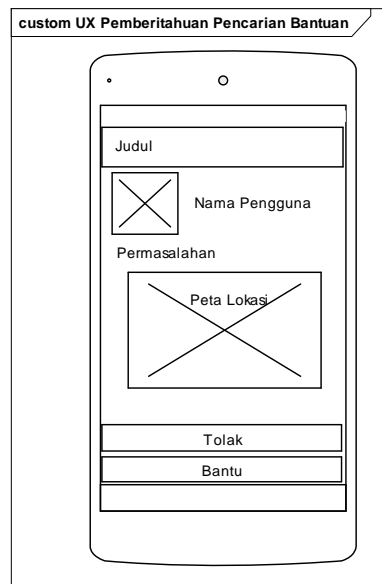
Halaman akhiri pencarian bantuan merupakan halaman yang berfungsi untuk mengakhiri pencarian bantuan dan memberikan nilai terhadap pemberi bantuan. Pada halaman ini menampilkan data pemberi bantuan dan tombol akhiri yang akan menampilkan formulir peniliaian setelah ditekan. Rancangan antarmuka akhiri pencarian bantuan dapat dilihat pada Gambar 4.34.



Gambar 4.34 Rancangan antarmuka akhiri pencarian bantuan

6. Halaman detail pemberitahuan pencarian bantuan

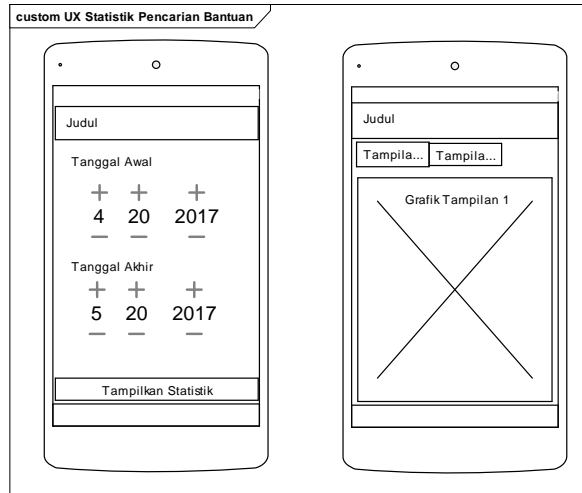
Halaman detail pemberitahuan pencarian bantuan merupakan halaman yang tampil setelah membuka pemberitahuan (notifikasi) yang berfungsi untuk memberikan respon atas permintaan bantuan. Pada data pencari bantuan dan permasalahannya dengan tombol respon bantu atau tolak. Rancangan antarmuka detail pemberitahuan pencarian bantuan dapat dilihat pada Gambar 4.35.



Gambar 4.35 Rancangan antarmuka detail pemberitahuan pencarian bantuan

7. Halaman lihat statistik pencarian bantuan

Halaman lihat statistik pencarian bantuan merupakan halaman yang berfungsi untuk menampilkan data statistik dari pencarian bantuan yang terjadi disekitar bengkel. Rancangan antarmuka lihat statistik pencarian bantuan dapat dilihat pada Gambar 4.36.



Gambar 4.36 Rancangan antarmuka lihat statistik pencarian bantuan

4.2.5 Perancangan komunikasi data

Perancangan komunikasi data antara aplikasi *client* dan *server* menggunakan *web service*. Proses pengiriman data menggunakan format data string dan *file* gambar yang hendak diterima oleh aplikasi *server* untuk disimpan. Proses pengiriman data dari *server* ke *client* menggunakan format data JSON, yang nantinya hendak di ubah oleh aplikasi *client* menjadi objek. Tabel menjelaskan salah satu contoh format pertukaran data antara aplikasi *client* dan *server*.

Tabel 4.11 Rancangan komunikasi data

No	Fitur Terkait	Format Data	Header
1	Pendaftaran pengguna personal. (proses pengiriman dari aplikasi client ke server dengan metode <i>POST</i>)	username= "" email= "" password= "" full_name="" phone="" gender="" birth_date="" identity_number="" address="" device_id="" type="" avatar="" identity_picture=""	-

Tabel 4.11 Rancangan komunikasi data (lanjutan)

No	Fitur Terkait	Format Data	Header
2	Pendaftaran pengguna bengkel. (proses pengiriman dari aplikasi client ke server dengan metode <i>POST</i>)	username= "" email= "" password= "" full_name="" phone="" gender="" birth_date="" identity_number="" address="" device_id="" type="" avatar="" identity_picture="" garage_name="" garage_open="" garage_close="" garage_address="" garage_latitude="" garage_longitude=""	-
3	Pencarian bantuan. (proses pengiriman dari aplikasi client ke server dengan metode <i>POST</i>)	latitude= "" longitude= "" help_type= "" message="" location_name=""	Authorization: Bearer token
4	Pemilihan pemberi bantuan. (proses pengiriman dari aplikasi client ke server dengan metode <i>POST</i>)	id_response= ""	Authorization: Bearer token
5	Respon pemberitahuan pencarian bantuan. Dipisahkan fungsi respon menerima dan menolak (proses pengiriman dari aplikasi client ke server dengan metode <i>POST</i>)	id_response= ""	Authorization: Bearer token

Tabel 4.11 Rancangan komunikasi data (lanjutan)

No	Fitur Terkait	Format Data	Header
6	Detail pemberi bantuan (proses pengiriman data dari <i>server</i> ke aplikasi <i>client</i> dengan metode <i>GET</i>)	{ "status"="", "status_code"=200, "status_message"="", "data"={ "username"="", "email"="", "full_name"="", "phone"="", "gender"=0, "avatar"="", "address"="", "type"=0, "status"=0, "latitude"=0.0, "longitude"=0.0, "garage_name"="" } }	Authorization: Bearer token
7	Mengambil data statistik pencarian bantuan. (proses pengiriman dari aplikasi <i>client</i> ke <i>server</i> dengan metode <i>POST</i>)	date_start= "" date_end= ""	Authorization: Bearer token
8	Mendapatkan data statistik pencarian bantuan. (proses pengiriman data dari <i>server</i> ke aplikasi <i>client</i> dengan metode <i>GET</i>)	{ "status"="", "status_code"=200, "status_message"="", "data"={{ "total"=0, "motor_1"=0, "motor_1_id"="", "motor_2"=0, "motor_2_id"="", "motor_3"=0, "motor_3_id"="", "car_1"=0, "car_1_id"="", "car_2"=0, "car_2_id"="", "car_3"=0, "car_3_id"="" }}, }}	Authorization: Bearer token