BAB 3
ANALISIS DAN PERANCANGAN SISTEM

Analisis dan perancangan aplikasi memerlukan tahapan yang sistematis untuk mendapatkan aplikasi yang baik dan bersesuaian dengan tujuannya. Tahapan awal dari analisis adalah menganalisis masalah dan analisis arsitektur sistem. Sedangkan untuk tahap perancangan aplikasi yaitu perancangan antarmuka.

3.1. Analisis Masalah

Kurangnya pengetahuan tentang cara memperbaiki kendaraan menjadi kendala bagi sebagian orang. Sebagian dari mereka hanya bisa menggunakannya saja akan tetapi tidak bisa memperbaiki sendiri jika ada kerusakan. Hal ini dinilai cukup wajar karena tidak semua orang dituntut untuk bisa terutama kaum wanita. Selain itu sebagian orang merasa malas untuk datang ke bengkel karena harus mengantri dan menunggu lama untuk proses perbaikan, oleh sebab itu lebih memilih mendatangkan montir ahli ke lokasi mereka. Namun yang menjadi masalah apakah semua montir bersedia untuk datang ke lokasi pemilik kendaraan.

Oleh Karena itu dibutuhkan suatu sistem yang efektif dan mudah digunakan untuk memberikan pelayanan kepada pemilik kendaraan, yaitu dalam sebuah aplikasi *mobile*. Dengan aplikasi ini masyarakat tidak perlu repot-repot untuk datang ke bengkel untuk mengantri dan menunggu proses perbaikan. Mereka dapat mendatangkan montir ahli ke tempat mereka kapanpun. Pemilik kendaraan hanya perlu memposting pekerjaan melalui sistem dan memilih montir sesuai dengan kriteria yang diinginkan.

3.2. Analisis Aplikasi Sejenis

Analisis aplikasi sejenis merupakan analisis yang akan membahas mengenai aplikasi yang menjadi acuan dalam pembangunan sebuah aplikasi. Dalam aplikasi sejenis akan dibahas mengenai berbagai hal yang ada didalam aplikasi sejenis diantaranya meliputi konten, cara menggunakan serta komponen-komponen apa

saja yang ada dalam aplikasi tersebut sehingga dapat menjadi acuan bagi kebutuhan yang akan dibangun. Berikut sedikit ulasan mengenai aplikasi yang akan dibahas.

3.2.1. Pengenalan Aplikasi Obeng

Obeng yaitu aplikasi pencari bengkel di seluruh Indonesia, kini bisa diandalkan untuk menjadi referensi bengkel terdekat bagi kendaraan, baik roda dua maupun empat. Dengan adanya aplikasi ini diharapkan dapat memberi solusi kepada pemilik kendaraan untuk memudahkan mencari bengkel yang baik di sekitar mereka. Dan juga bagi pemilik bengkel (motor dan mobil) dapat mendaftarkan bengkelnya berikut informasi, foto-foto, dan posisi (lat-long) di maps secara gratis.

Saat ini, pengguna aplikasi Obeng dapat memperoleh fitur-fitur dasar, seperti *rute* menuju bengkel, informasi lalu l*Int*as menggunakan *Google Maps*, dan tentu saja informasi seluruh bengkel terdekat yang sudah terdaftar pada aplikasi ini. Pengguna juga dapat memberikan masukan dan penilaian mereka terhadap bengkel yang mereka kunjungi.



Gambar 3. 1 Halaman Utama Aplikasi Obeng

3.2.1.1. Cara Penggunaan Aplikasi

Pada aplikasi Obeng ini terdapat 4 pilihan menu ketika masuk pada halaman utama aplikasi yaitu bengkel mobil terdekat, bengkel motor terdekat, *top review*, dan pencarian.

1. Menu bengkel mobil terdekat

Menu ini merupakan menu untuk pengguna yang ingin mencari lokasi bengkel yang menyediakan pelayanan khusus mobil. Bila diklik maka akan menampilkan lokasi bengkel mobil terdekat dengan posisi *user* saat ini dalam bentuk *maps*.

2. Menu bengkel motor terdekat

Menu ini merupakan menu untuk pengguna yang ingin mencari lokasi bengkel yang menyediakan pelayanan khusus motor. Bila diklik maka akan menampilkan lokasi bengkel motor terdekat dengan posisi *user* saat ini dalam bentuk *maps*.

3. Menu *Top ReViews*

Menu ini merupakan menu yang disediakan untuk menampilkan beberapa bengkel yang direkomendasikan oleh sistem. Bengkel yang direkomendasikan dipilih berdasarkan *rating* yang telah dilakukan oleh para *user* yang telah menggunakan jasa bengkel tersebut.

4. Menu Pencarian

Menu ini merupakan menu untuk pengguna yang ingin melakukan pencarian bengkel berdasarkan *keyword* yang diinputkan.

3.2.1.2. Komponen Pada Aplikasi Obeng

Berdasarkan percobaan yang telah dilakukan terhadap aplikasi Obeng maka didapatkan komponen-komponen yang digunakan dalam aplikasi tersebut yaitu dapat dilihat pada tabel dibawah ini :

Tabel 3. 1 Komponen aplikasi Obeng

No	Komponen	Keterangan
1	Menu	Aplikasi ini memiliki 4 menu utama, yaitu a. Bengkel Motor Terdekat b. Bengkel Mobil Terdekat *c. Top ReViews* d. Pencarian

No	Komponen	Keterangan
2	Informasi	a. Menampilkan lokasi bengkel terdekat b. Menampilkan informasi bengkel berdasarkan rekomendasi sistem
3	Fungsi aplikasi	Pada aplikasi ini terdapat beberapa fungsi, diantaranya sebagai berikut : a. Menyediakan informasi lokasi bengkel terdekat dengan posisi pengguna saat ini. b. Memberikan petunjuk jalan menuju lokasi bengkel yang ingin dituju.
4	Pemanfaatan teknologi	*GPS*

3.2.1.3. Hasil analisis Aplikasi Obeng

Berdasarkan analisis dari aplikasi Obeng maka dapat diperoleh beberapa *poInt* penting yang berkaitan dengan aplikasi yang dibangun. Fungsi utama aplikasi ini yaitu ditujukan untuk memudahkan pemilik kendaraan untuk menemukan lokasi bengkel terdekat dengan lokasinnya saat ini dan juga memberikan petunjuk jalan menuju lokasi tersebut. Aplikasi ini juga memiliki beberapa kelebihan diantaranya dapat menampilkan lokasi bengkel sesuai dengan rekomendasi dari sistem.

Namun dengan segala kelebihannya bukan berarti aplikasi Obeng ini tanpa kekurangan. Layanan yang disediakan oleh aplikasi ini hanya sebatas menampilkan informasi dalam bentuk maps, sehingga pemilik kendaraan tetap yang harus datang ke bengkel untuk melakukan perbaikan.

3.2.2. Pengenalan Aplikasi Montir *Freelancer*

Aplikasi montir *freelancer* merupakan sebuah aplikasi yang memberikan layanan kepada masyarakat mengenai kendala atau masalah yang terjadi pada kendaraannya. Pemilik kendaraan bisa menggunakan aplikasi ini jika terjadi kendala pada kendaraannya sehingga bisa memanggil montir ahli untuk datang ke lokasi mereka. Pada aplikasi ini siapapun bisa mendaftar kedalam sistem. Pengguna juga bisa menentukan sendiri montir yang ingin didatangkan.

3.2.2.1. Cara Penggunaan Aplikasi Montir *Freelancer*

Aplikasi montir *freelancer* ini terdiri dari beberapa menu, diantaranya :

1. Buat pekerjaan merupakan menu yang dipilih ketika pemilik kendaraan akan memposting pekerjaan melalui sistem.
2. Rekrut pekerja merupakan menu yang akan menampilkan daftar calon pekerja yang sudah mengajukan diri pada pekerjaan yang ditawarkan, pemilik kendaraan tinggal memilih salah satu montir untuk dipekerjakan.
3. Detail pekerjaan merupakan menu untuk menampilkan detail pekerjaan yang sedang dikerjakan.

3.2.2.2. Komponen Pada Aplikasi Montir *Freelancer*

Berdasarkan observasi yang dilakukan pada aplikasi tersebut maka didapatkan komponen yang digunakan dalam aplikasi montir *freelancer*, diantaranya yaitu :

Tabel 3. 2 Komponen aplikasi montir *freelancer*

No	Komponen	Keterangan
1	Menu	Aplikasi ini memiliki 7 menu utama, yaitu: a. Buat pekerjaan b. Cari pekerjaan c. Rekrut pekerja d. Profil e. Konfirmasi pekerjaan selesai f. Ajukan penawaran *g. Get Directions*
2	Informasi	a. Menampilkan info montir yang akan dipekerjakan
3	Fungsi aplikasi	Pada aplikasi ini terdapat beberapa fungsi, diantaranya sebagai berikut : a. Menyediakan jasa perbaikan kendaraan khususnya mobil. b. Menampilkan data montir yang akan dipekerjakan.
4	Pemanfaatan teknologi	*GPS*

3.2.2.3. Hasil Analisis Aplikasi Montir *Freelancer*

Adapun hasil analisis dari aplikasi montir *freelancer* adalah sebagai berikut :

1. Fungsi utama aplikasi ini yaitu ditujukan untuk memudahkan pemilik kendaraan untuk perbaikan kendaraannya.
2. Jasa yang ditawarkan lebih difokuskan kepada perbaikan mobil saja.
3. Untuk menjadi mitra dalam layanan ini siapapun bisa mendaftar kedalam aplikasi.
4. Pemilik kendaraan bisa memilih sendiri montir yang akan dipekerjakan.

3.2.3. Perbandingan Komponen Pada Aplikasi Sejenis

Perbandingan komponen pada aplikasi sejenis yaitu untuk membandingkan antara komponen - komponen dari kedua aplikasi sejenis. Berikut ini adalah tabel kesimpulan yang akan menjelaskan tentang komponen - komponen pendukung pada dua aplikasi sejenis yang telah dianalisis sebelumnya yaitu sebagai berikut :

Tabel 3. 3 Perbandingan Komponen kedua aplikasi

No	Komponen	Aplikasi Obeng	Aplikasi montir *freelancer*
1	Menu	Aplikasi ini memiliki 3 menu utama, yaitu bengkel mobil terdekat, bengkel motor terdekat, top re*view*s, dan pencarian	Aplikasi ini memiliki 3 menu utama, yaitu Buat pekerjaan, Detail pekerjaan, dan riwayat pekerjaan
2	informasi	Menampilkan lokasi bengkel terdekat dengan posisi pengguna saat ini.	Menampilkan info montir yang akan dipekerjakan
3	Fungsi aplikasi	Aplikasi ini memiliki fungsi utama yaitu jasa perbaikan mobil dan motor. Dimana sistem menampilkan informasi lokasi bengkel dan *rute* menuju lokasi tersebut.	Aplikasi ini memiliki fungsi utama yaitu jasa perbaikan mobil. Tetapi dengan mendatangkan montir ke lokasi pemilik kendaraan.
4	Pemanfaatan teknologi	*GPS*	*GPS*

3.2.4. Kesimpulan Hasil Analisis Aplikasi Sejenis

Dari hasil analisis yang dilakukan dengan melakukan observasi terhadap kedua aplikasi tersebut maka dapat disimpulkan :

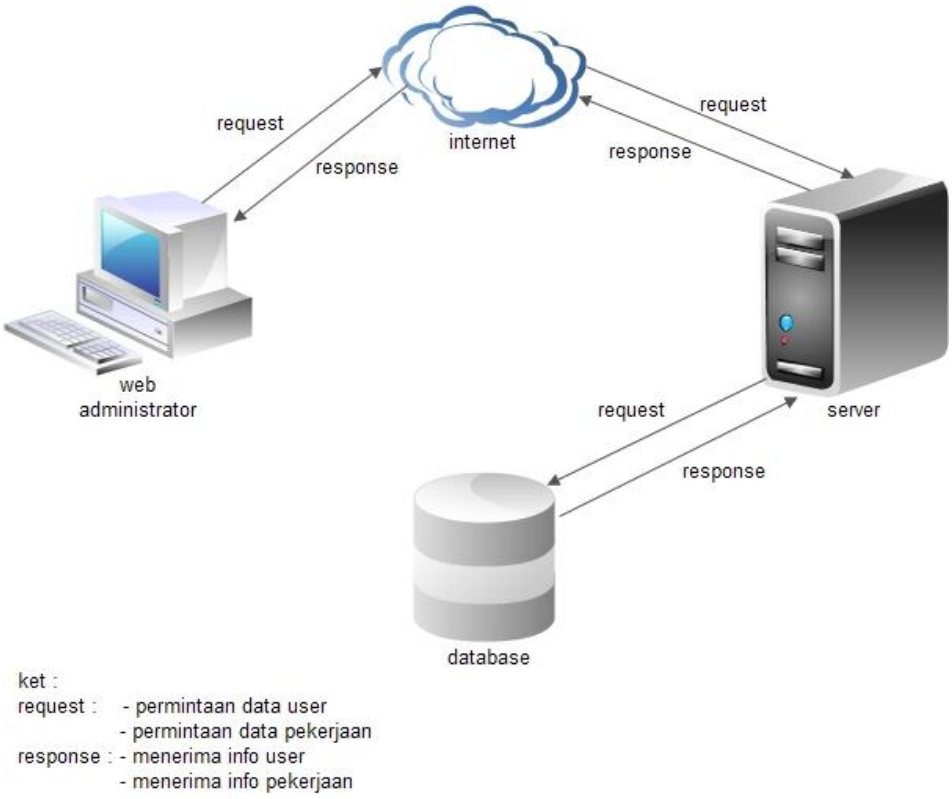
1. Pada kedua aplikasi tersebut, walaupun keduanya tampak sama namun setelah dianalisis lebih lanjut kedua aplikasi tersebut memiliki beberapa perbedaan. Terutama dalam aturan bisnis, dimana aplikasi obeng menampilkan lokasi bengkel dalam bentuk *maps* dan menampilkan *rute* menuju lokasi tersebut, sedangkan aplikasi yang dibangun montir/bengkel yang mendatangi lokasi pemilik kendaraan.

3.3. Analisis arsitektur sistem

Analisis arsitektur sistem bertujuan untuk mengidentifikasi arsitektur yang akan dibangun berdasarkan dua subsistem web dan *mobile*.

1. Web

Platform web adalah salah satu subsistem yang dipilih untuk pembangunan dari perangkat lunak ini. Pengguna perangkat lunak *platform* ini yaitu administrator. Administrator bertugas untuk mengolah data. Berikut adalah Gambar 3.2 Arsitektur Perangkat Lunak pada *Platform* Web. Gambar ini menggambarkan secara keseluruhan arsitektur sistem pada *platform* web.



Gambar 3. 2 arsitektur sistem *platform* web

Berikut adalah deskripsi dari Gambar 3.2 Arsitektur Perangkat Lunak pada *Platform* Web :

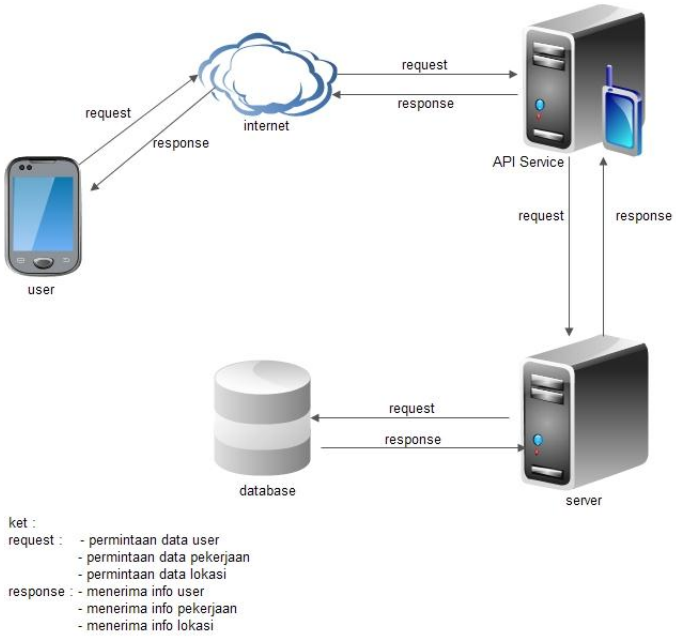
a. Sub sistem web admin melakukan *request* perm*Int*aan data ke *server* melalui jaringan *Int*ernet.

b. *Server* menerima *request* data dan mengambil data sesuai perm*Int*aan dari *database*.

c. *Server* mengirimkan data yang dim*Int*a melalui jaringan *Int*ernet ke komputer admin.

2. *Mobile*

Platform Mobile adalah salah satu subsistem yang dipilih untuk pembangunan dari perangkat lunak ini. Arsitektur perangkat lunak pada *platform Mobile* mengambarkan bagaimana perangkat lunak saling ber*Int*eraksi seperti diilustrasikan pada Gambar 3.3 Arsitektur Perangkat Lunak pada *Platform Mobile*. Gambar tersebut menggambarkan keseluruhan arsitektur sistem pada *platform mobile*.



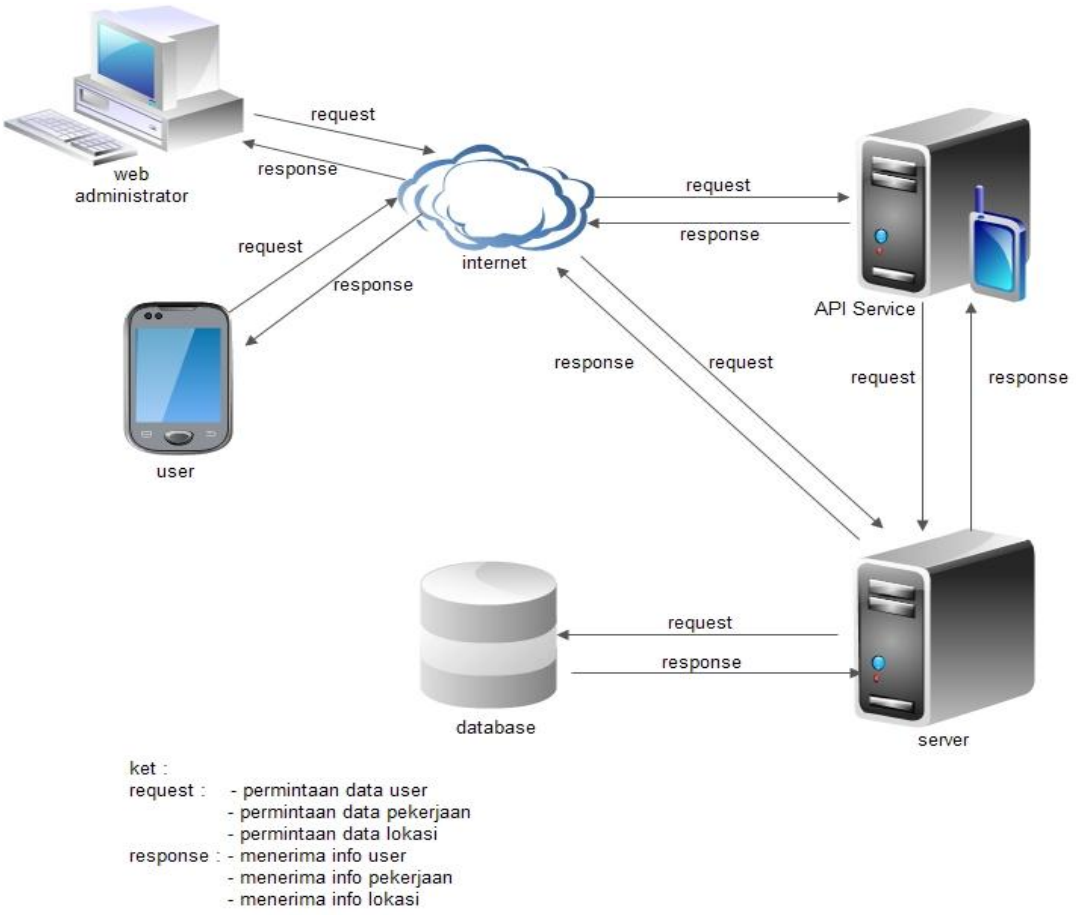
Gambar 3. 3 arsitektur sistem pada *platform mobile*

Berikut adalah deskripsi dari Gambar 3.3 Arsitektur Perangkat Lunak pada *Platform Mobile*:

a. Perangkat *Mobile* pengguna melakukan *request* data ke *server* melalui API
b. *Server* menerima *request* data dari *server* dan menentukan jenis *request* yang dim*Int*a
c. Jika *server* menerima perm*Int*aan lokasi maka perm*Int*aan data akan diteruskan ke *server* google place.

d. Jika *server* menerima perm*Int*aan data maka *server* akan mengambil data yang ada di *database*
e. Setelah *server* menerima data yang dim*Int*a data tersebut akan dikembalikan dalam bentuk JSON untuk diproses perangkat *Mobile* pengguna.

Arsitektur sistem yang digambarkan pada Gambar 3.3 Arsitektur Perangkat Lunak pada *Platform Mobile* merupakan penggambaran sistem berdasarkan perspektif dua subsistem, sedangkan pada Gambar 3.4 Arsitektur Sistem Keseluruhan menggambarkan arsitektur sistem dari perspektif secara keseluruhan.



Gambar 3. 4 arsitektur sistem secara keseluruhan

3.4. Analisis Kebutuhan Non Fungsional

Analisis kebutuhan non fungsional menggambarkan kebutuhan sistem yang menitikberatkan pada properti perilaku yang dimiliki oleh sistem, diantaranya kebutuhan perangkat keras, perangkat lunak, serta *user* sebagai bahan analisis kekurangan dan kebutuhan yang harus dipenuhi dalam perancangan sistem yang akan diterapkan. Kebutuhan non fungsional dapat dilihat pada tabel 3.4 dan 3.5.

Spesifikasi kebutuhan non fungsional.

Tabel 3. 4 Spesifikasi kebutuhan non fungsional platform mobile

Nomor SKPL	Spesifikasi Kebutuhan non fungsional
SKPL-NF-01	Memberikan informasi mengenai kemungkinan kerusakan pada kendaraan pemilik kendaraan.
SKPL-NF-02	Memberikan rekomendasi petunjuk arah menuju lokasi yang akan dituju

Tabel 3. 5 Spesifikasi kebutuhan non fungsional platform web

Nomor SKPL	Spesifikasi Kebutuhan non fungsional
SKPL-NF-01	Sistem yang dibangun menggunakan sistem operasi web.
SKPL-NF-02	Sistem yang dibangun dengan spesifikasi *hardware* yang memenuhi standar minimum kebutuhan.
SKPL-NF-03	Sistem yang dibangun menggunakan Bahasa pemrograman PHP

3.4.1. Analisis Kebutuhan Data

Setelah melakukan proses analisis kebutuhan sistem, maka analisis selanjutnya adalah analisis kebutuhan data. Analisis ini bertujuan untuk memudahkan dalam perancangan informasi. Adapun analisis kebutuhan data yang diperlukan antara lain :

1. Data bengkel atau montir

   Data yang akan ditampilkan adalah nama bengkel atau nama montir, alamat, nomor telepon.

2. Data *user* pemilik kendaraan

   Data yang dibutuhkan yaitu identitas pemilik kendaraan. Data ini diinputkan sendiri oleh *user* pada saat proses pendaftaran.

3. Data kerusakan kendaraan

   Data yang dibutuhkan yaitu gejala kerusakan kendaraan dan penyebabnya.

3.4.2. Analisis Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi antara lain :

1. Perangkat Keras Aplikasi *Backend*

Pada aplikasi *backend* yang berupa web, aplikasi dijalankan pada sebuah perangkat komputer dengan spesifikasi minimum kebutuhan perangkat keras komputer sebagai berikut :

Tabel 3. 6 *minimum requirement* perangkat keras *backend*

No	Perangkat Keras	Spesifikasi
1	Monitor	14”
2	Prosesor	*Int*el Pentium Dual Core
3	Memori / RAM	1GB
4	Harddisk	40gb
5	VGA Card	VGA Onboard
6	Keyboard	Standar
7	Mouse	Standar

2. Perangkat Keras Aplikasi Montir *Freelance*

Aplikasi yang akan dibangun dijalankan pada sebuah perangkat yaitu *Smartphone* dengan sistem operasi *Android*. Kebutuhan perangkat keras untuk *Smartphone* tersebut yaitu :

Tabel 3. 7 *minimum requirement* aplikasi montir freelance

No	Perangkat Keras	Spesifikasi
1	Sistem Operasi	Versi 4.0 (*Ice Cream Sandwich*)
2	*Display*	854 x 480 piksel
3	Memori	4gb / 512mb RAM
4	Teknologi	GPS, kamera

3.4.3. Analisis Kebutuhan Perangkat Lunak

Analisis perangkat lunak terdiri dari spesifikasi minimum perangkat yang dibutuhkan. Spesifikasi perangkat lunak untuk membangun aplikasi ini adalah sebagai berikut :

Tabel 3. 8 spesifikasi perangkat lunak

No	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	*Windows*
2	Aplikasi	IDE Eclipse, JDK dan *Android* SDK, ADT, Dreamweaver
3	Bahasa Pemrograman	PHP, Java
4	DBMS	MySQL

5	*Browser*	Mozilla firefox

3.4.4. Analisis Pengguna Sistem (*User*)

Analisis dan kebutuhan pengguna dimaksudkan untuk mengetahui seberapa jauh pengguna sistem ini. Hal ini bertujuan untuk mengetahui karakteristik pengguna dalam mengoperasikan sistem yang telah ada, maupun sistem usulan.

Pengguna yang akan terlibat dalam sistem adalah:

1. Admin, sebagai pengelola aplikasi *backend* (web). Bertugas untuk mengelola dan mengontrol kegiatan dalam sistem.
2. Pemilik Kendaraan, sebagai *user frontend*. Bertugas untuk menggunakan aplikasi sesuai dengan hak aksesnya.
3. Montir, sebagai *user frontend*. Bertugas untuk menggunakan aplikasi sesuai dengan hak aksesnya.

Berikut karakteristik pengguna sistem yang ditunjukan oleh tabel 3.9

Tabel 3. 9 karakteristik pengguna sistem

Kategori Pengguna	Hak Akses	Tingkat Pendidikan	Tingkat Keterampilan
Administrator	Mengelola kegiatan dalam sistem	Minimal S1 Teknik Informatika	Mengerti pengelolaan *database*, *database* SQL dan koneksi data
Pemilik Kendaraan	Mencari dan memanggil montir freelance melalui system	-	Mengerti penggunaan *smartphone android*, mengerti *Int*ernet
Montir	Mendapatkan pekerjaan melalui sistem	Minimal SMA/SMK	Mengerti penggunaan *smartphone android*, mengerti *Int*ernet, ahli dibidang otomotif

3.5. Analisis Kebutuhan Fungsional

Analisis kebutuhan fungsional menggambarkan proses kegiatan yang akan diterapkan dalam sebuah sistem dan menjelaskan kebutuhan yang diperlukan sistem agar sistem dapat berjalan dengan baik serta sesuai dengan kebutuhan. Kebutuhan fungsional dapat dilihat pada tabel 3.10 Spesifikasi kebutuhan fungsional.

Tabel 3. 10 Spesifikasi kebutuhan fungsional *platform mobile*

Nomor SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-01	Subsistem menyediakan fasilitas *login* untuk *user*
Nomor SKPL	**Spesifikasi Kebutuhan Perangkat Lunak**
SKPL-F-02	Subsistem menyediakan fasilitas untuk *user* membuat dan membagikan pekerjaan
SKPL-F-03	Subsistem menyediakan fasilitas untuk mencari pekerjaan
SKPL-F-04	Subsistem menyediakan fasilitas untuk melihat detail pekerjaan
SKPL-F-05	Subsistem menyediakan fasilitas menampilkan peta lokasi serta petunjuk jalan menuju lokasi.
SKPL-F-06	Subsistem menyediakan fasilitas bagi *user* untuk melihat dan mengedit profil.
SKPL-F-07	Subsistem menyediakan fasilitas bagi *user* untuk melihat riwayat pekerjaan

Tabel 3. 11 Spesifikasi kebutuhan fungsional *platform* web

Nomor SKPL	Spesifikasi Kebutuhan Perangkat Lunak
SKPL-F-01	Menyediakan fasilitas bagi admin untuk *login*
SKPL-F-02	Menyediakan fasilitas bagi admin untuk *logout*
SKPL-F-03	Menyediakan fasilitas bagi admin untuk mengolah data pemilik kendaraan
SKPL-F-04	Menyediakan fasilitas bagi admin untuk mengolah data *freelancer*
SKPL-F-05	Menyediakan fasilitas bagi admin untuk mengolah data pekerjaan

3.6. Perancangan Fungsional Sistem

Pada tahap ini dilakukan pemodelan sistem untuk menentukan fungsi-fungsi yang dapat dilakukan oleh sistem aplikasi serta menentukan kelas yang dibutuhkan untuk realisasi fungsi-fungsi sistem yang telah dianalisis sebelumnya dan mendeskripsikannya kedalam bentuk diagram.

3.6.1 *Use Case Diagram*

Use Case Diagram merupakan bagian tertinggi dari fungsionalitas yang dimiliki sistem yang akan menggambarkan bagaimana seseorang atau aktor akan menggunakan dan memanfaatkan sistem. Diagram ini juga mendeskripsikan apa saja yang akan dilakukan oleh sistem. *Use Case* terdiri dari tiga bagian yaitu identifikasi aktor, *Use Case Diagram* dan deskripsi *Use Case*.

1. Identifikasi Aktor

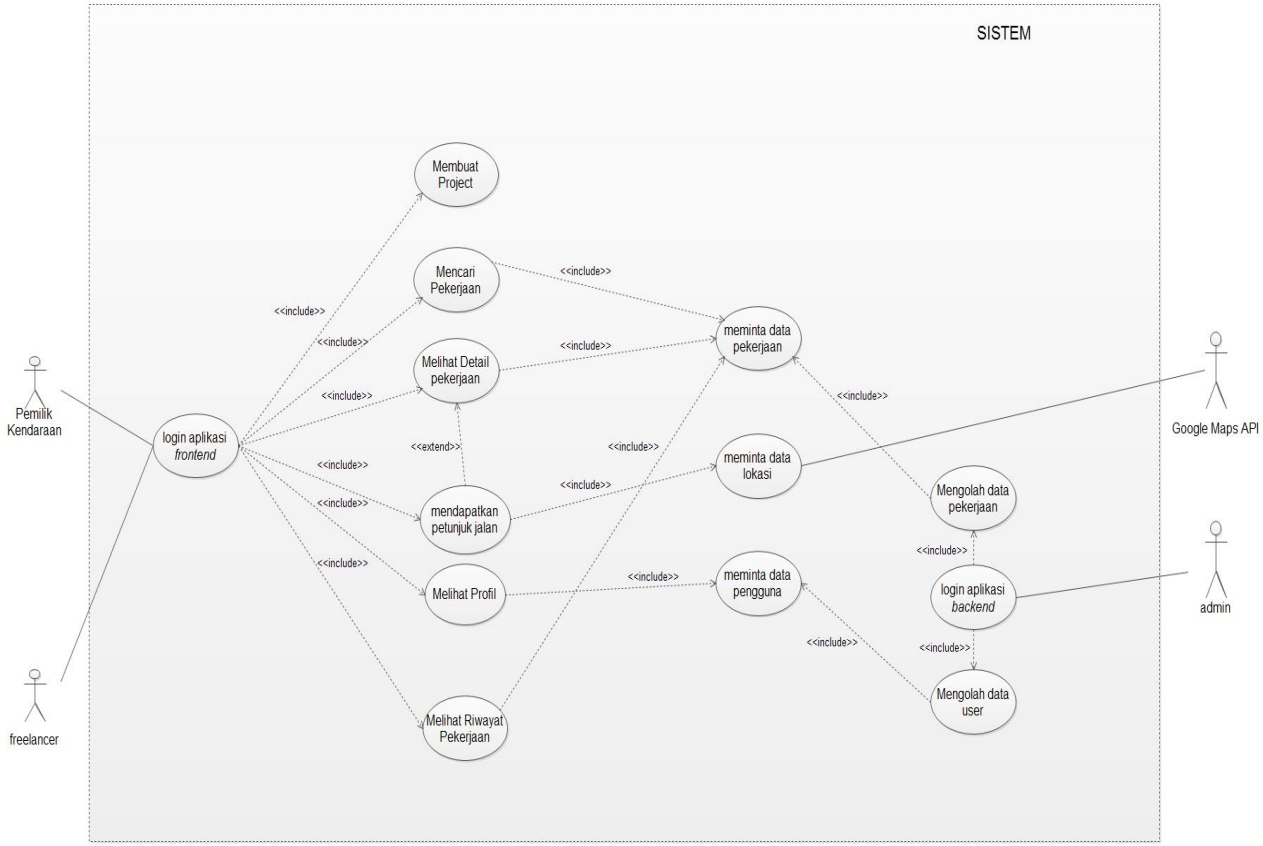
Aktor yang berperan dalam menjalankan aplikasi dapat dilihat pada tabel 3.12

Tabel 3. 12 identifikasi aktor

No	Aktor	Deskripsi
1	Pemilik Kendaraan	Merupakan aktor yang mengakses dan menggunakan aplikasi *frontend.* Aktor ini dapat membagikan pekerjaan dan merekrut pekerja sesuai keinginannya.
2	*Freelancer*	Merupakan aktor yang mengakses dan menggunakan aplikasi *frontend.* Aktor ini dapat mencari pekerjaan yang sesuai dengan lokasinya.
3	Admin	Merupakan aktor yang akan melakukan pemeliharaan pada sistem web.
4	Google Maps API	Merupakan aktor yang berfungsi untuk mengetahui lokasi

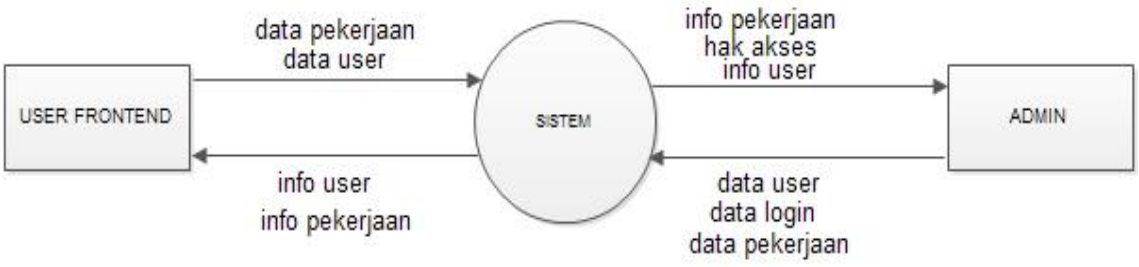
Pada Gambar 3.3 merupakan *Int*eraksi antara aktor dengan sistem. Dimana terdiri dari 4 aktor. Pemilik kendaraan yaitu pengguna ber*Int*eraksi dengan aplikasi *frontend*, pengguna dapat membagikan pekerjaan, mendapatkan informasi pada aplikasi dan merekrut pekerja sesuai keinginannya, aktor ke 2 yaitu *freelancer* atau montir/pemilik bengkel ber*Int*eraksi dengan *frontend* untuk mencari pekerjaan yang dekat dengan lokasinya, aktor ke 3 yaitu admin ber*Int*eraksi dengan *backend* untuk mengolah data, dan aktor ke 4 yaitu google maps API sebagai penyedia layanan maps.

Use Case Diagram aplikasi *frontend*



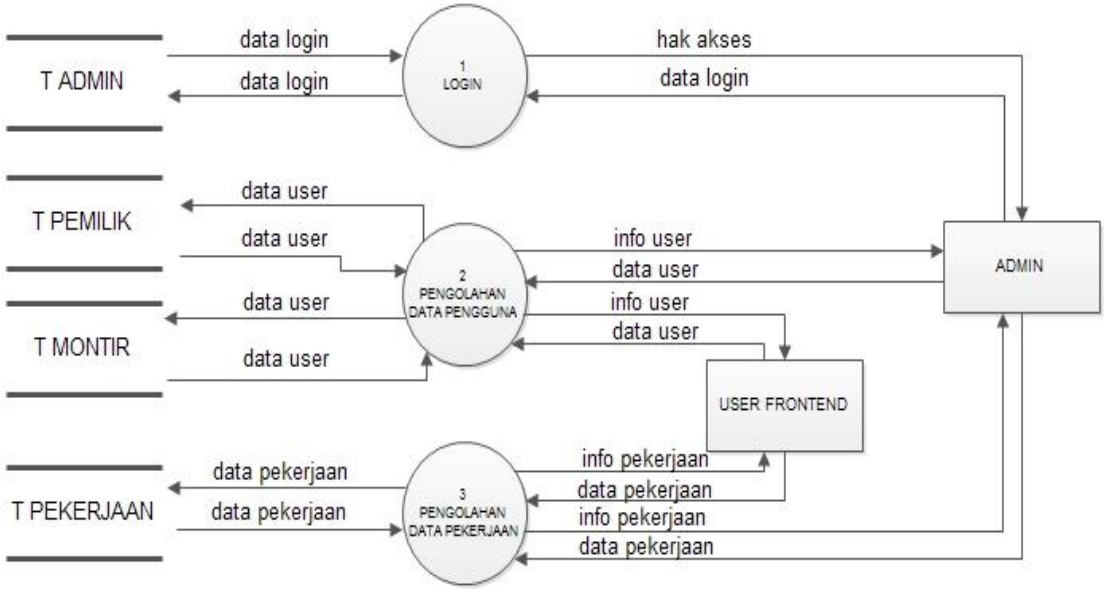
Gambar 3. 5 Use Case Diagram *frontend*

2. *Diagram* konteks *Backend*



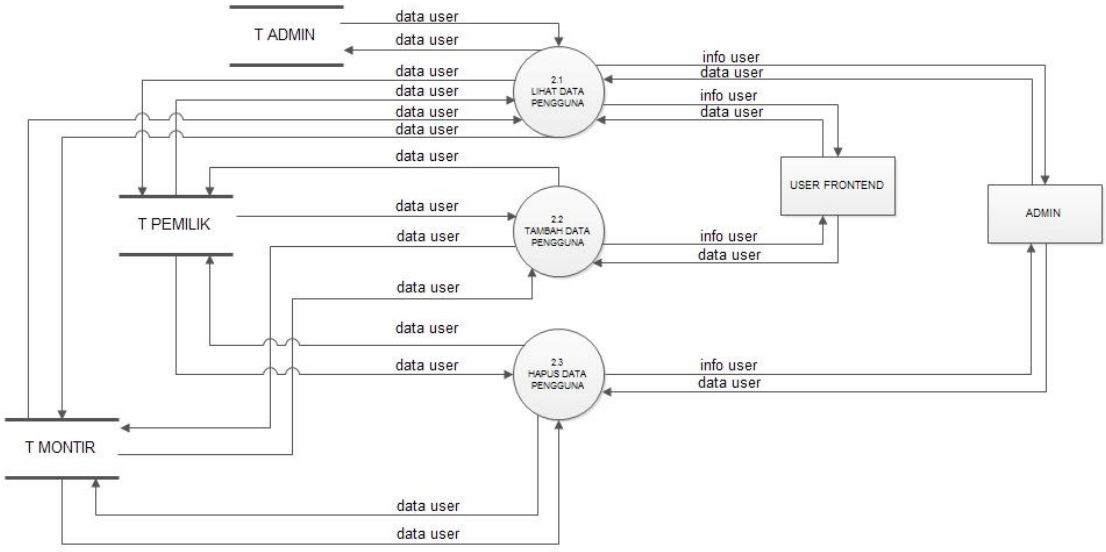
Gambar 3. 6 Diagram Konteks backend

3. *DFD* aplikasi *backend*



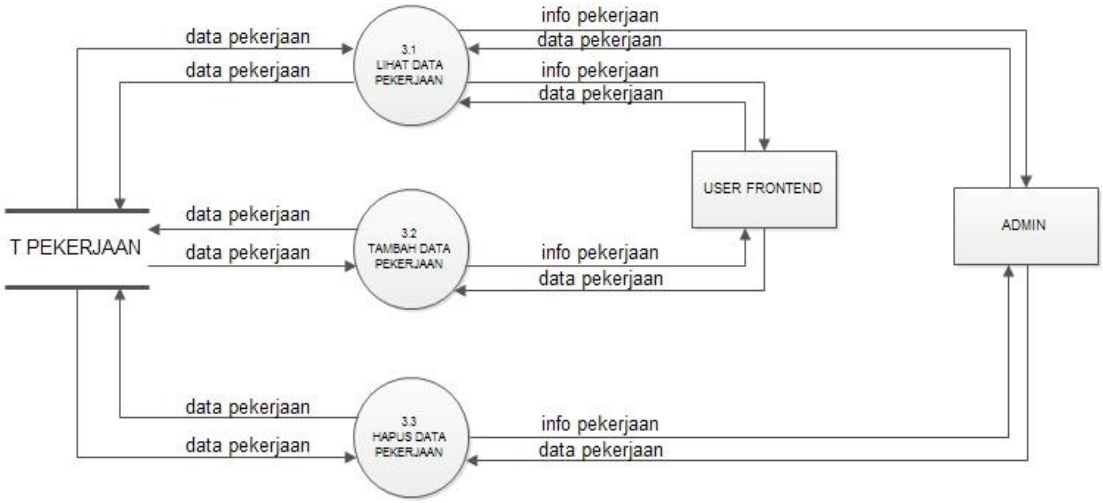
Gambar 3. 7 DFD level 1 backend

4. DFD level 2 proses 2 aplikasi *backend*



Gambar 3. 8 DFD level 2 proses 2 backend

5. DFD level 2 proses 3 aplikasi *backend*



Gambar 3. 9 DFD level 2 proses 3 backend

6. Deskripsi *Use Case*

Tabel 3. 13 Deskripsi Use Case Diagram

No	Usecase	Deskripsi
UC-01	*Login* aplikasi *Frontend*	Fungsionalitas bagi pengguna *frontend* agar bisa menggunakan aplikasi
UC-02	Membuat *project*	Fungsionalitas untuk membuat projek yang akan dibagikan dalam aplikasi
UC-03	Mencari Pekerjaan	Fungsionalitas untuk *freelancer* yang akan melakukan pencarian pekerjaan
UC-04	Melihat Detail Pekerjaan	Fungsionalitas untuk pemilik kendaraan dan *freelancer* mendapatkan detail informasi tentang pekerjaan yang sedang dilakukan
UC-05	Mendapatkan petunjuk jalan	Fungsionalitas untuk pengguna untuk mendapatkan petunjuk jalan
UC-06	Melihat Profil	Fungsionalitas untuk melihat profil pengguna, berisi keterangan berupa identitas pengguna
UC-07	Melihat Riwayat Pekerjaan	Fungsionalitas untuk *user* melihat riwayat pekerjaan yang pernah dilakukan
UC-08	Mem*Int*a data pekerjaan	Berfungsi untuk mem*Int*a data pekerjaan dari *database*
UC-09	Mem*Int*a data lokasi	Berfungsi untuk mem*Int*a data lokasi dari *database*
UC-10	Mem*Int*a data pengguna	Berfungsi untuk mem*Int*a data pengguna dari *database*
UC-11	Mengolah data *user*	Fungsionalitas yang memungkinkan pengguna

		backend dapat mengelola data pengguna.
UC-12	Mengolah data pekerjaan	Fungsionalitas yang memungkinkan pengguna *backend* dapat mengelola data pekerjaan yang telah diposting
No	**Usecase**	**Deskripsi**
UC-13	*Login* aplikasi *Backend*	Fungsionalitas bagi pengguna *frontend* agar bisa menggunakan aplikasi

3.7. Skenario *Use Case*

Skenario proses-proses yang terdapat didalam *Use Case* dapat dilihat pada table-tabel berikut ini :

Skenario *Use Case login* pemilik kendaraan dapat dilihat pada tabel 3.14

Tabel 3. 14 Skenario *Use Case login*

Nomor	UC-01	
Nama *Use Case*	*Login*	
Tujuan	untuk akses pengguna agar bisa menggunakan aplikasi	
Aktor	Pemilik Kendaraan, *freelancer*	
Skenario Utama		
Kondisi Awal	Menampilkan form *login*	
Aksi Aktor		**Reaksi sistem**
		1. Menampilkan form isian *login* berupa *username* dan *password*
2. Mengisi *username* dan *password*		
3. Menekan tombol *login*		
		4. Melakukan validasi *username* dan *password*
		5. Jika data yang dimasukan benar maka akan masuk ke halaman awal pengguna berdasarkan hak akses nya.
Kondisi Akhir	Masuk ke halaman utama berdasarkan hak aksesnya	
Skenario Alternatif		
Kondisi Awal	Menampilkan form *login*	
Aksi Aktor		**Reaksi sistem**
		1. Menampilkan form isian *login* berupa *username* dan *password*
2. Mengisi *username* dan *password*		
3. Menekan tombol *login*		
		4. Melakukan validasi *username* dan *password*
		5. Jika data yang dimasukan salah maka akan menampilkan pesan kesalahan *login*
Kondisi Akhir	Menampilkan pesan kesalahan *login*	

Skenario *Use Case* buat *project* dapat dilihat pada tabel 3.15.

Tabel 3. 15 Skenario *Use Case* buat *project*

Nomor	UC-02	
Nama *Use Case*	Buat Project	
Tujuan	Membuat projek yang akan dibagikan dalam aplikasi	
Aktor	Pemilik Kendaraan	
Skenario Utama		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi sistem**
1. Pilih menu buat project		
		2. Menampilkan form deskripsi pekerjaan
3. Mengisi form deskripsi pekerjaan		
4. Menekan tombol selesai		
		5. Menampilkan detail deskripsi pekerjaan
		6. Menampilkan pesan apakah deskripsi yang diisikan sudah benar
7. Menekan tombol OK		
		8. Menyimpan kedalam *database*
Kondisi Akhir	Menyimpan kedalam *database*	
Skenario Alternatif		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi sistem**
1. Pilih menu buat project		
		2. Menampilkan form deskripsi pekerjaan
3. Mengisi form deskripsi pekerjaan		
4. Menekan tombol selesai		
		5. Menampilkan detail deskripsi pekerjaan
		6. Menampilkan pesan apakah deskripsi yang diisikan sudah benar
7. Menekan tombol batal		
		8. Kembali menampilkan form deskripsi pekerjaan
Kondisi Akhir	Membatalkan proses dan kembali ke form kosong	

Skenario *Use Case* cari *project* dapat dilihat pada tabel 3.16.

Tabel 3. 16 Skenario Use Case cari project

Nomor	UC-03	
Nama *Use Case*	Cari Project	
Tujuan	Mencari projek yang telah diposting para pemilik kendaraan	
Aktor	*Freelancer*	
Skenario Utama		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi sistem**
1. Pilih menu cari project		
		2. Menampilkan daftar pekerjaan yang

	tersedia
3. Memilih salah satu pekerjaan yang tersedia	
	4. Menampilkan detail deskripsi pekerjaan
Kondisi Akhir	Menampilkan detail deskripsi pekerjaan
Skenario Alternatif	
Kondisi Awal	Menampilkan tampilan utama aplikasi
Aksi Aktor	**Reaksi sistem**
1. Pilih menu cari project	
	2. Menampilkan pesan pekerjaan tidak tersedia
Kondisi Akhir	Menampilkan pesan pekerjaan tidak tersedia

Skenario *Use Case* detail pekerjaan dapat dilihat pada tabel 3.17

Tabel 3. 17 Skenario *Use Case* detail pekerjaan

Nomor	UC-04
Nama *Use Case*	Melihat Detail Pekerjaan
Tujuan	Menampilkan detail pekerjaan yang sedang dilakukan
Aktor	Pemilik kendaraan, *Freelancer*
Skenario Utama	
Kondisi Awal	Menampilkan tampilan utama aplikasi
Aksi Aktor	**Reaksi sistem**
1. Pilih menu detail pekerjaan	
	2. Menampilkan detail informasi pekerjaan yang sedang dilakukan
Kondisi Akhir	Menampilkan detail informasi pekerjaan yang sedang dilakukan
Skenario Alternatif	
Kondisi Awal	Menampilkan tampilan utama aplikasi
Aksi Aktor	**Reaksi sistem**
1. Pilih menu detail pekerjaan	
	2. Menampilkan pesan pekerjaan tidak tersedia
Kondisi Akhir	Menampilkan pesan pekerjaan tidak tersedia

Skenario *Use Case* mendapatkan petunjuk jalan dapat dilihat pada tabel 3.18

Tabel 3. 18 Skenario Use Case mendapatkan petunjuk jalan

Nomor	UC-05
Nama *Use Case*	Mendapatkan petunjuk jalan
Tujuan	untuk *freelancer* mendapatkan petunjuk jalan menuju pemilik kendaraan.
Aktor	*Freelancer*
Skenario Utama	
Kondisi Awal	Menampilkan tampilan utama aplikasi
Aksi Aktor	**Reaksi sistem**
1. Memilih salah satu pekerjaan yang	

tersedia	
	2. Menampilkan detail pekerjaan
3. Memilih menu dapatkan petunjuk jalan	
	4. Menampilkan rute/petunjuk jalan menuju lokasi
Kondisi Akhir	Menampilkan rute/petunjuk jalan menuju lokasi

Skenario *Use Case* profil dapat dilihat pada tabel 3.19

Tabel 3. 19 Skenario Use Case Profil

Nomor	UC-06	
Nama *Use Case*	Melihat Profil	
Tujuan	untuk melihat profil pengguna, berisi keterangan berupa identitas pengguna	
Aktor	Pemilik Kendaraan, *Freelancer*	
Skenario Utama		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi sistem**
1. Pilih menu profil		
		2. Menampilkan profil pengguna
Kondisi Akhir	Menampilkan profil pengguna	

Skenario *Use Case* riwayat pekerjaan dapat dilihat pada tabel 3.20

Tabel 3. 20 Skenario Use Case riwayat pekerjaan

Nomor	UC-07	
Nama *Use Case*	Melihat Riwayat pekerjaan	
Tujuan	untuk melihat riwayat pekerjaan yang pernah dilakukan dilakukan.	
Aktor	Pemilik Kendaraan	
Skenario Utama		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi sistem**
1. Pilih menu riwayat pekerjaan		
		2. Menampilkan halaman riwayat pekerjaan
Kondisi Akhir	Menampilkan halaman riwayat pekerjaan	
Skenario Alternatif		
Kondisi Awal	Menampilkan tampilan utama aplikasi	
Aksi Aktor		**Reaksi sistem**
1. Pilih menu riwayat pekerjaan		
		2. Menampilkan pesan error tidak ada data.
Kondisi Akhir	Menampilkan pesan error tidak ada data.	

Skenario *Use Case* Mem*Int*a data pekerjaan dapat dilihat pada tabel 3.21

Tabel 3. 21 Skenario Use Case Meminta data pekerjaan

Nomor	UC-08	
Nama *Use Case*	Mem*Int*a data pekerjaan	
Tujuan	Mem*Int*a data pekerjaan	
Aktor	Pemilik Kendaraan, Montir, *Database*	
Skenario Utama		
Kondisi Awal	-	
Aksi Aktor		**Reaksi sistem**
1. Mem*Int*a data pekerjaan		
		2. Mem*Int*a data pekerjaan
		3. Mendapatkan data pekerjaan
4. Mendapatkan data pekerjaan		
Kondisi Akhir		Mendapatkan data pekerjaan

Skenario *Use Case* Mem*Int*a data lokasi dapat dilihat pada tabel 3.22

Tabel 3. 22 Skenario Use Case Meminta data lokasi

Nomor	UC-09	
Nama *Use Case*	Mem*Int*a data lokasi	
Tujuan	Mem*Int*a data lokasi dalam bentuk *Longitude* dan *latitude*	
Aktor	Montir, google maps API	
Skenario Utama		
Kondisi Awal	Menampilkan detail pekerjaan	
Aksi Aktor		**Reaksi sistem**
1. Mem*Int*a data lokasi melalui API		
		2. Mem*Int*a data lokasi melalui API
		3. Mendapatkan data lokasi dari API
4. Mendapatkan data lokasi dari API		
Kondisi Akhir		Mendapatkan data lokasi dari API

Skenario *Use Case* Mem*Int*a data pengguna dapat dilihat pada tabel 3.23

Tabel 3. 23 Skenario Use Case MemInta data pengguna

Nomor	UC-10
Nama *Use Case*	Mem*Int*a data pengguna
Tujuan	Mem*Int*a data pengguna
Aktor	Pemilik Kendaraan, Montir, *Database*
Skenario Utama	

Kondisi Awal	-
Aksi Aktor	**Reaksi sistem**
1. Mem*Int*a data pengguna	
	2. Mem*Int*a data pengguna
	3. Mendapatkan data pengguna
4. Mendapatkan data pengguna	
Kondisi Akhir	Mendapatkan data pengguna

Skenario *Use Case* olah data pemilik kendaraan dapat dilihat pada tabel 3.24

Tabel 3. 24 Skenario Use Case olah data pemilik kendaraan

Nomor	UC-11
Nama *Use Case*	Olah data *user*
Tujuan	Memungkinkan pengguna *backend* (admin) dapat mengelola data pengguna
Aktor	Admin
Skenario Utama	
Kondisi Awal	Menampilkan tampilan utama aplikasi *backend*
Aksi Aktor	**Reaksi sistem**
1. Memilih menu olah data pengguna	
	2. Menampilkan opsi lihat data, tambah data dan hapus data pengguna
Kondisi Akhir	Menampilkan opsi lihat data, tambah data dan hapus data pengguna
Skenario Alternatif	
Kondisi Awal	Menampilkan tampilan utama aplikasi *backend*
Aksi Aktor	**Reaksi sistem**
1. Memilih menu olah data pengguna	
	2. Menampilkan pesan error data tidak tersedia
Kondisi Akhir	Menampilkan pesan error data tidak tersedia

Skenario *Use Case* olah data pekerjaan dapat dilihat pada tabel 3. 25

Tabel 3. 25 Skenario Use Case olah data pekerjaan

Nomor	UC-12
Nama *Use Case*	Olah data pekerjaan
Tujuan	Memungkinkan pengguna *backend* (admin) dapat mengelola data pekerjaan
Aktor	Admin
Skenario Utama	
Kondisi Awal	Menampilkan tampilan utama aplikasi *backend*
Aksi Aktor	**Reaksi sistem**
1. Menekan tombol olah data pekerjaan	
	2. Menampilkan opsi lihat data dan hapus data pekerjaan
Kondisi Akhir	Menampilkan opsi lihat data, tambah data dan hapus data pekerjaan
Skenario Alternatif	
Kondisi Awal	Menampilkan tampilan utama aplikasi *backend*

Aksi Aktor	Reaksi sistem
1. Menekan tombol olah data pekerjaan	
	2. Menampilkan pesan error data tidak tersedia
Kondisi Akhir	Menampilkan pesan error data tidak tersedia

Skenario *Use Case Login* aplikasi *backend* dapat dilihat pada tabel 3. 26

Tabel 3. 26 Skenario Use Case Login aplikasi *backend*

Nomor	UC-13
Nama *Use Case*	*Login* aplikasi *backend*
Tujuan	untuk akses admin agar bisa menggunakan aplikasi
Aktor	Admin
Skenario Utama	
Kondisi Awal	Menampilkan form *login*
Aksi Aktor	**Reaksi sistem**
	1. Menampilkan form isian *login* berupa *username* dan *password*
2. Mengisi *username* dan *password*	
3. Menekan tombol *login*	
	4. Melakukan validasi *username* dan *password*
	5. Jika data yang dimasukan benar maka akan masuk ke halaman awal pengguna berdasarkan hak akses nya.
Kondisi Akhir	Masuk ke halaman utama berdasarkan hak aksesnya
Skenario Alternatif	
Kondisi Awal	Menampilkan form *login*
Aksi Aktor	**Reaksi sistem**
	6. Menampilkan form isian *login* berupa *username* dan *password*
7. Mengisi *username* dan *password*	
8. Menekan tombol *login*	
	9. Melakukan validasi *username* dan *password*
	10. Jika data yang dimasukan salah maka akan menampilkan pesan kesalahan *login*
Kondisi Akhir	Menampilkan pesan kesalahan *login*

3.8. Spesifikasi Proses

Spesifikasi proses 1 proses *login* dapat dilihat pada tabel 3.27

Tabel 3. 27 Spesifikasi Proses 1 Proses Login

No	Proses	Keterangan
1.	No. Proses	1
	Nama Proses	*Login*
	Source	Admin
	Input	Data *login*
	Output	Hak Akses
	Destination	Admin
	Logika Proses	1. Admin memasukan data *login* seperti *username* dan *password* 2. Sistem akan memvalidasi kesesuaian *input* dengan data di *database* 3. Jika data sesuai maka akan masuk ke halaman utama admin dan jika salah maka akan menampilkan pesan "*username* or *password invalid*"

Spesifikasi proses 2.1 proses lihat data pengguna dapat dilihat pada tabel 3.28

Tabel 3. 28 Spesifikasi Proses 2.1 proses lihat data pengguna

No	Proses	Keterangan
2.	No. Proses	2.1
	Nama Proses	Lihat data pengguna
	Source	Admin, Pemilik Kendaraan, Montir
	Input	Data *user*
	Output	Info *user*
	Destination	Admin, Pemilik Kendaraan, Montir
	Logika Proses	1. *user* memilih menu lihat data pengguna 2. sistem menampilkan data pengguna

Spesifikasi proses 2.2 proses Tambah data pengguna dapat dilihat pada tabel 3.29

Tabel 3. 29 Spesifikasi Proses 2.2 proses Tambah data pengguna

No	Proses	Keterangan
3.	No. Proses	2.2
	Nama Proses	Tambah data pengguna
	Source	Pemilik kendaraan, montir
	Input	Data *user*
	Output	Info *user*
	Destination	Pemilik kendaraan, montir
	Logika Proses	1. pemilik kendaraan dan montir memilih menu daftar dan memasukan data pengguna 2. sistem memproses dan memasukan kedalam *database*

Spesifikasi proses 2.3 proses Hapus data pengguna dapat dilihat pada tabel 3.30

Tabel 3. 30 Spesifikasi Proses 2.3 proses hapus data pengguna

No	Proses	Keterangan
4.	No. Proses	2.3
	Nama Proses	Hapus data pengguna
	Source	Admin
	Input	Data *user*
	Output	Info *user*
	Destination	Admin
	Logika Proses	1. admin mencari data pengguna yang akan dihapus 2. menekan tombol hapus data pengguna 3. sistem melakukan proses penghapusan data pengguna

Spesifikasi proses 3.1 proses Lihat data pekerjaan dapat dilihat pada tabel 3.31

Tabel 3. 31 Spesifikasi Proses 3.1 Proses Lihat data pekerjaan

No	Proses	Keterangan
5.	No. Proses	3.1
	Nama Proses	Lihat data pekerjaan
	Source	Admin, Pemilik Kendaraan, Montir
	Input	Data pekerjaan
	Output	Info pekerjaan
	Destination	Admin, Pemilik Kendaraan, Montir
	Logika Proses	1. *User* memilih menu lihat data pekerjaan 2. sistem menampilkan data pekerjaan

Spesifikasi proses 3.2 proses Tambah data pekerjaan dapat dilihat pada tabel 3.32

Tabel 3. 32 Spesifikasi Proses 3.2 Proses Lihat data pekerjaan

No	Proses	Keterangan
6.	No. Proses	3.2
	Nama Proses	Tambah data pekerjaan
	Source	Pemilik kendaraan
	Input	Data pekerjaan
	Output	Info pekerjaan
	Destination	Pemilik kendaraan
	Logika Proses	1. pemilik kendaraan memilih menu buat *project* dan memasukan data pekerjaan 2. sistem memproses dan memasukan kedalam *database*

Spesifikasi proses 3.3 proses Hapus data pekerjaan dapat dilihat pada tabel 3.33

Tabel 3. 33 Spesifikasi Proses 3.3 Proses Hapus data pekerjaan

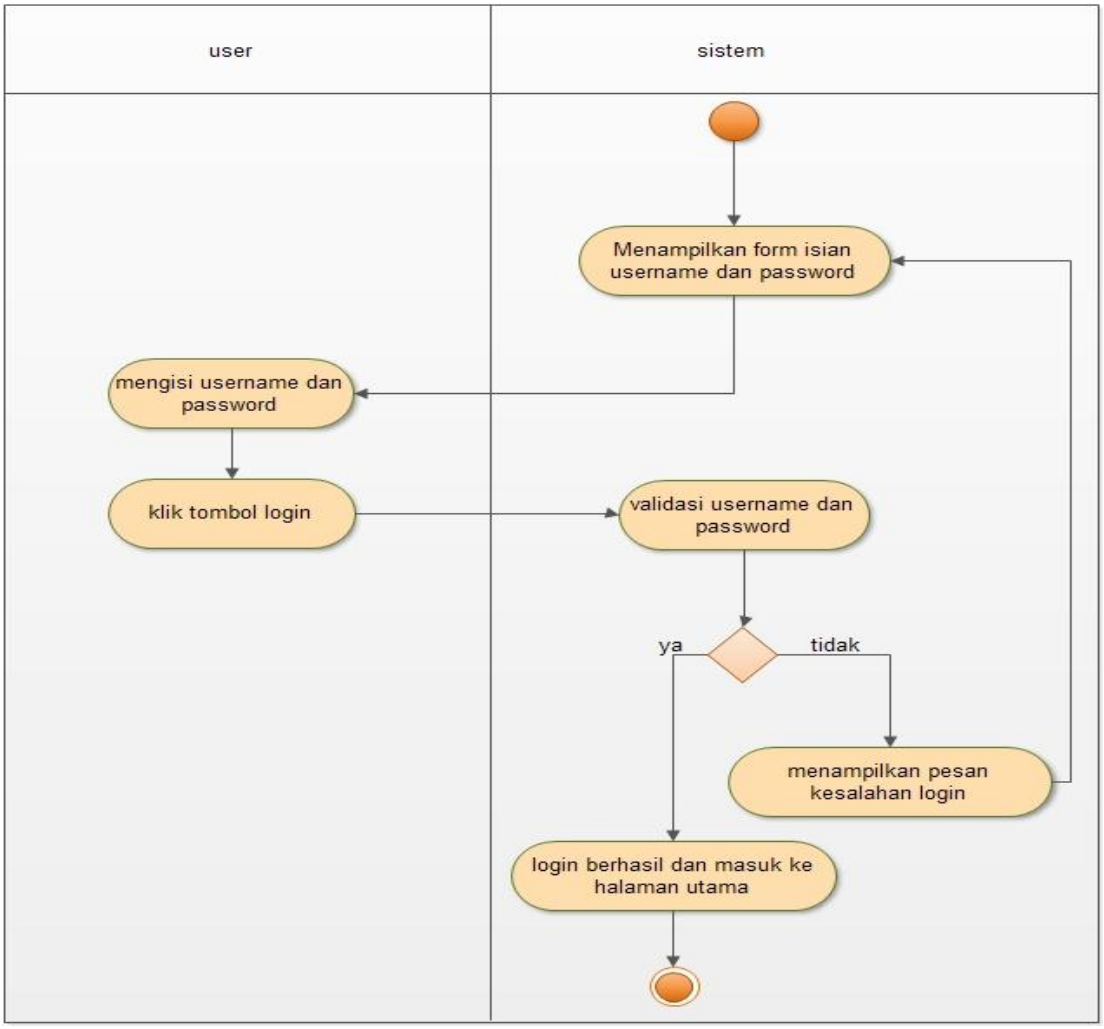
No	Proses	Keterangan
7.	No. Proses	3.3
	Nama Proses	Hapus data pekerjaan
	Source	Admin
	Input	Data pekerjaan
	Output	Info pekerjaan
	Destination	Admin
	Logika Proses	1. admin mencari data pekerjaan yang akan dihapus 2. menekan tombol hapus data pekerjaan 3. sistem melakukan proses penghapusan data pekerjaan

3.9. *Activity Diagram*

Activity diagram menggambarkan berbagai alur aktivitas dalam sistem yang dirancang, bagaimana masing-masing alur berawal, *decision* yang mungkin terjadi dan bagaimana mereka berakhir. Penggambaran *activity diagram* memiliki kemiripan dengan *flowchart diagram*. *Activity diagram* memodelkan *event-event* yang terjadi pada *Use Case* dan digunakan untuk pemodelan aspek dinamis dari sistem.

3.9.1. *Activity Diagram Login* Aplikasi *Backend*

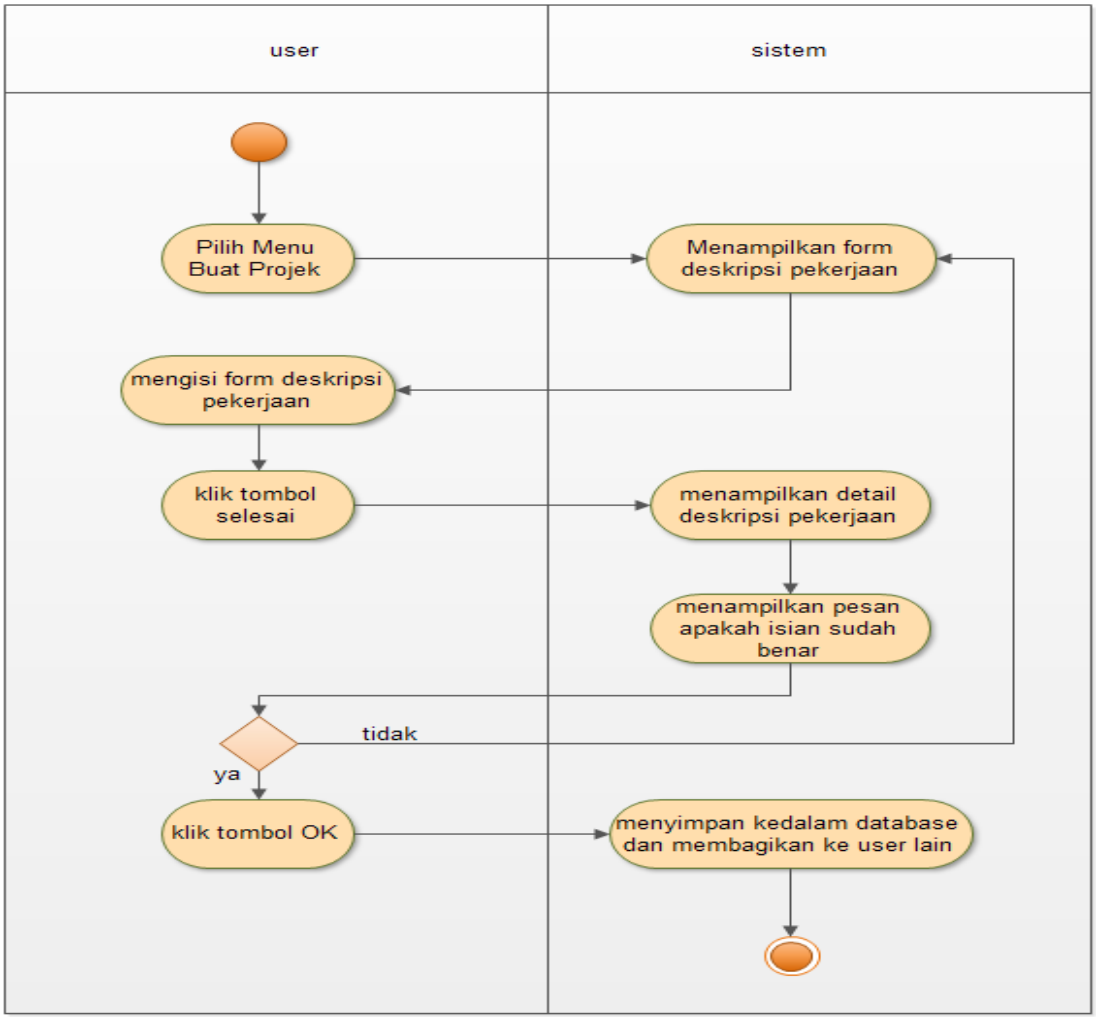
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas *Login* yang dapat dilihat pada Gambar 3.10 berikut ini.



Gambar 3. 10 *Activity Diagram Login*

3.9.2. *Activity Diagram* Buat *Project*

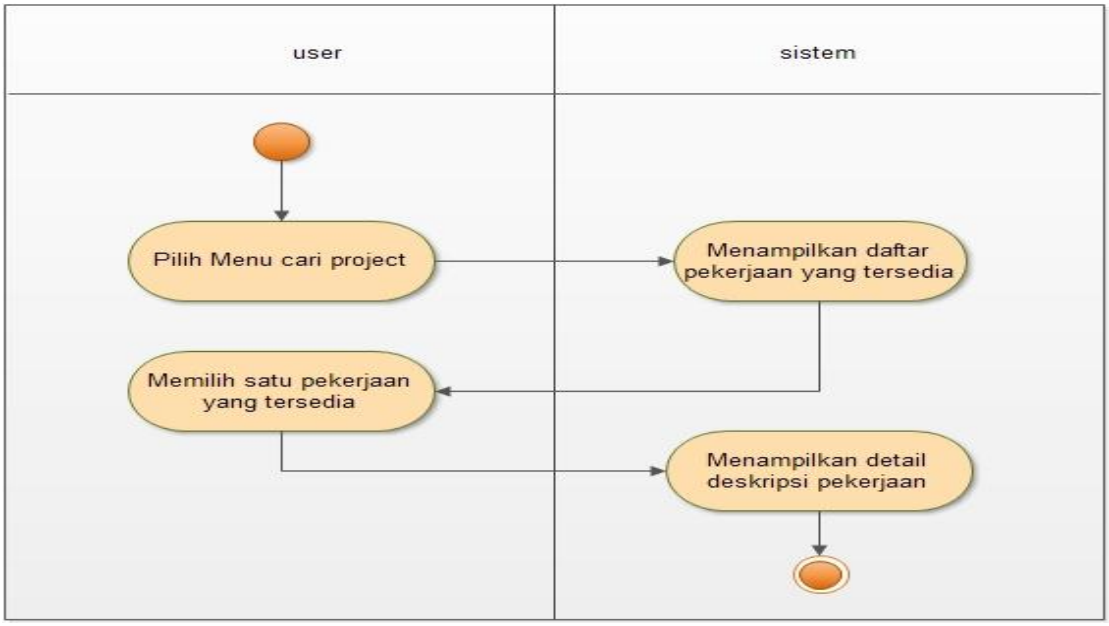
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas Buat *Project* yang dapat dilihat pada Gambar 3.11 berikut ini.



Gambar 3. 11 *Activity Diagram* Buat *Project*

3.9.3. *Activity Diagram* Cari *Project*

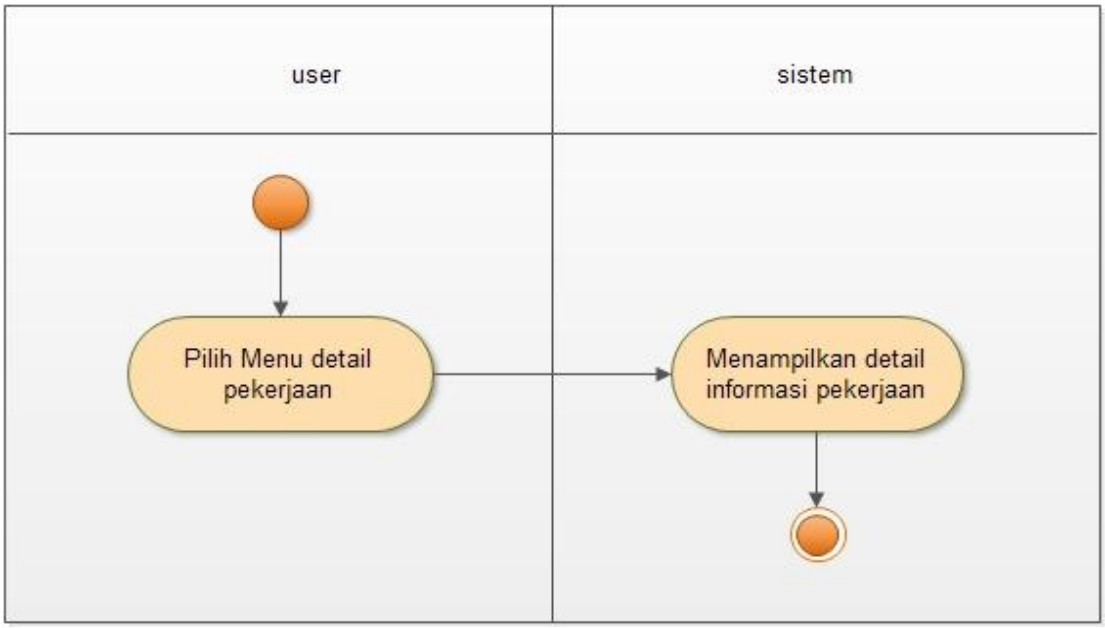
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas cari *project* yang dapat dilihat pada Gambar 3.12 berikut ini.



Gambar 3. 12 *Activity Diagram* Cari *Project*

3.9.4. *Activity Diagram* Detail Pekerjaan

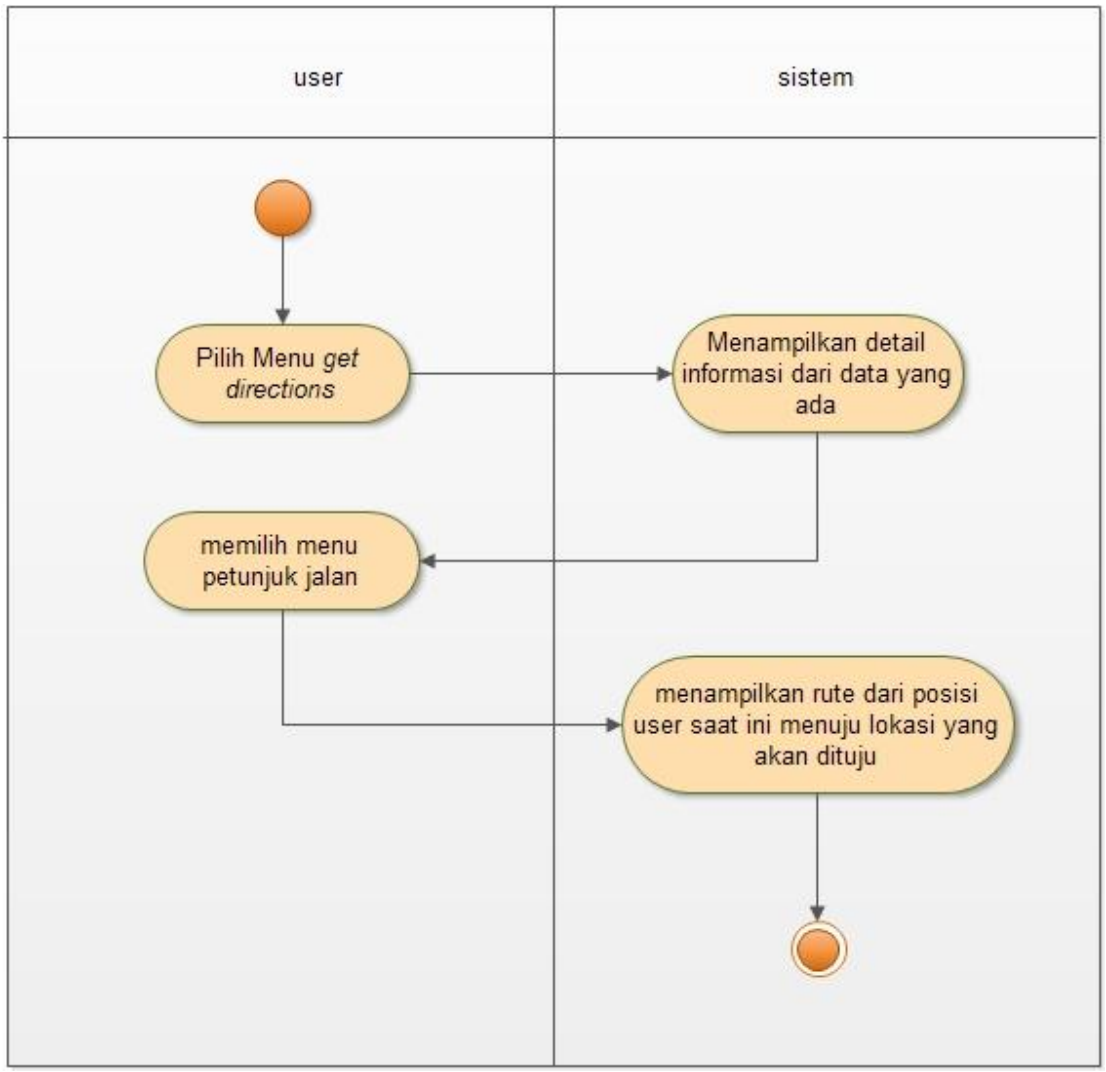
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas melihat detail pekerjaan yang dapat dilihat pada Gambar 3.13 berikut ini.



Gambar 3. 13 *Activity Diagram* detail pekerjaan

3.9.5. *Activity Diagram* Mendapatkan Petunjuk Jalan

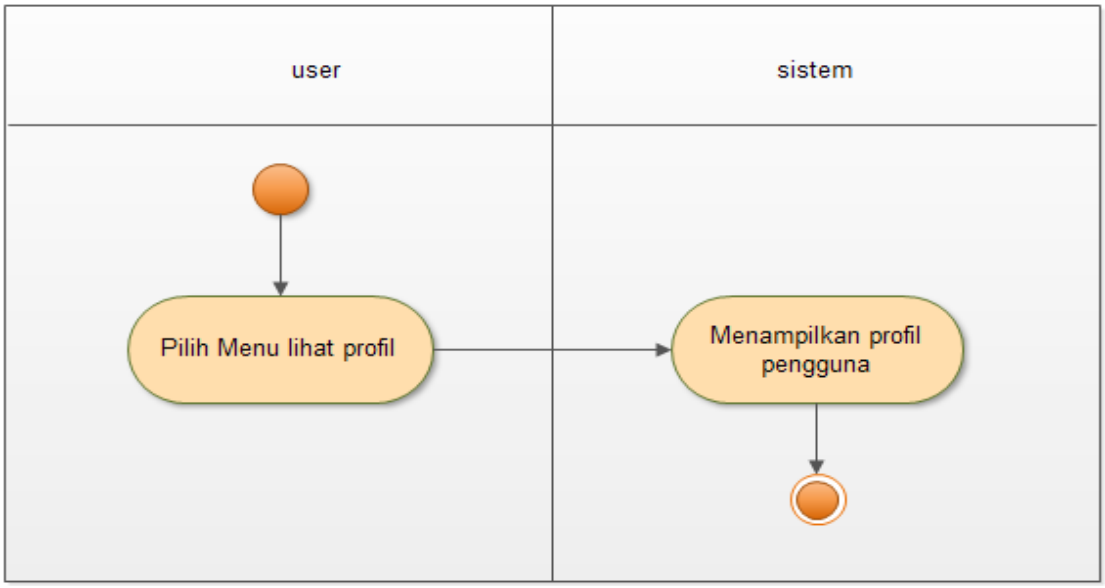
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas mendapatkan petunjuk jalan yang dapat dilihat pada Gambar 3.14 berikut ini.



Gambar 3. 14 *Activity Diagram* mendapatkan petunjuk jalan

3.9.6. *Activity Diagram* Profil

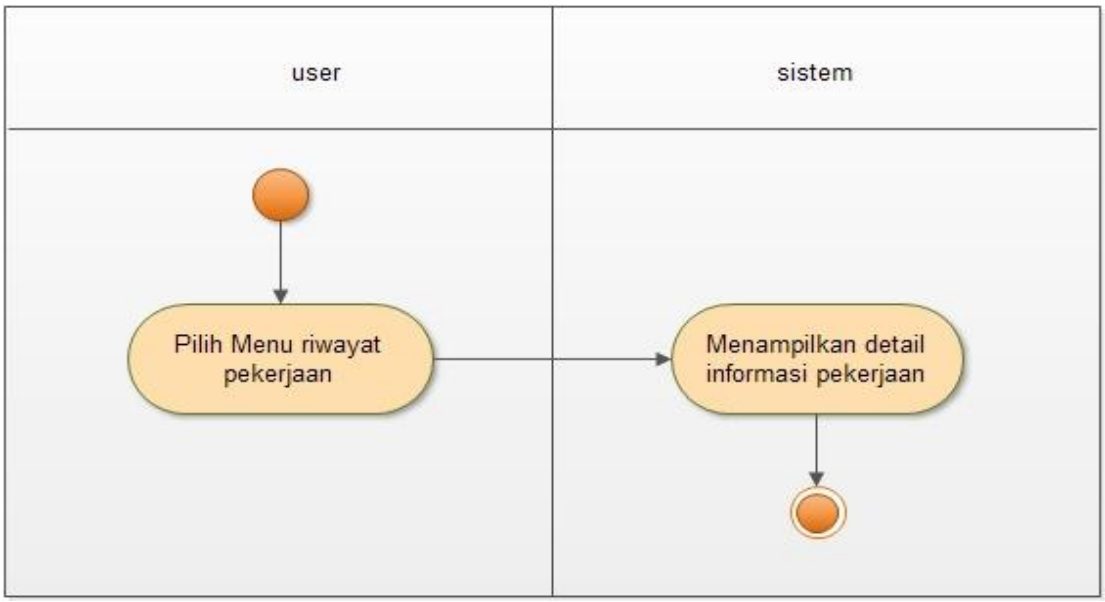
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas melihat Profil yang dapat dilihat pada Gambar 3.15 berikut ini.



Gambar 3. 15 *Activity Diagram* melihat Profil

3.9.7. *Activity Diagram* Riwayat Pekerjaan

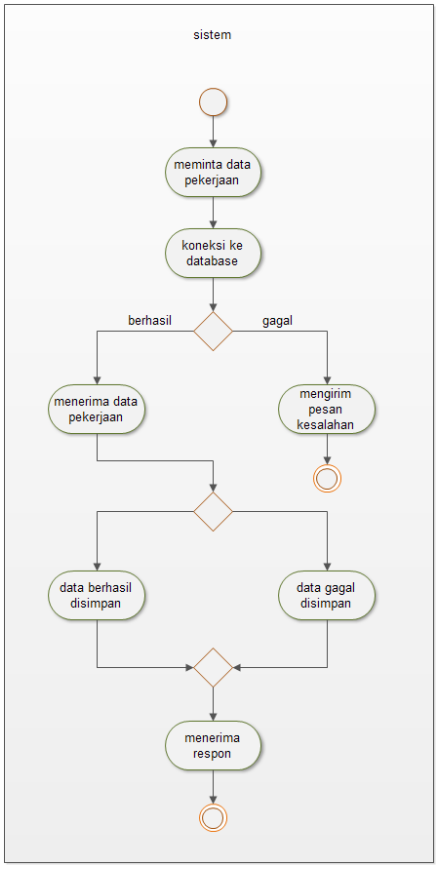
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas melihat riwayat pekerjaan yang dapat dilihat pada Gambar 3.16 berikut ini.



Gambar 3. 16 *Activity Diagram* Riwayat Pekerjaan

3.9.8. *Activity Diagram* menyimpan data pekerjaan

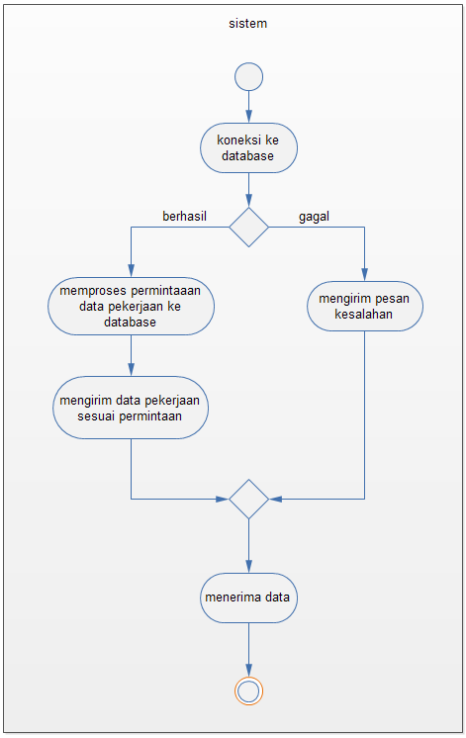
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas menyimpan data pekerjaan yang dapat dilihat pada Gambar 3.17 berikut ini.



Gambar 3. 17 *Activity Diagram* menyimpan data pekerjaan

3.9.9. *Activity Diagram* mem*Int*a data pekerjaan

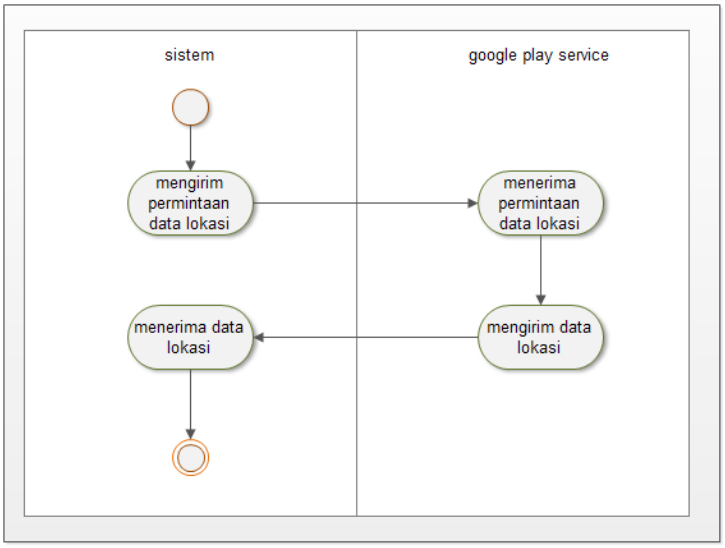
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas mem*Int*a data pekerjaan yang dapat dilihat pada Gambar 3.18 berikut ini.



Gambar 3. 18 *Activity Diagram* mem*int*a data pekerjaan

3.9.10. *Activity Diagram* mem*Int*a data lokasi

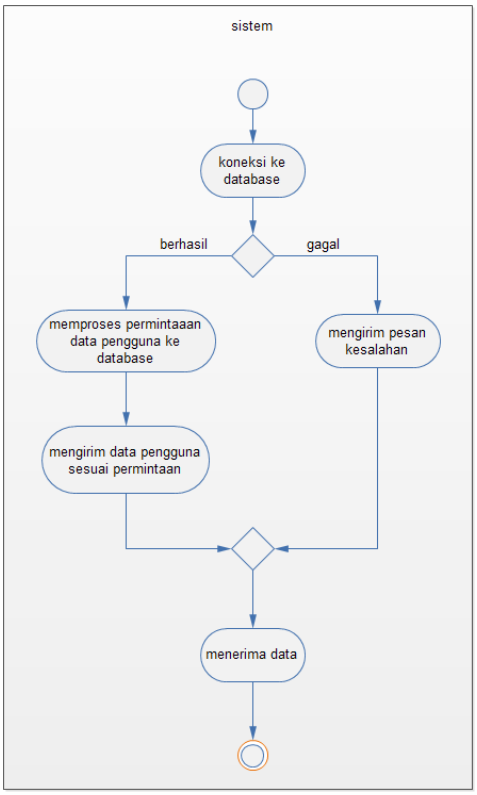
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas mem*Int*a data lokasi yang dapat dilihat pada Gambar 3.19 berikut ini.



Gambar 3. 19 *Activity Diagram* mem*int*a data lokasi

3.9.11. *Activity Diagram* mem*Int*a data pengguna

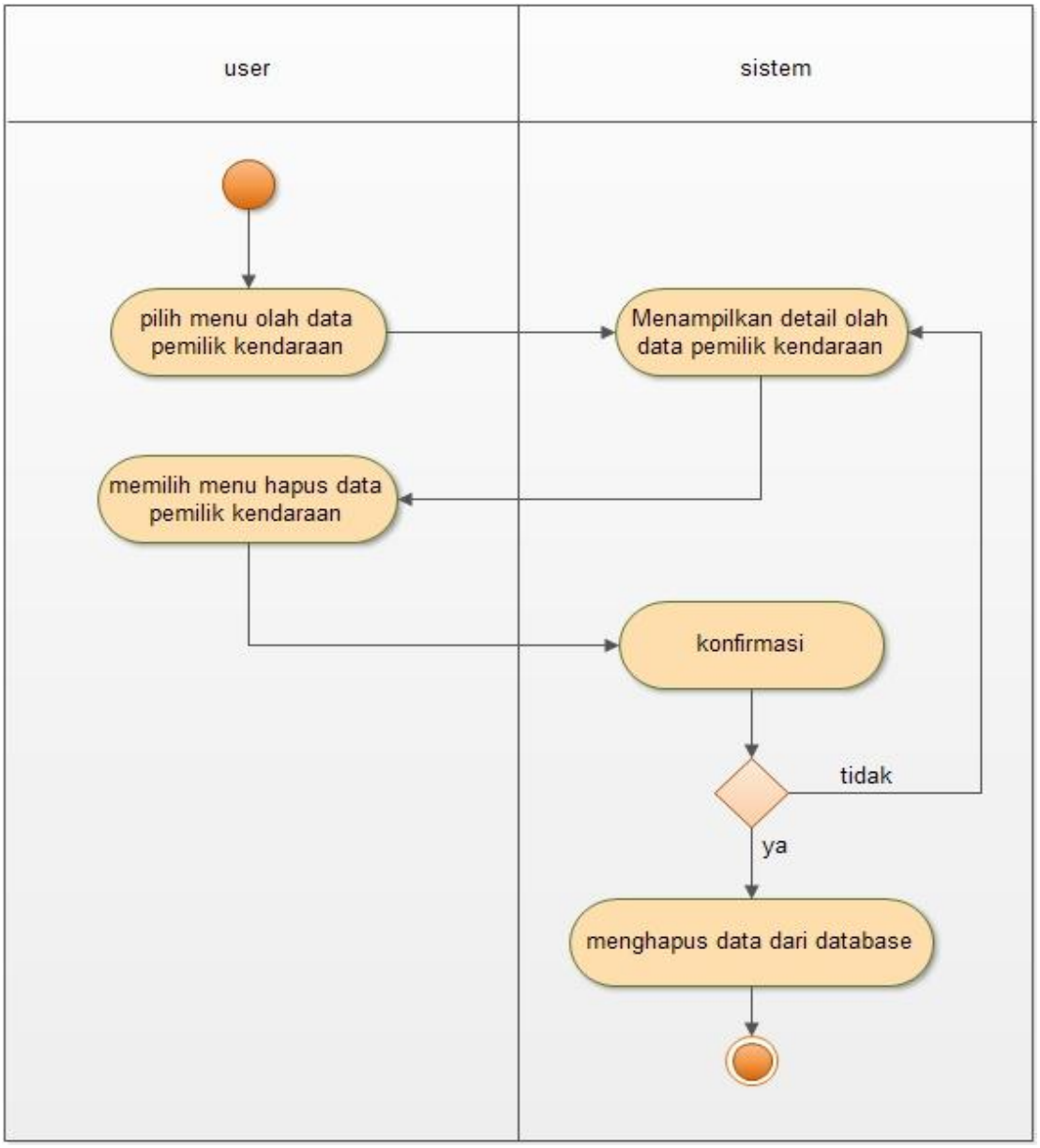
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas mem*Int*a data pengguna yang dapat dilihat pada Gambar 3.20 berikut ini.



Gambar 3. 20 *Activity Diagram* mem*int*a data pengguna

3.9.12. *Activity Diagram* Hapus Data Pemilik Kendaraan

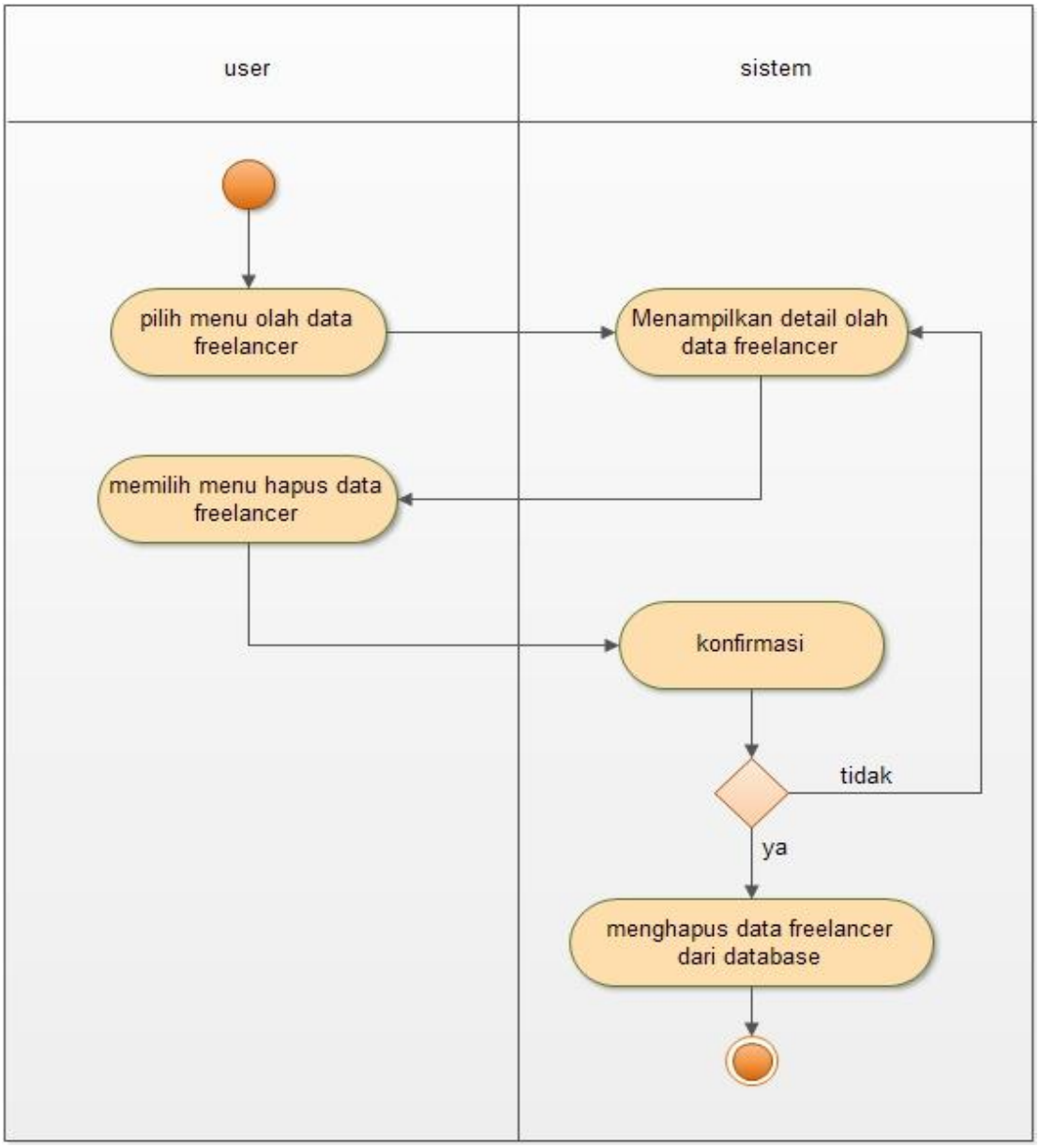
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas hapus data pemilik kendaraan yang dapat dilihat pada Gambar 3.21 berikut ini.



Gambar 3. 21 *Activity Diagram* Olah Data Pemilik Kendaraan

3.9.13. *Activity Diagram* Hapus Data *Freelancer*

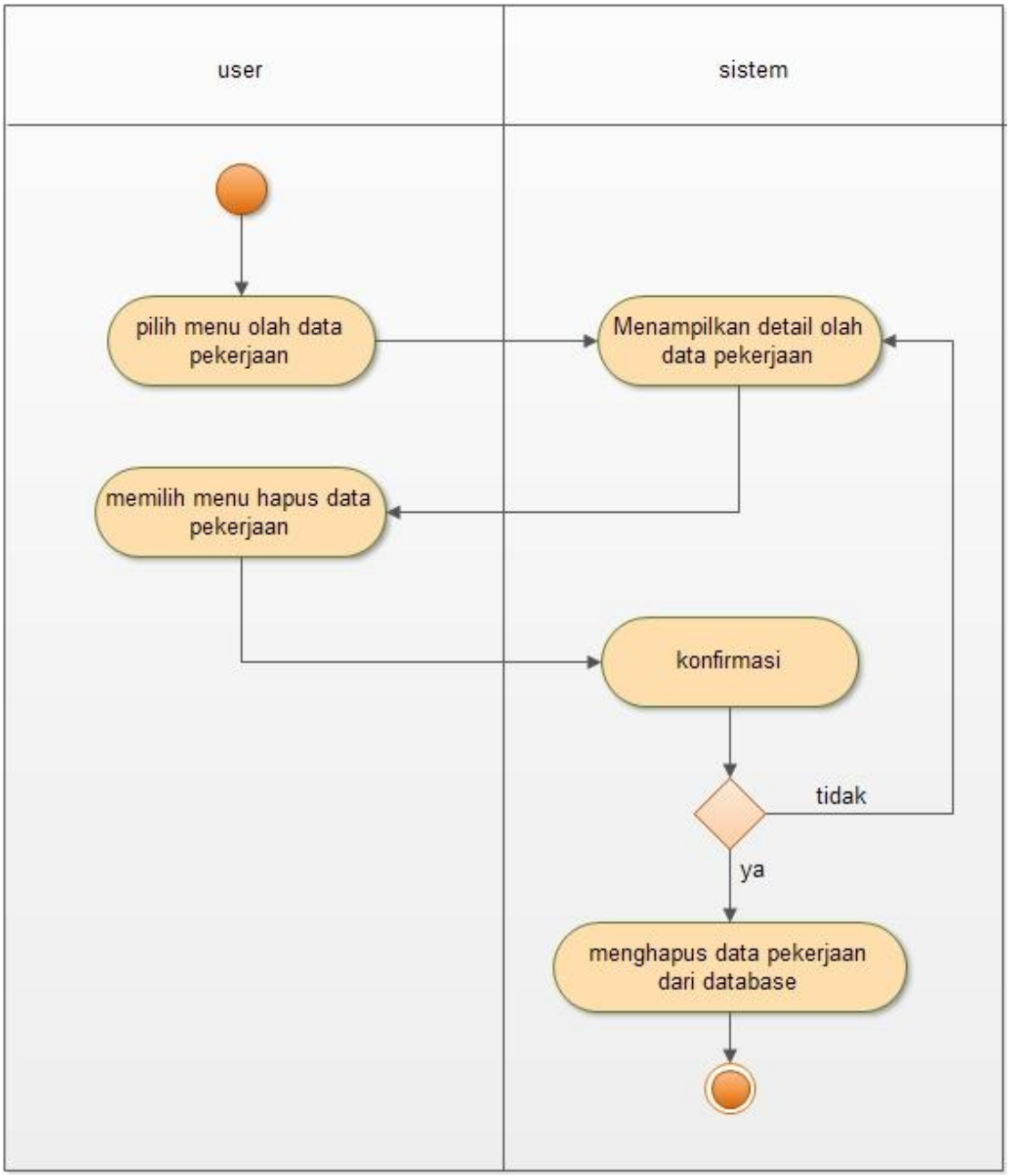
Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas hapus data *freelancer* yang dapat dilihat pada Gambar 3.22 berikut ini.



Gambar 3. 22 *Activity Diagram* Olah Data *Freelancer*

3.9.14. *Activity Diagram* Hapus Data Pekerjaan

Berikut ini adalah *diagram* yang menunjukan alur aksi pada aktivitas hapus data pekerjaan yang dapat dilihat pada Gambar 3.23 berikut ini.



Gambar 3. 23 *Activity Diagram* Olah Data Pekerjaan

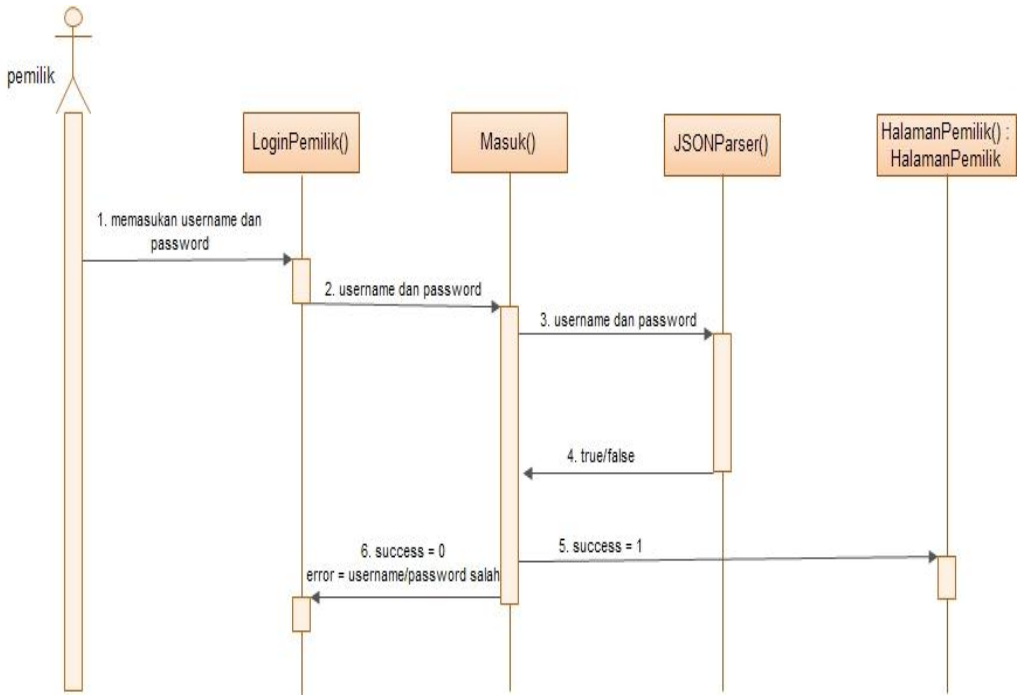
3.10. *Sequence Diagram*

Pada tahap ini *Sequence Diagram* menjelaskan secara detail urutan proses yang dilakukan sistem untuk mencapai tujuan dari *Use Case. Int*eraksi yang terjadi antar

Class, operasi apa saja yang terlibat, urutan antar operasi, dan informasi yang diperlukan oleh masing-masing operasi.

a. *Sequence Diagram Login*

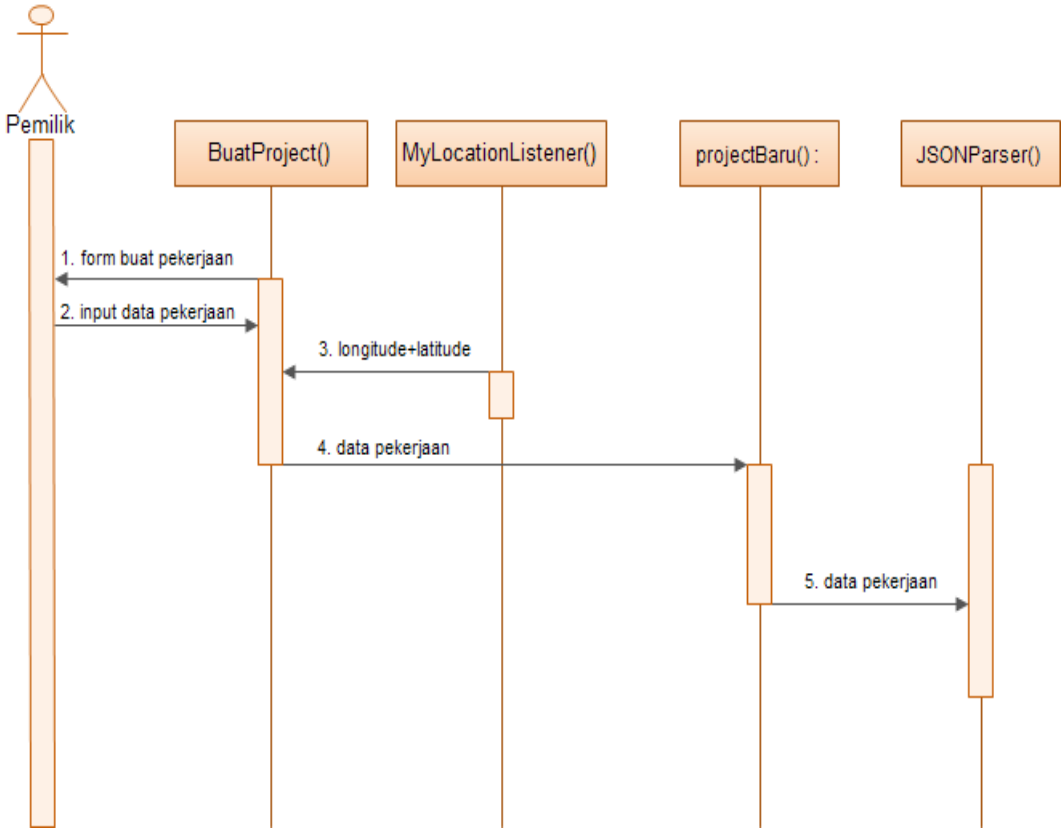
Sequence Diagram pada proses *login* akan dijelaskan pada gambar 3.24 berikut:



Gambar 3. 24 *Sequence Diagram Login*

b. *Sequence Diagram* Buat *Project*

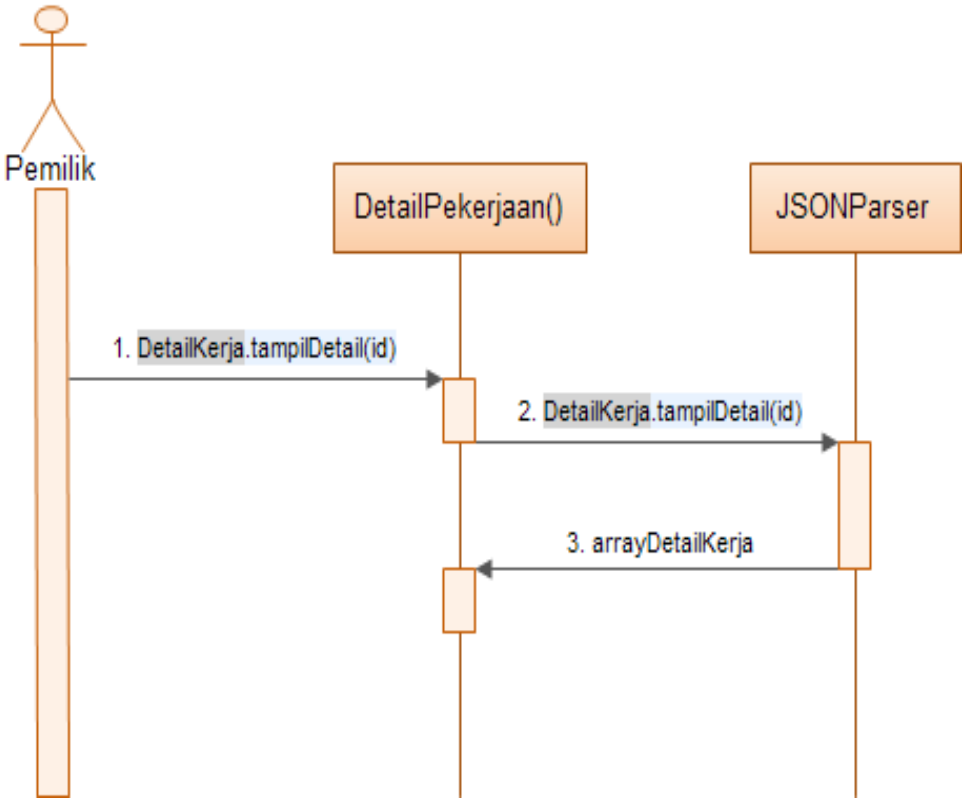
Sequence Diagram pada proses buat *Project* akan dijelaskan pada gambar 3.25 berikut



Gambar 3. 25 *Sequence Diagram* buat *Project*

c. *Sequence Diagram* Detail Pekerjaan

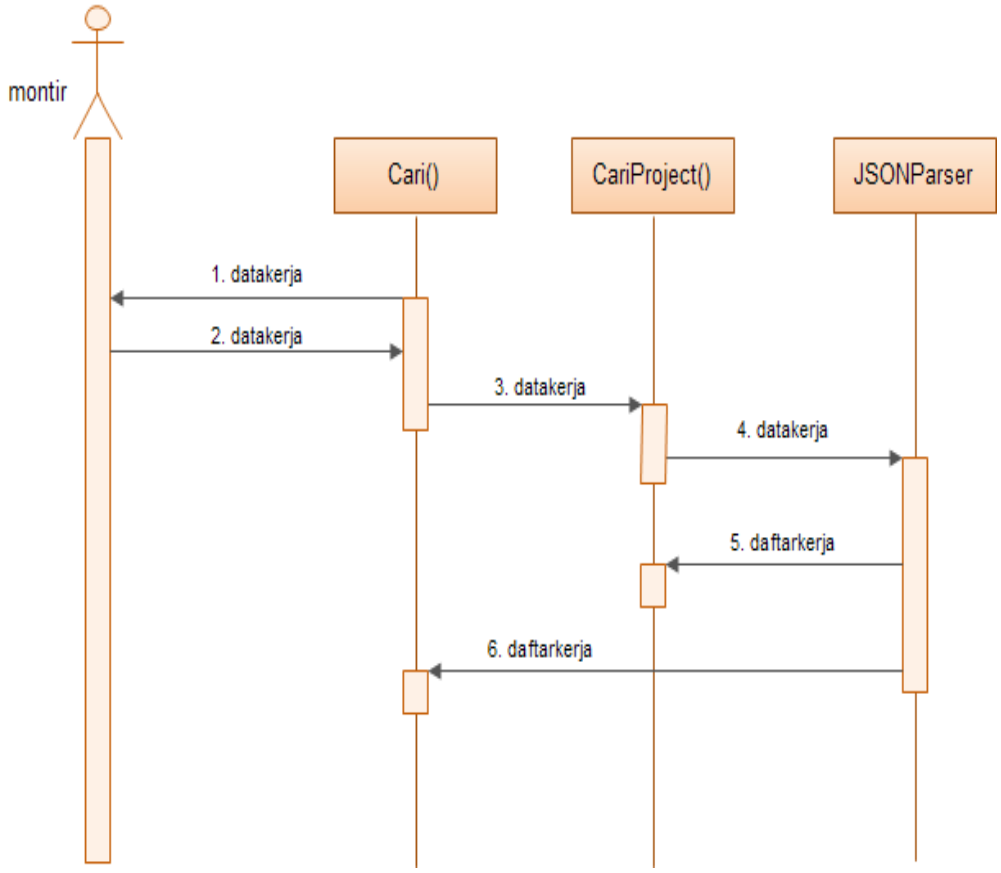
Sequence Diagram pada proses Detail Pekerjaan akan dijelaskan pada gambar 3.26 berikut



Gambar 3. 26 *Sequence Diagram* Detail Pekerjaan

d. *Sequence Diagram* Cari *project*

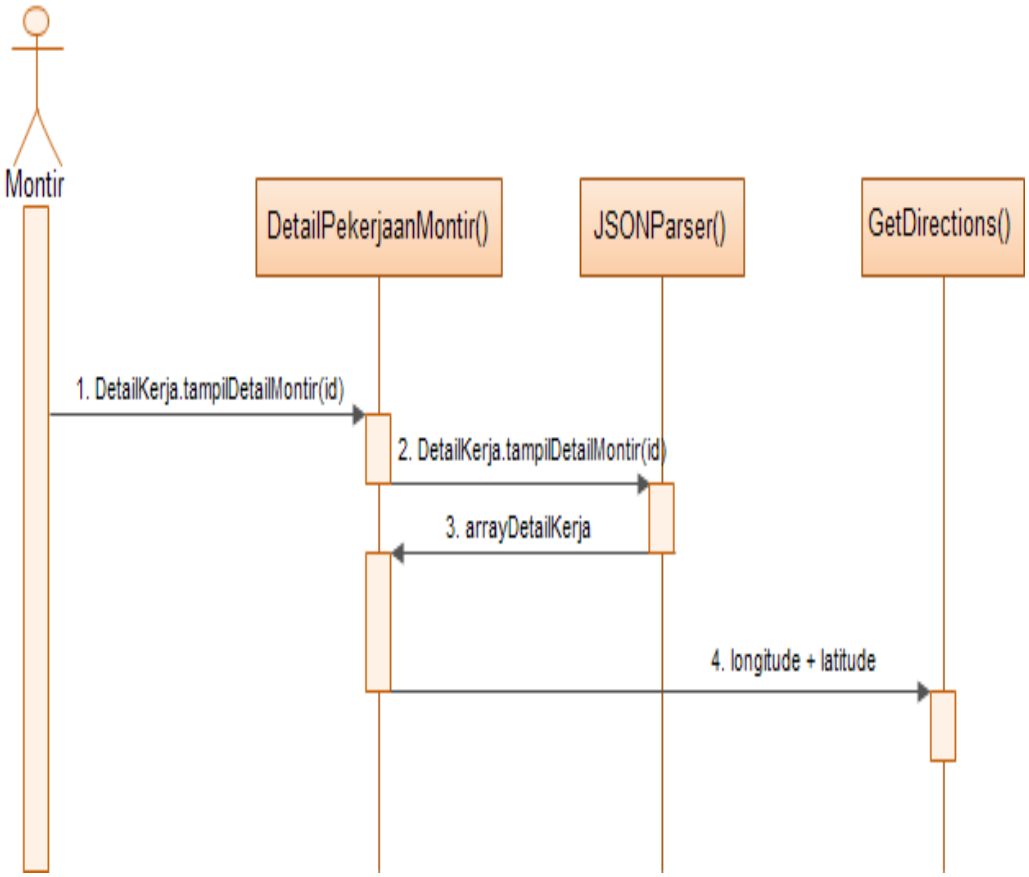
Sequence Diagram pada proses Cari *project* akan dijelaskan pada gambar 3.27 berikut



Gambar 3. 27 *Sequence Diagram* Cari *project*

e. *Sequence Diagram* petunjuk jalan

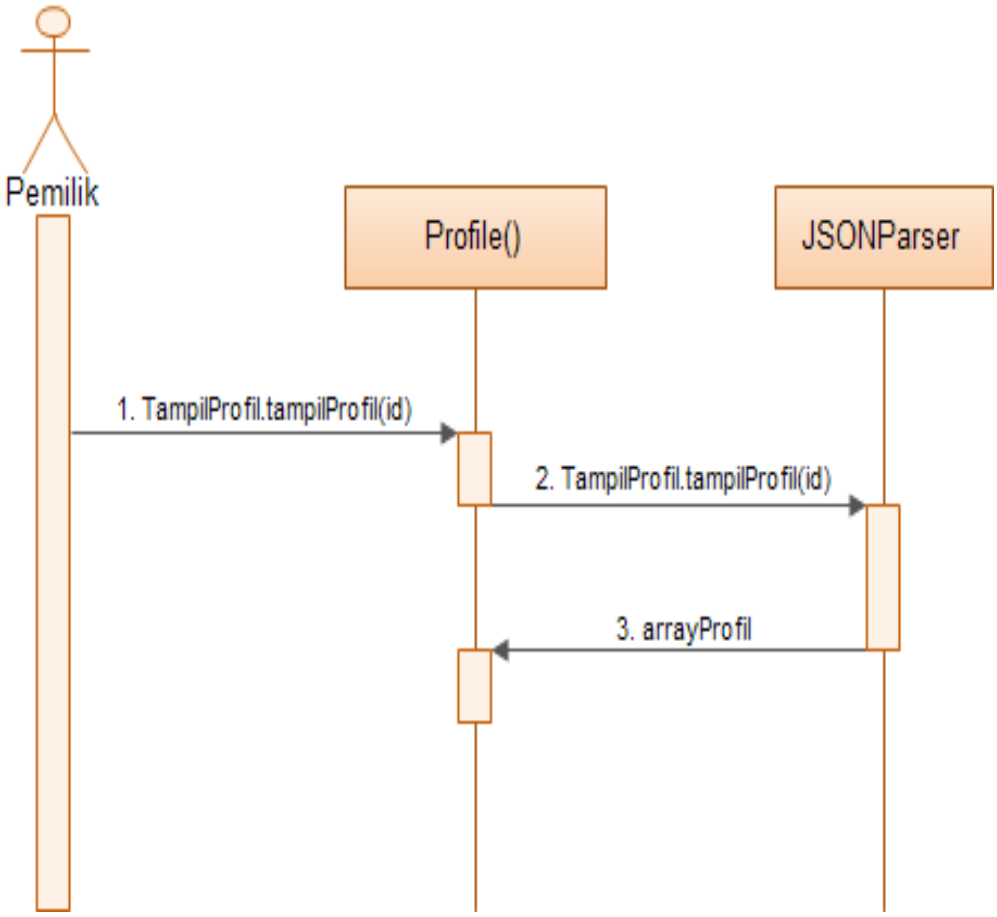
Sequence Diagram pada proses petunjuk jalan akan dijelaskan pada gambar 3.28 berikut



Gambar 3. 28 *Sequence Diagram* petunjuk jalan

f. *Sequence Diagram* Profil

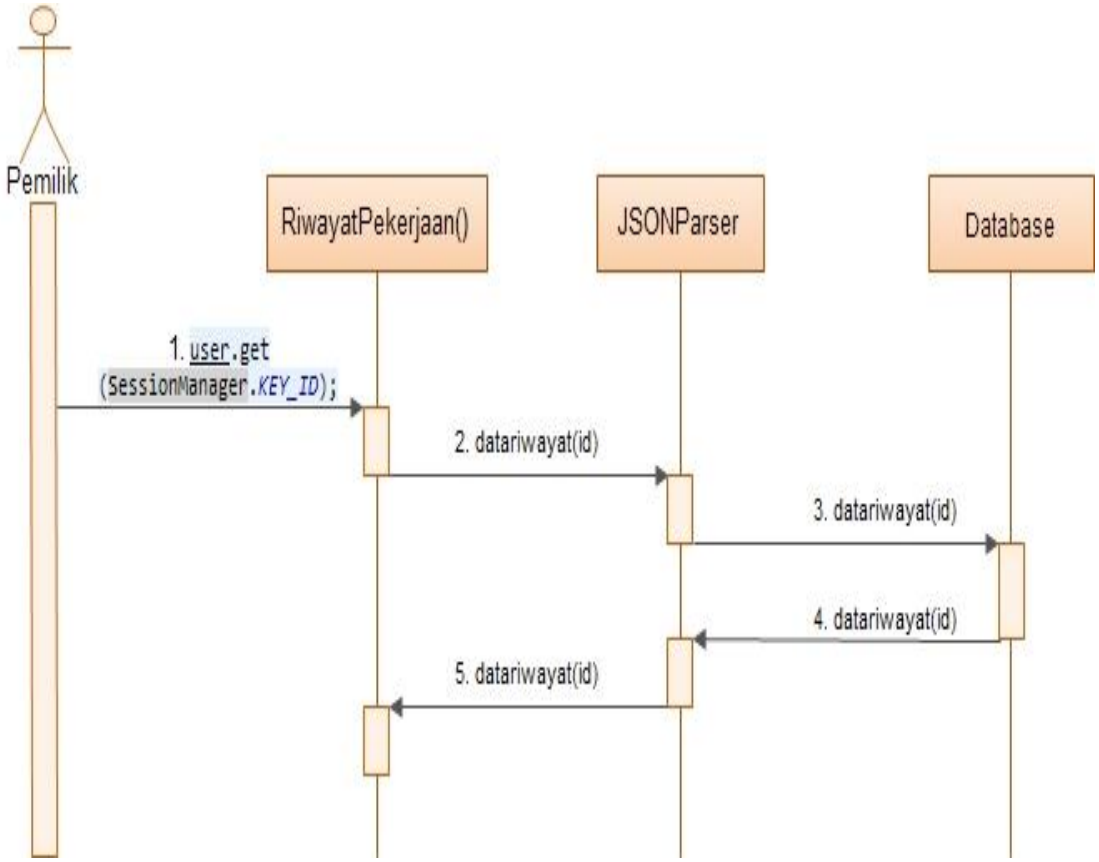
Sequence Diagram pada proses Profil akan dijelaskan pada gambar 3.29 berikut



Gambar 3. 29 *Sequence Diagram* Profil

g. *Sequence Diagram* Riwayat Pekerjaan

Sequence Diagram pada proses Riwayat Pekerjaan akan dijelaskan pada gambar 3.30 berikut

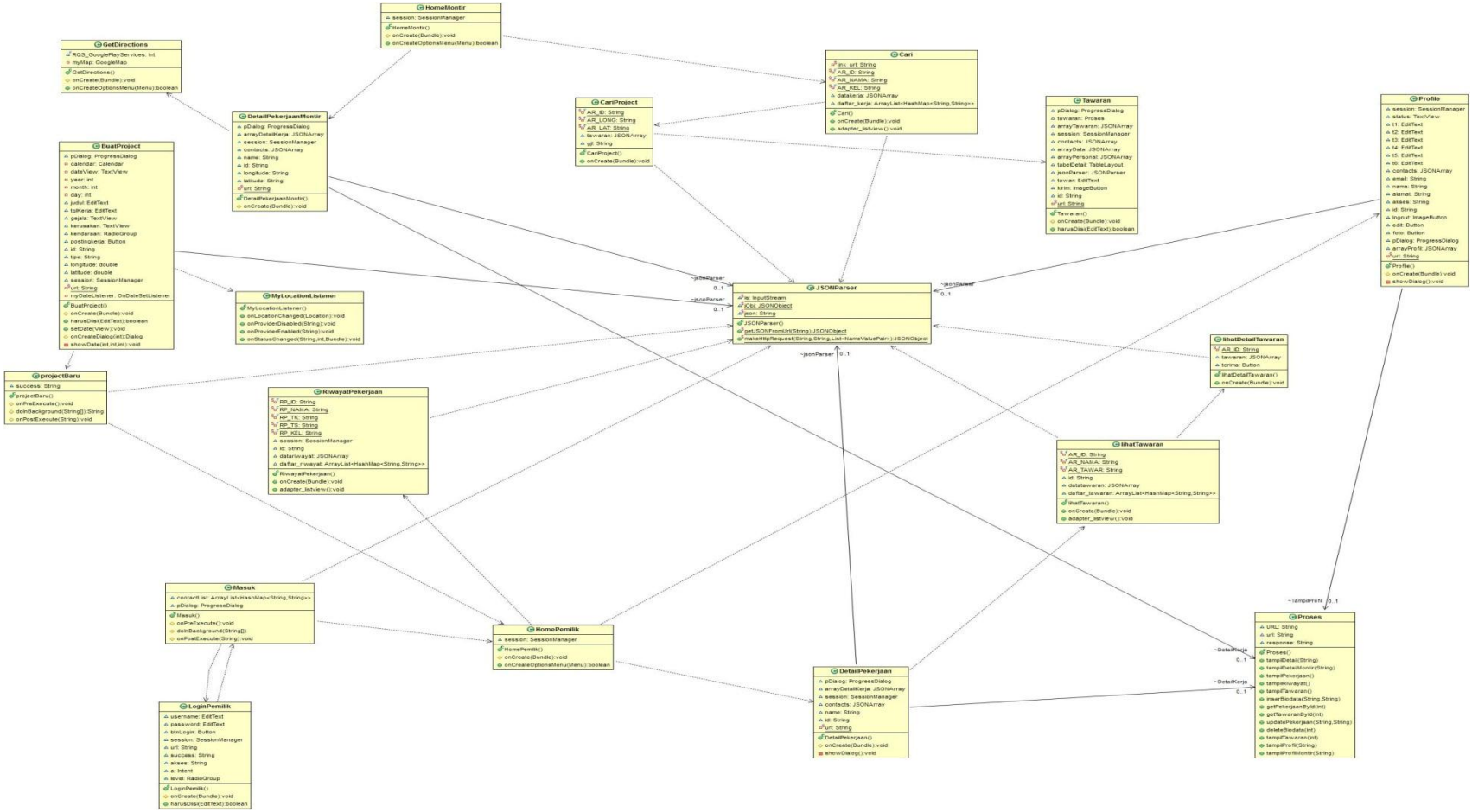


Gambar 3. 30 *Sequence Diagram* Riwayat Pekerjaan

3.11. Class Diagram

Class Diagram adalah sebuah spesifikasi dari fungsionalitas yang menghasilkan objek dan merupakan *Int*i dari pengembangan aplikasi ini. *Diagram* ini juga merupakan gambaran keadaan atribut atau properti dari sistem yang melakukan manipulasi atau metode.

Class Diagram pada aplikasi *frontend* akan dijelaskan pada gambar 3.31 berikut



Gambar 3. 31 *Class Diagram*

Tabel 3. 34 Deskripsi Class Diagram Mobile

Class	Jenis *Class*	Deskripsi
LoginPemilik	*Control*	*Class* yang berisikan *method* untuk validasi *username* dan *password* pangguna
Masuk	*Control*	*Class* yang berisikan *method* untuk masuk ke halaman utama pengguna
HomePemilik	*Control*	*Class* utama bagi pengguna pemilik kendaraan yang berhubungan dengan *Class* lain
HomeMontir	*Control*	*Class* utama bagi pengguna montir yang berhubungan dengan *Class* lain
GetDirections	*Control*	*Class* yang berisikan *method* untuk memanggil maps API dan petunjuk jalan.
BuatProject	*Control*	*Class* yang berisikan *method* untuk mengambil inputan data buat *project* baru
ProjectBaru	*Control*	*Class* yang berisikan *method* untuk mengirim data inputan kedalam *database*
DetailPekerjaan	*Control*	*Class* yang berisikan *method* untuk menampilkan detail pekerjaan yang sedang dikerjakan pemilik kendaraan.
LihatTawaran	*View*	*Class* yang berisikan *method* untuk menampilkan tawaran para montir
Profil	*View*	*Class* yang berisikan *method* untuk menampilkan profil pengguna
RiwayatPekerjaan	*View*	*Class* yang berisikan *method* untuk menampilkan riwayat pekerjaan yang pernah dilakukan pengguna
DetailPekerjaanMontir	*Control*	*Class* yang berisikan *method* untuk menampilkan detail pekerjaan yang sedang dikerjakan montir.
Cari	*Control*	*Class* yang berisikan *method* untuk menampilkan pekerjaan yang tersedia.
CariProject	*Control*	*Class* yang berisikan *method* untuk menampilkan pekerjaan yang tersedia.
Tawaran	*Control*	*Class* yang berisikan *method* untuk montir melakukan penawaran terhadap suatu project
JSONParser	*Control*	*Class* yang berisikan *method* untuk melakukan format pertukaran data
LIhatDetailTawaran	*Control*	*Class* yang berisikan *method* untuk menampilkan detail tawaran montir
Proses	*Control*	*Class* yang berisikan *method* untuk melakukan pengambilan data pada *server*
MyLocationListener	*Control*	*Class* yang berisikan *method* untuk mendapatkan posisi *user* saat ini.

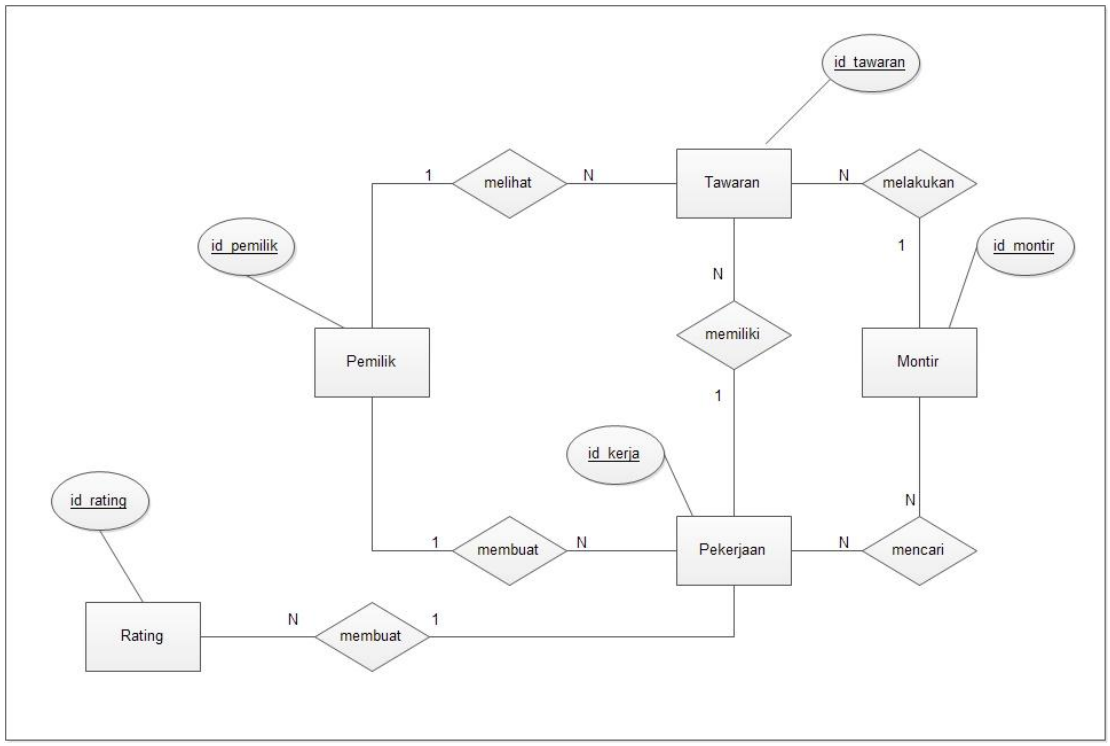
3.12. Analisis Basis Data

Analisis basis data dilakukan untuk mengetahui kebutuhan data pada sistem dan melakukan analisis terhadap hubungan antar data menggunakann *Entity*

Relationship Diagram. Dan mendefinisikan kamus data yang ada dalam setiap entitas yang berelasi tersebut.

3.12.1. Entity Relationship Diagram

Diagram relasi entitas merupakan *diagram* pemodelan basis data yang dibuat dengan menggambarkan hubungan antar entitas yang memiliki atribut yang berbeda-beda. *Diagram* relasi entitas dari sistem yang dibangun dapat dilihat pada Gambar 3.



Gambar 3. 32 *ERD*

3.12.2. Kamus Data

Kamus data ERD menjelaskan atribut apa saja yang terdapat pada setiap entitas.

Montir : {id_montir, nama, nama_bengkel, alamat, *username, password, Email, Phone, level*, status, kode}

Pemilik : {id_pemilik, nama, alamat, *username*, *password*, *Email*, *Phone*, *level*, status, kode}

Pekerjaan : {id_kerja, judul, *Owner*, *worker*, tglKerja, tglSelesai, kerusakan, gejala, status, *Longitude*, *latitude*}

Rating : {id_*rating*, *rating*, komentar, id_kerja}

Tawaran : {id_tawaran, id_kerja, tawaran}

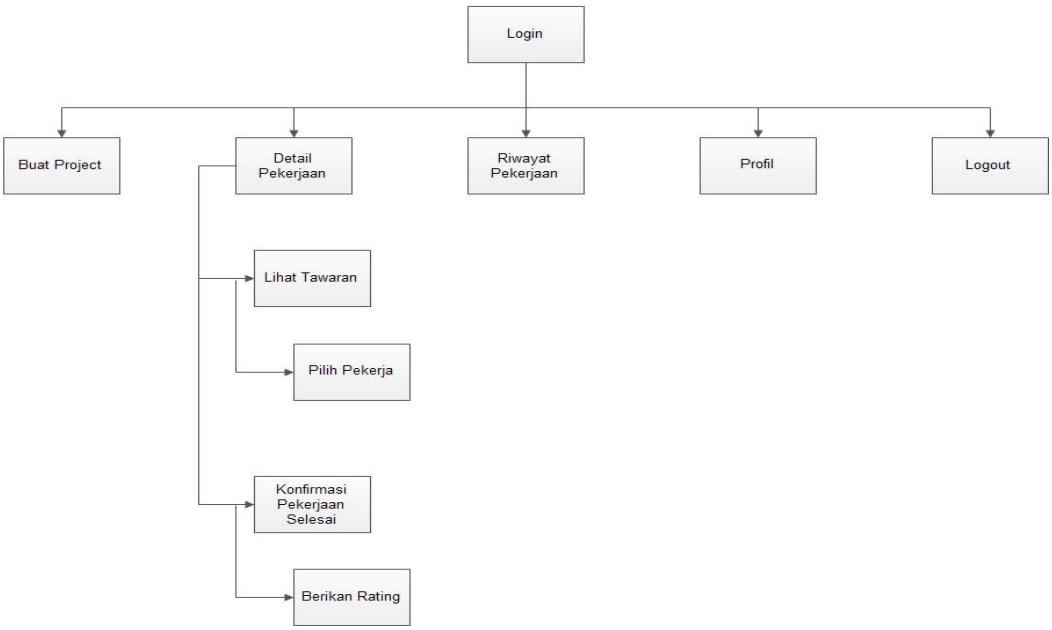
3.13. Perancangan Sistem

Perancangan sistem bertujuan untuk menspesifikasikan aspek-aspek teknik yang menjadi solusi dalam perencanaan. Pada tahap ini perancangan akan didefinisikan secara detail untuk mengatasi masalah-masalah yang lebih teknis, berkaitan dengan kegiatan implementasi seperti perancangan *database*, dan perancangan antarmuka.

3.13.1. Perancangan Struktur Menu

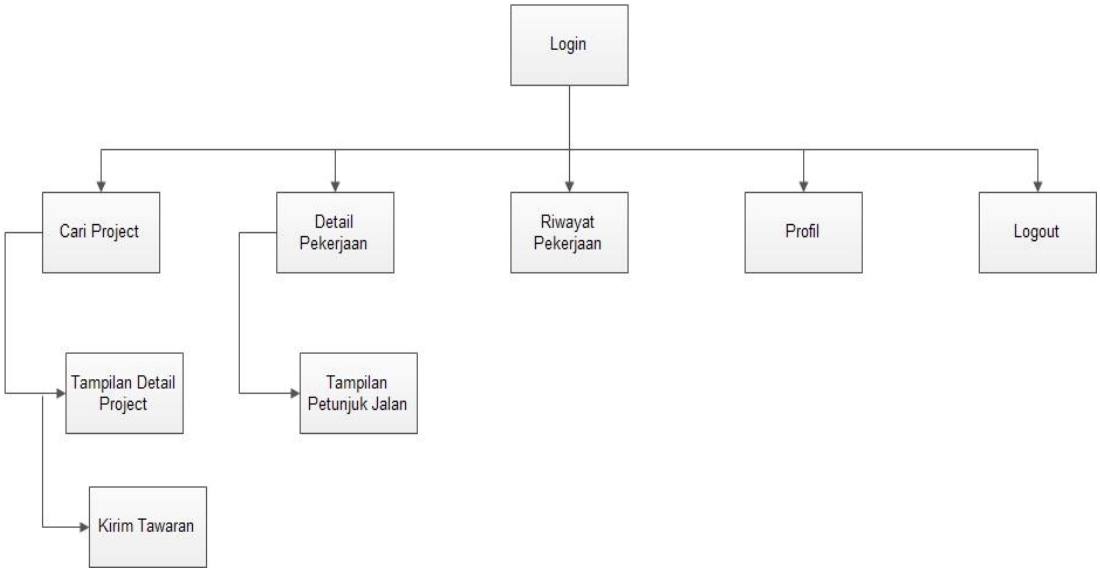
Perancangan arsitektur menyediakan gambaran dari stuktur menu program dari perangkat lunak yang akan dibuat. Berikut adalah gambaran perancangan arsitektural menu pada Aplikasi Montir *Freelancer* pada bidang otomotif pada *platform android*:

1. Perancangan menu aplikasi *android* pemilik kendaraan



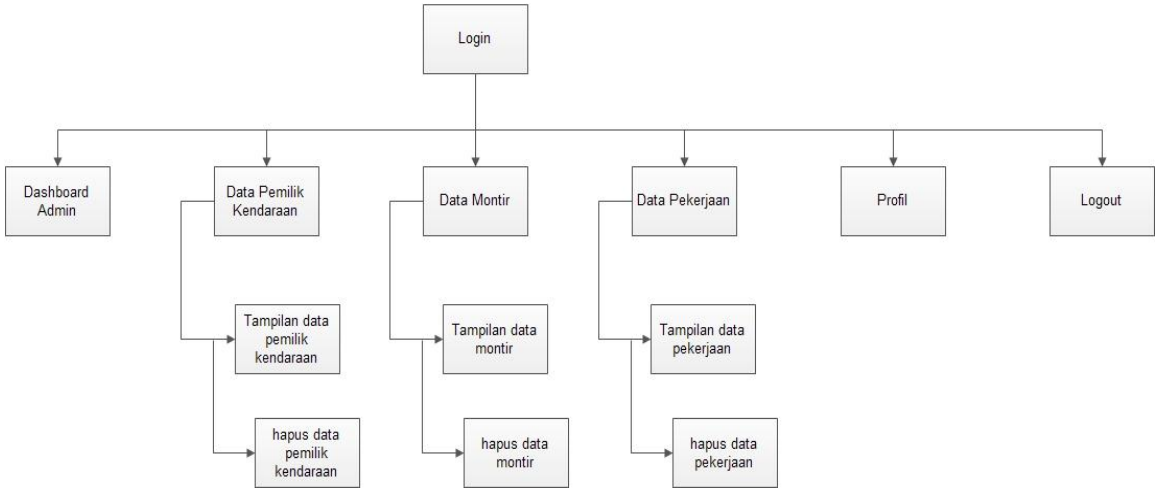
Gambar 3. 33 Struktur Menu pemilik kendaraan

2. Perancangan menu aplikasi *android* montir



Gambar 3. 34 Struktur Menu Montir

3. Perancangan menu aplikasi web admin

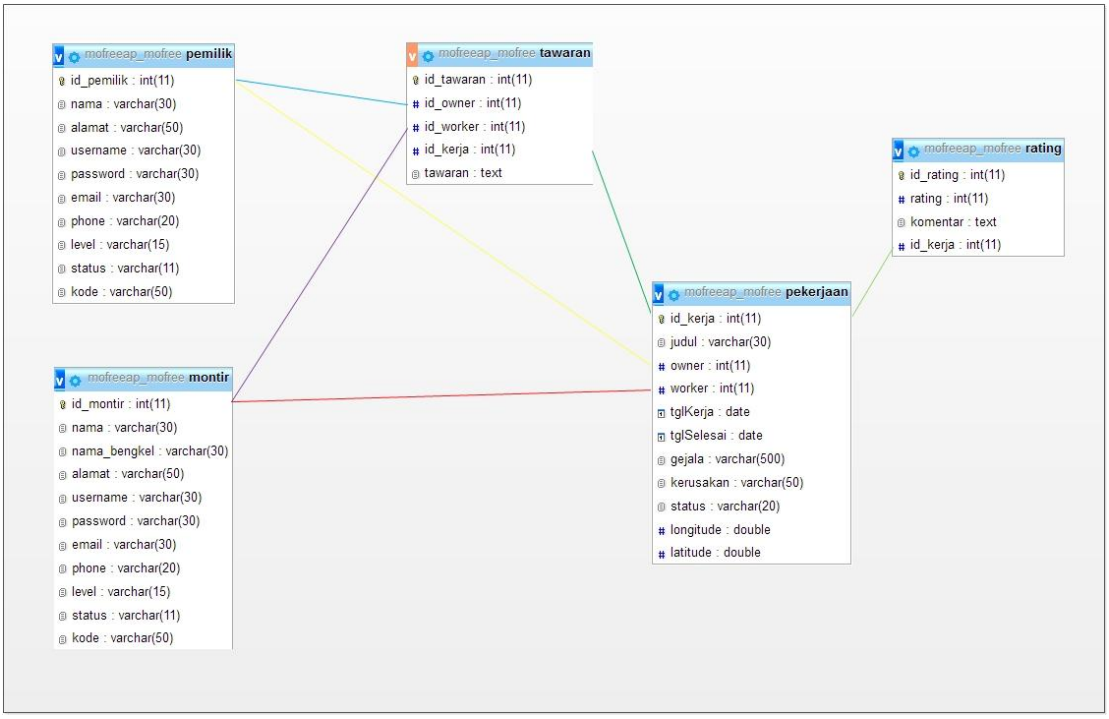


Gambar 3. 35 Struktur Menu Admin

3.13.2. Skema Relasi

Relasi antar tabel merupakan gabungan antar file yang mempunyai kunci utama yang sama, sehingga file-file tersebut menjadi satu kesatuan yang dihubungkan oleh *field* kunci (*Primary Key*). Pada proses ini elemen-elemen data

dikelompokkan menjadi satu file *database* beserta entitas dan hubungannya. Skema relasi ini dapat dilihat pada gambar 3.36



Gambar 3. 36 Skema Relasi

3.13.3. Struktur Tabel

Perancangan struktur tabel adalah perancangan tabel-tabel yang akan digunakan pada *database.* Tabel-tabel yang terdapat dalam *database* yang digunakan dalam aplikasi *Mobile android* ini adalah **:**

Tabel 3. 35 Struktur Tabel Pemilik Kendaraan

No	Nama *Field*	Tipe data	Panjang	Keterangan
1	Id_pemilik	*Integer*	11	PK
2	Nama	*Varchar*	30	
3	Alamat	*Varchar*	50	
4	*Username*	*Varchar*	30	
5	*Password*	*Varchar*	30	
6	*Email*	*Varchar*	30	
7	*Phone*	*Varchar*	20	
8	Level	*Varchar*	15	
9	Status	*Varchar*	11	
10	kode	*Varchar*	50	

Tabel 3. 36 Struktur Tabel Montir

No	Nama *Field*	Tipe data	Panjang	Keterangan
1	Id_montir	*Integer*	11	PK
2	Nama	*Varchar*	30	
3	Nama_bengkel	*Varchar*	30	
4	Alamat	*Varchar*	50	
5	*Username*	*Varchar*	30	
6	*Password*	*Varchar*	30	
7	*Email*	*Varchar*	30	
8	*Phone*	*Varchar*	20	
8	Level	*Varchar*	15	
9	Status	*Varchar*	11	
10	kode	*Varchar*	50	

Tabel 3. 37 Struktur Tabel Pekerjaan

No	Nama *Field*	Tipe data	Panjang	Keterangan
1	Id	*Integer*	11	PK
2	Judul	*Varchar*	30	
3	*Owner*	*Integer*	11	FK
4	*worker*	*Integer*	11	FK
5	tglKerja	date		
6	tglSelesai	date		
7	Gejala	*Varchar*	500	
8	kerusakan	*Varchar*	50	
9	status	*Varchar*	20	
10	*Longitude*	*Double*		
11	*latitude*	*Double*		

Tabel 3. 38 Struktur Tabel Tawaran

No	Nama *Field*	Tipe data	Panjang	Keterangan
1	Id_tawaran	*Integer*	11	PK
2	Id_*Owner*	*Integer*	11	FK
3	Id_*worker*	*Integer*	11	FK
4	Id_kerja	*Integer*	11	FK
5	tawaran	*text*		

Tabel 3. 39 Struktur Tabel Rating

No	Nama *Field*	Tipe data	Panjang	Keterangan
1	Id	*Integer*	11	PK
2	*Rating*	*Integer*	11	
3	komentar	*Text*		
4	Id_kerja	*Integer*	11	FK

3.13.4. Perancangan Antarmuka

Perancangan antarmuka merupakan penggambaran tampilan (*Interface*) sebuah sistem yang akan digunakan secara langsung oleh pengguna, dan juga merupakan penggambaran Interaksi yang dapat dilakukan oleh pengguna dalam sistem. Adapun dalam perancangan antarmuka ini terdapat beberapa bagian yang harus dilakukan, yaitu dari melakukan rancangan *input* dan *output* pada setiap fungsi yang telah ditentukan.

1. Perancangan antarmuka frontend

a. Perancangan antarmuka *login*

Berikut ini adalah antarmuka pada saat *login* yang dapat dilihat pada tabel 3.40 berikut ini.

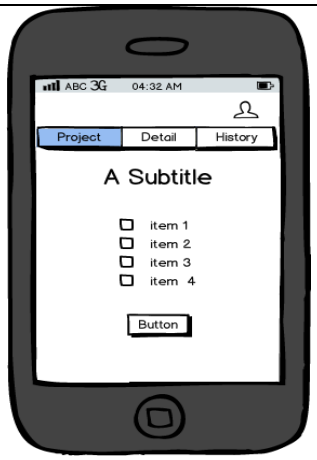
Tabel 3. 40 antarmuka login

<table>
<tr><td colspan="2">No : T001 (Login)</td></tr>
<tr><td>
</td><td>- Klik login maka akan menuju T002/T004
- Klik daftar maka akan menuju T009</td></tr>
<tr><td colspan="2">Warna background : putih</td></tr>
</table>

b. Perancangan antarmuka buat *project*

Berikut ini adalah antarmuka pada buat *project* yang dapat dilihat pada tabel 3.41 berikut ini.

Tabel 3. 41 antarmuka buat project

No : T002 (buat project)	
	- Klik Detail maka akan menuju T005 - Klik *History* maka akan menuju T006 - Klik Submit maka akan menuju T003 - Klik  maka akan menuju T008
Warna background : putih	

c. Perancangan antarmuka buat project

Berikut ini adalah antarmuka pada buat *project* yang dapat dilihat pada tabel 3.42 berikut ini.

Tabel 3. 42 antarmuka buat project

No : T003 (buat project)	
	- Klik posting maka sistem akan memproses dan kembali menuju T002
Warna background : putih	

d. Perancangan antarmuka cari *project*

Berikut ini adalah antarmuka pada cari *project* yang dapat dilihat pada tabel 3.43 berikut ini.

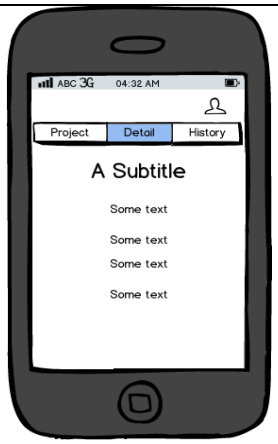
Tabel 3. 43 antarmuka cari project

<table>
<tr><td colspan="2">No : T004 (Cari Project)</td></tr>
<tr><td>
</td><td>- Klik History maka akan menuju T007
- Klik  maka akan menuju T008</td></tr>
<tr><td colspan="2">Warna background : putih</td></tr>
</table>

e. Perancangan antarmuka detail pekerjaan

Berikut ini adalah antarmuka pada detail pekerjaan yang dapat dilihat pada tabel 3.44 berikut ini.

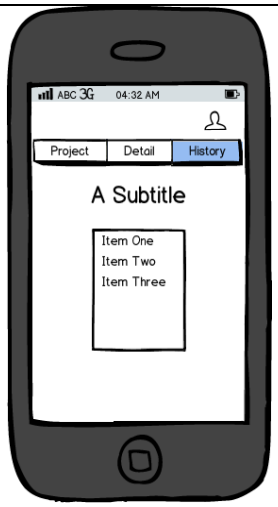
Tabel 3. 44 antarmuka detail pekerjaan

No : T005 (detail pekerjaan)	
	- Klik *Project* maka akan menuju T002 - Klik *History* maka akan menuju T006 - Klik  maka akan menuju T008
Warna background : putih	

f. Perancangan antarmuka riwayat pekerjaan Pemilik

Berikut ini adalah antarmuka pada menu riwayat pekerjaan yang dapat dilihat pada tabel 3.45 berikut ini.

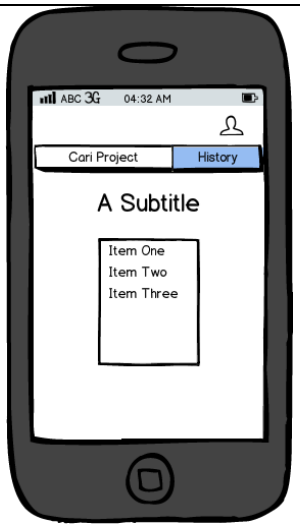
Tabel 3. 45 antarmuka login

No : T006 (riwayat pekerjaan)	
	- Klik *Project* maka akan menuju T002 - Klik Detail maka akan menuju T005 - Klik  maka akan menuju T008
Warna background : putih	

g. Perancangan antarmuka riwayat pekerjaan Montir

Berikut ini adalah antarmuka pada menu riwayat pekerjaan yang dapat dilihat pada tabel 3.46 berikut ini.

Tabel 3. 46 antarmuka riwayat pekerjaan Montir

<table>
<tr><td colspan="2">No : T007 (riwayat pekerjaan)</td></tr>
<tr><td>
</td><td>- Klik Cari Project maka akan menuju T004
- Klik  maka akan menuju T008</td></tr>
<tr><td colspan="2">Warna background : putih</td></tr>
</table>

h. Perancangan antarmuka melihat profil

Berikut ini adalah antarmuka pada menu lihat profil yang dapat dilihat pada tabel 3.47 berikut ini.

Tabel 3. 47 antarmuka melihat profil

No : T008 (profil)	
	-Klik maka akan menuju T001
Warna background : putih	

i. Perancangan antarmuka pendaftaran

Berikut ini adalah antarmuka pada menu pendaftaran yang dapat dilihat pada tabel 3.48 berikut ini.

Tabel 3. 48 antarmuka pendaftaran

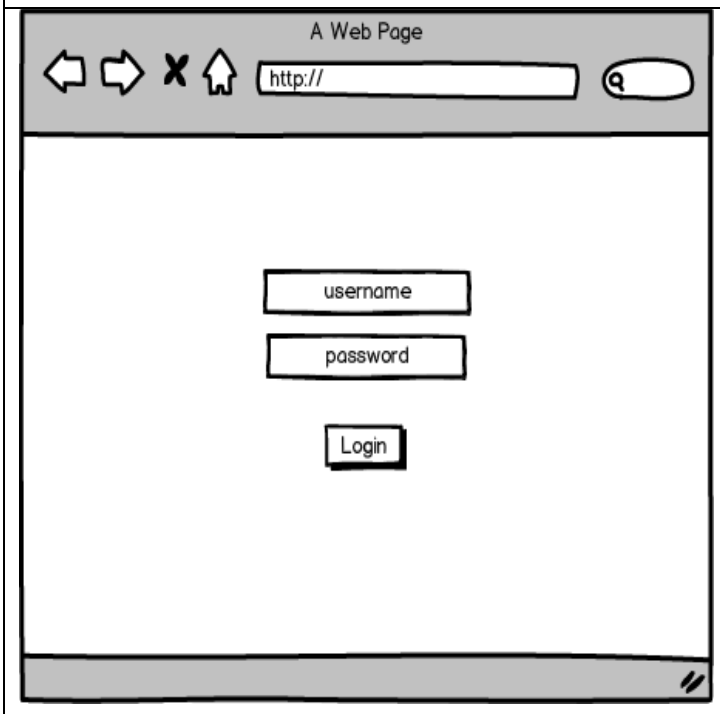
No : T009 (pendaftaran)	
	-Klik daftar maka akan menuju T001 - Klik batal maka akan menuju T001
Warna background : putih	

2. Perancangan antarmuka *backend*

a. Perancangan Antarmuka *Login* Admin

Berikut ini adalah antarmuka pada menu *login* admin yang dapat dilihat pada tabel 3.49 berikut ini.

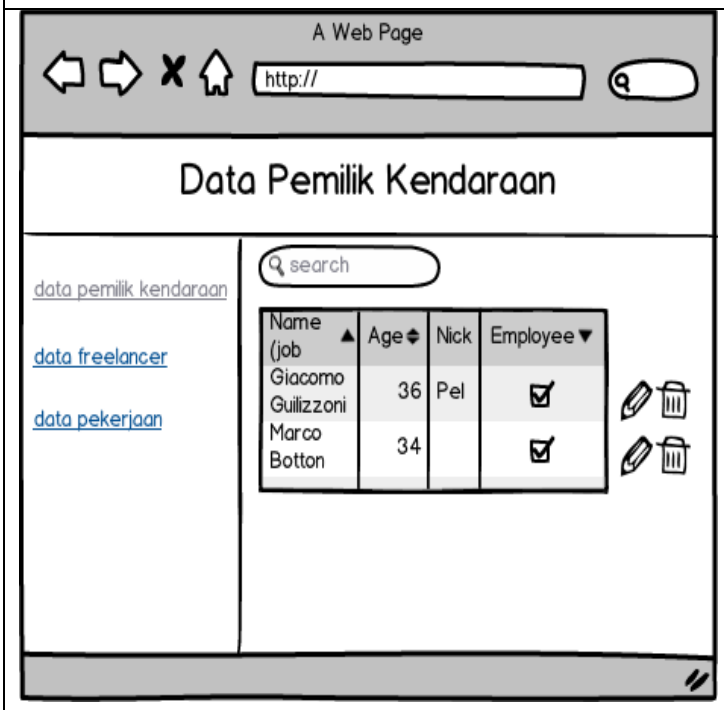
Tabel 3. 49 antarmuka Login Admin

<table>
<tr><td colspan="2">No : B001 (Login)</td></tr>
<tr><td>
</td><td>-Klik login maka akan menuju B002</td></tr>
<tr><td colspan="2">Warna background : putih</td></tr>
</table>

b. Perancangan antarmuka Olah data pemilik kendaraan

Berikut ini adalah antarmuka pada menu olah data pemilik kendaraan yang dapat dilihat pada tabel 3.50 berikut ini.

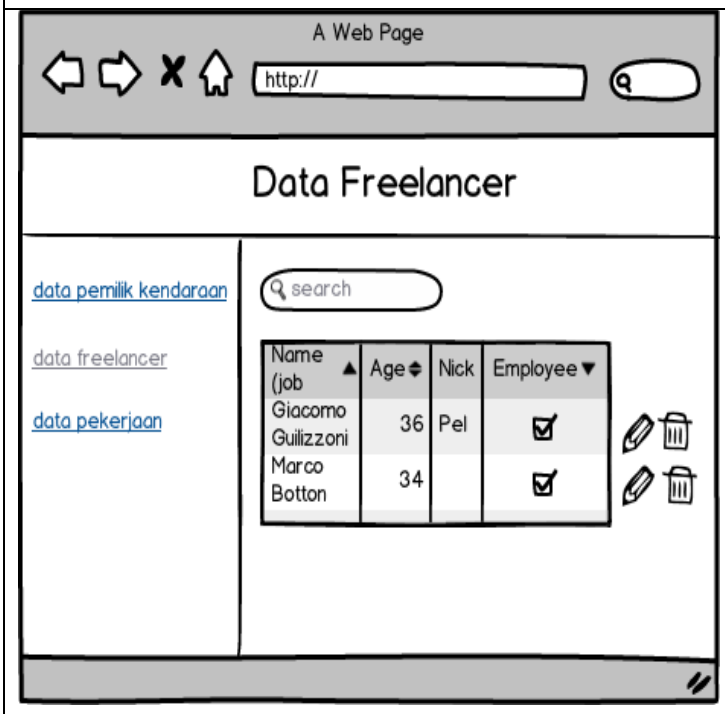
Tabel 3. 50 antarmuka Olah data pemilik kendaraan

<table>
<tr><td colspan="2">No : B002 (olah data pemilik kendaraan)</td></tr>
<tr><td>
</td><td>-Klik data freelancer maka akan menuju B003
- Klik data pekerjaan maka akan menuju B004</td></tr>
<tr><td colspan="2">Warna background : putih</td></tr>
</table>

c. Perancangan antarmuka Olah data *freelancer*

Berikut ini adalah antarmuka pada menu olah data *freelancer* yang dapat dilihat pada tabel 3.51 berikut ini.

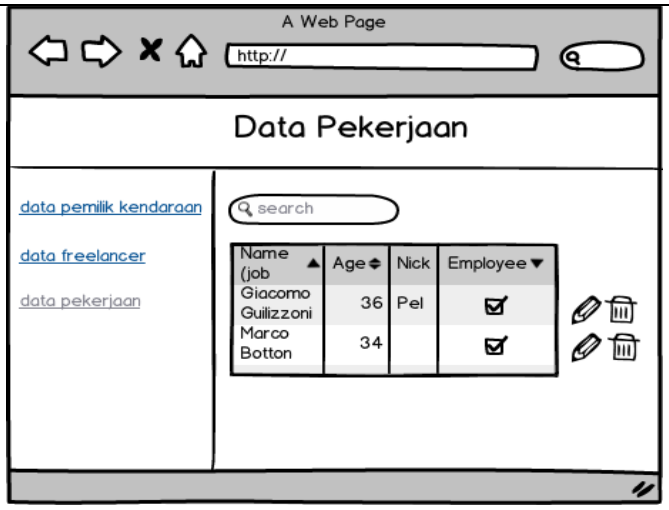
Tabel 3. 51 antarmuka Olah data freelancer

<table>
<tr><td colspan="2">No : B003 (olah data freelancer)</td></tr>
<tr><td>
</td><td>- Klik data pemilik kendaraan maka akan menuju B002
- Klik data pekerjaan maka akan menuju B004</td></tr>
<tr><td colspan="2">Warna background : putih</td></tr>
</table>

d. Perancangan antarmuka Olah data pekerjaan

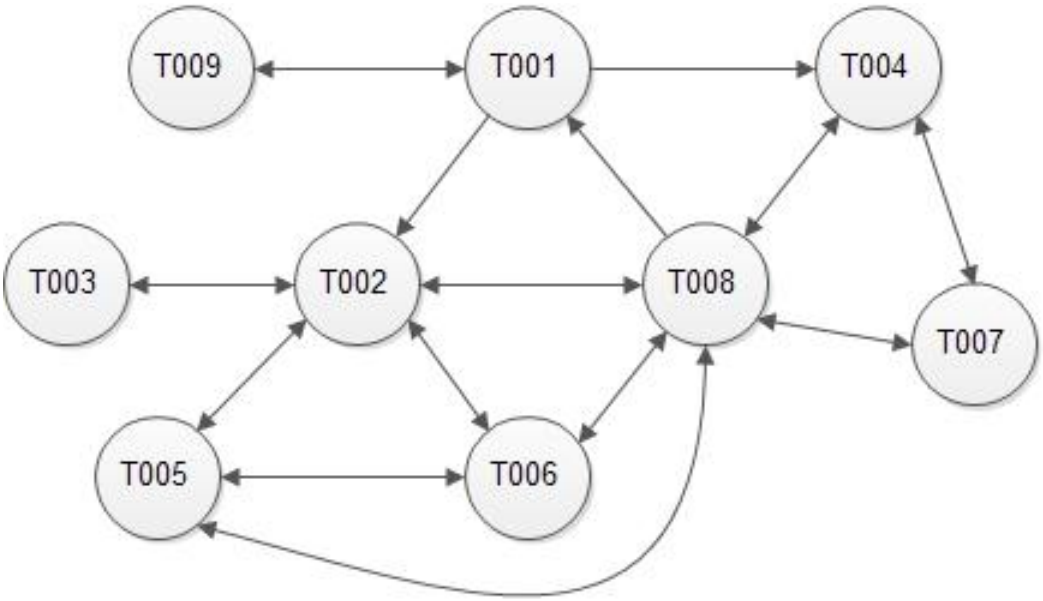
Berikut ini adalah antarmuka pada menu olah data pekerjaan yang dapat dilihat pada tabel 3.52 berikut ini.

Tabel 3. 52 antarmuka Olah data pekerjaan

<table>
<tr><td colspan="2">No : B004 (olah data pekerjaan)</td></tr>
<tr><td>
</td><td>-Klik data pemilik kendaraan maka akan menuju B002
- Klik data freelancer maka akan menuju T003</td></tr>
<tr><td colspan="2">Warna background : putih</td></tr>
</table>

3.13.5. Jaringan Semantik

Setelah melakukan perancangan antarmuka, maka dilakukanlah perancangan terhadap aliran dari menu-menu yang ada pada program. Untuk menggambarkan hubungannya maka digambarkan dalam sebuah jaringan semantik yang ada pada gambar 3.37.



Gambar 3. 37 Jaringan Semantik