

LAPORAN FINAL PROJECT

Analisis Desain Perangkat Lunak



BengkelQ APPS

“Ketika Bengkel Terasa Lebih Dekat”

ADPL-E

Kelompok 2:

- | | |
|----------------------------|------------|
| 1. Sholihatur Rahma A | 5215100025 |
| 2. Alkautsar | 5215100048 |
| 3. Wildan Azka F | 5215100077 |
| 4. Gilang Dimizza Mahadika | 5215100106 |
| 5. Azza Amalia | 5215100701 |

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Project BengkelQ

XML Legal Document Utility Software Design Document

Version <1.2>
Kelompok 2
2016-05-30

Revision History

Date	Version	Description	Author
10/05/17	<1.0>	Initial Version of Document	Kelompok 2
21/05/17	<1.1>	Full Version Document	Kelompok 2
30/05/17	<1.2>	Final Version of Document	Kelompok 2

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

DAFTAR ISI

1 Pendahuluan.....	4
1.1 Tujuan.....	4
1.2 Ruang Lingkup	5
1.3 Definisi, Akronim, and Abreviasi	5
1.4 References	6
1.5 Ikhtisar	6
2 Glossary	6
3 Use Cases.....	7
3.1 Aktor.....	7
3.2 List of Use Cases	7
3.3 Use Case Diagram	8
3.4 System Use-Case Descriptions	9
4 Robustness Diagram	19
5 Sequence Diagram	25
6 Class Diagram.....	35
7 GUI	37
8 Kebutuhan Non-Fungsional	46
9 Dokumen Tambahan	46

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Software Design Document

1 Pendahuluan

Dokumen ini merupakan portofolio untuk memberikan deskripsi desain system yang akan digunakan. SDD ini bertujuan untuk membantu pengembangan piranti lunak dengan memberikan rincian untuk bagaimana piranti lunak harus dibangun. Dalam SDD ini meliputi deskripsi tekstual dan grafis untuk proyek, termasuk *use case*, *sequence diagram*, *class diagram*, dan penunjang persyaratan informasi lainnya. Selain itu juga disajikan dokumentasi yang akan digunakan untuk membantu dalam proses development dari suatu software dengan menunjukkan detail bagaimana software tersebut dibuat.

Teknologi Informasi (TI), atau dalam bahasa Inggris dikenal dengan istilah Information Technology (IT) adalah istilah umum yang menjelaskan teknologi apa pun yang membantu manusia dalam membuat, mengubah, menyimpan, mengomunikasikan dan/atau menyebarkan informasi. Seiring berjalannya waktu, kemajuan IT di dunia sudah semakin canggih. Dalam setiap harinya pun masyarakat sekarang sudah terbiasa menggunakan IT dalam segala pekerjaan, sehingga hal ini menyebabkan masyarakat zaman sekarang memiliki ketergantungan terhadap teknologi kapanpun dan dimanapun.

Mencari bengkel dengan cepat dan tepat terkadang menjadi masalah pada masyarakat umum yang memiliki kendaraan roda dua maupun roda empat apalagi disaat yang mendadak seperti macet di perjalanan karena kerusakan mesin maupun ban bocor. Begitupun sebaliknya, terkadang pemilik bengkel juga merasa kesulitan dalam mencari pelanggan dikarenakan faktor tempat yang tidak strategis dan lain-lain. Oleh karena itu, kelompok kami ingin membuat suatu aplikasi yang memfasilitasi antara pengguna bengkel dan pemilik bengkel.

BengkelQ merupakan aplikasi berbasis android yang dirancang untuk mempermudah pengguna bengkel dalam mendapatkan lokasi bengkel motor maupun mobil terdekat dari keberadaan pengguna bengkel. Tidak hanya itu, pengguna juga dapat mencari bengkel yang sesuai dengan pencarian manual sesuai dengan keinginan. Aplikasi ini bertujuan untuk memfasilitasi transaksi antara pengguna bengkel dan pemilik bengkel dengan fitur yang utama yaitu ketepatan posisi geografis user di peta dengan memanfaatkan GPS pada perangkat android. Bengkel disini bukan hanya bengkel mesin namun bengkel accessoris, ban&velg, mesin, dan lain-lain.

1.1 Tujuan

Dokumen Perancangan Perangkat Lunak (DPPL) merupakan dokumen deskripsi dari perancangan perangkat lunak yang akan dikembangkan dan bertujuan untuk memberikan landasan yang diperlukan dalam proses code aplikasi BengkelQ. Dokumen ini digunakan oleh pengembang perangkat lunak sebagai acuan teknis pengembangan perangkat lunak pada tahap selanjutnya.

Sedangkan tujuan dari pembuatan Aplikasi ini sendiri adalah membangun sebuah aplikasi online untuk perawatan atau servis sepeda motor dengan cepat dan tepat. Sehingga akan memudahkan:

1. Transaksi antara pemilik bengkel dan pengguna bengkel
2. Pengguna Bengkel untuk mencari lokasi bengkel terdekat.
3. Pengguna Bengkel untuk memperbaiki kerusakan maupun perawatan terhadap kendaraan yang dimiliki.

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

1.2 Ruang Lingkup

Cakupan dari aplikasi BengkelQ adalah sebagai berikut:

1. Bengkel

Bengkel adalah sasaran pembuatan aplikasi dengan cakupan bahwa bengkel tersebut telah terdaftar pada sistem BengkelQ. Di mana dalam hal pengelolaan sistem untuk aplikasi akan dipegang oleh salah satu pegawai bengkel yang disebut admin. Admin adalah perpanjangan tangan dari bengkel yang menangani masalah Pengguna bengkel bersangkutan.

2. Pengguna Bengkel

Pengguna bengkel yang menggunakan aplikasi bengkelQ berasal dari seluruh lapisan masyarakat dengan syarat sudah memiliki akun. Bila belum memiliki akun, maka pengguna bengkel akan diarahkan untuk sign up/mendaftar terlebih dahulu.

3. Fungsi

Aplikasi ini berfokus untuk memfasilitasi transaksi antara pengguna bengkel dan pemilik bengkel agar menjadi lebih mudah, cepat dan tepat.

1.3 Definisi, Akronim, and Abreviasi

1. Objects

Objek pada java adalah sekumpulan software yang terdiri dari variable dan method-method yang terkait. Objek juga merupakan benda nyata yang di buat berdasarkan rancangan yang di definisikan di dalam class

Object adalah instance dari class. Jika class secara umum merepresentasikan (template) sebuah object, sebuah instance adalah representasi nyata dari class itu sendiri. Contoh: Dari class Fruit kita dapat membuat object Mangga, Pisang, Apel, dan lain-lain.

2. Sequence Diagram

Sequence diagram adalah suatu diagram yang menggambarkan interaksi antar obyek dan mengindikasikan komunikasi diantara obyek-obyek tersebut. Diagram ini juga menunjukkan serangkaian pesan yang dipertukarkan oleh obyek-obyek yang melakukan suatu tugas atau aksi tertentu. Obyek-obyek tersebut kemudian diurutkan dari kiri ke kanan, aktor yang menginisiasi interaksi biasanya ditaruh di paling kiri dari diagram.

3. Class Diagram

Class Diagram adalah diagram UML yang menggambarkan kelas- kelas dalam sebuah sistem dan hubungannya antara satu dengan yang lain, serta dimasukkan pula atribut dan operasi. https://id.wikipedia.org/wiki/Diagram_kelas

4. Atribut

Atribut merupakan gambaran karakteristik dari sebuah entitas atau himpunan entitas. Contoh : atribut untuk himpunan entitas mahasiswa adalah nim, nama, alamat, ipk, program studi, hobi, dsb. Setiap atribut mempunyai domain value set yaitu batasan batasan yg dibolehkan bagi suatu atribut. <http://informatika.web.id/atribut.htm>

5. Class

Class adalah struktur dasar dari OOP (Object Oriented Programming). Terdiri dari dua tipe yaitu : field (attribute/property) dan method (behavior). Class digunakan untuk

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

mendeklarasikan sebuah variabel yang berupa objek atau dinamakan “referensi objek (object reference)”.

6. Generalisasi

Generalisasi adalah pendefinisian subclass-subclass yang disatukan menjadi entitas superclass tunggal berdasarkan karakteristik umum. Disamping proses desain top-down (dari inisial entitas ke level lebih rendah (subgrup)), desain juga dapat dilakukan dengan proses bottom-up, yaitu banyak entitas disintesis menjadi entity yang lebih tinggi berdasarkan kesamaan feature-nya.

7. Spesialisasi

Spesialisasi adalah proses yang memperhatikan perbedaan setiap anggota dari sebuah entitas sehingga anggota dari sebuah entitas bisa dikelompokkan ke dalam kelompok yang lebih kecil.

8. Method

Method adalah kumpulan program yang mempunyai nama. Method merupakan sarana bagi programmer untuk memecah program menjadi bagian- bagian yang kecil agar jadi lebih kompleks sehingga dapat di gunakan berulang- ulang.

Method merupakan suatu operasi berupa fungsi-fungsi yang dapat dikerjakan oleh suatu object. Method didefinisikan pada class akan tetapi dipanggil melalui object.

1.4 References

- Sholih. 2010. *Analisis dan Perancangan Berorientasi Obyek*, Muara Indah.
- Cecil Martin, Robert. 2002. *UML for Java Programmers*, New Jersey.

1.5 Ikhtisar

Dokumen Perancangan Perangkat Lunak dibagi menjadi 9 section dengan variasi sub-section yang akan menjelaskan detail terkait aplikasi BengkelQ. Berikut daftar sectionnya:

1. Pendahuluan
2. Glosarium
3. *Use Cases*
4. *Robustness Diagram*
5. *Sequence Diagram*
6. *Class Diagram*
7. GUI
8. Kebutuhan non-fungsional
9. Dokumentasi Tambahan

2 Glossary

- SUC: *Use case diagram*, merupakan diagram yang mendeskripsikan serangkaian interaksi antara pengguna dengan sistem. Sebuah use case diagram dapat menjelaskan manfaat dari suatu sistem jika dilihat menurut sudut pandang orang diluar system
- Selain itu, glossary dari dokumen ini telah dijelaskan pada sesi 1 di bagian ke 3

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

3 Use Cases

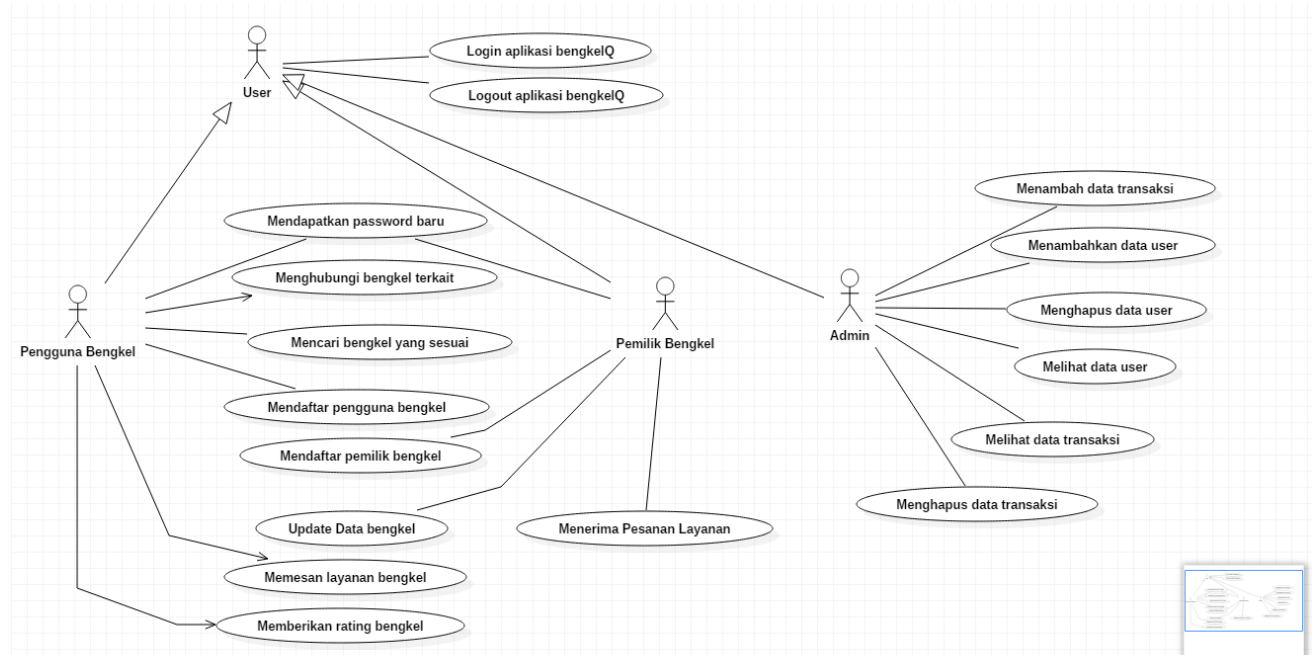
3.1 Aktor

1. Pengguna Bengkel
Merupakan pengguna aplikasi yang membutuhkan jasa layanan dari bengkel.
2. Pemilik Bengkel
Merupakan pengguna aplikasi yang memiliki bengkel dan mengelola akun bengkel.
3. Admin
Merupakan pengguna aplikasi yang bertugas untuk mengelola sistem BengkelQ.
4. User
Merupakan aktor generalisasi dari pengguna bengkel, pemilik bengkel dan admin yang dapat melakukan login dan logout.

3.2 List of Use Cases

1. Pengguna Bengkel
 - a. Mendaftar Pengguna Bengkel
 - b. Mencari bengkel yang sesuai
 - c. Menghubungi bengkel terkait
 - d. Memesan layanan bengkel
 - e. Memberi rating bengkel
 - f. Mendapatkan password baru
2. Pemilik Bengkel
 - a. Mendaftar Pemilik Bengkel
 - b. Mendapatkan password baru
 - c. Update data bengkel
 - d. Menerima pesanan layanan
3. Admin
 - a. Menambah data transaksi
 - b. Melihat data transaksi
 - c. Menghapus data transaksi
 - d. Menambah data user
 - e. Melihat data user
 - f. Menghapus data user
4. User
 - a. Login Aplikasi BengkelQ
 - b. Logout Aplikasi BengkelQ

3.3 Use Case Diagram

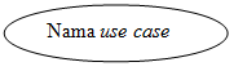
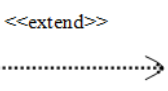
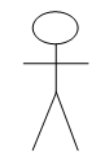
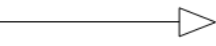

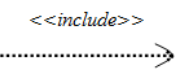


Gambar 3.9 System Use Case Aplikasi BengkelQ

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. yang menjadi persoalan itu apa yang dilakukan. Use case diagram mendefinisikan cakupan dari system sehingga dapat memvisualisasikan sistem yang akan dibuat.

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Berikut penjelasan atribut atribut Use case Diagram

Simbol	Deskripsi	Simbol	Deskripsi
<p><i>Use Case</i></p> 	Fungsional yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; biasanya dinyatakan dengan menggunakan kata kerja diawal frase nama <i>use case</i>	<p>Ekstensi / <i>extend</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan.
<p>Aktor / <i>Actor</i></p> 	Orang proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya menggunakan kata benda di awal frase nama aktor.	<p>Generalisasi / <i>generalization</i></p> 	Hubungan generalisasi dan spesialisasi (umum - khusus) antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang umum dari lainnya.
<p>Asosiasi / <i>association</i></p> 	Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.	<p>Menggunakan / <i>include</i> / <i>uses</i></p> 	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsinya atau sebagai syarat dijalankan <i>use case</i> ini.

3.4 System Use-Case Descriptions

Nama Use Case		Login aplikasi BengkelQ	
Aktor		User(Pengguna Bengkel, Pemilik Bengkel dan Admin)	
Deskripsi Singkat		Use case ini memungkinkan pemilik bengkel, pengguna bengkel dan admin untuk masuk ke dalam aplikasi dengan hak akses yang sesuai.	
Pra-syarat		Telah daftar	
Alur Utama	Aktor	System	
	1. Aktor membuka aplikasi	2. System menampilkan menu login	
	3. Aktor mengisi username dan password		
	4. Aktor menekan button login	5. System memvalidasi username & password	
	6. Use case berakhir		
Altrenatif		Jika username maupun password yang dimasukkan salah, maka akan muncul pop-up "Username dan Password yang anda masukkan salah, silahkan masukkan lagi" kemudian akan kembali ke menu Login	

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Kondisi akhir sukses	Login sukses dilakukan oleh user.
Kondisi akhir gagal	Login gagal dilakukan oleh user.

Nama Use Case		Logout aplikasi BengkelQ
Aktor	User(Pengguna Bengkel, Pemilik Bengkel dan Admin)	
Deskripsi Singkat	Use case ini memungkinkan user untuk keluar dari aplikasi BengkelQ	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor men-swap menu	2. System menampilkan menu swap
	3. Aktor memilih option Logout	4. System menampilkan pop-up “Apakah anda akan keluar dari aplikasi?” dan button yes & no
	5. Aktor memilih button yes	6. System masuk ke menu awal ketika akan login
	7. Use case berakhir	
Altrenatif	Apabila logout tidak berhasil karena akor sedang diproses transaksi maka akan muncul pop up “Anda sedang dalam proses transaksi, silahkan batalkan transaksi terlebih dahulu” kemudian akan kembali ke menu pemesanan.	
Kondisi akhir sukses	Aktor sukses keluar dari aplikasi	
Kondisi akhir gagal	Aktor gagal keluar dari aplikasi	

Nama Use Case		Mendaftar sebagai Pengguna Bengkel
Aktor	Pengguna Bengkel	
Deskripsi Singkat	Use case ini memungkinkan pengguna bengkel melakukan pendaftaran agar memiliki akses pada aplikasi BengkelQ.	
Pra-syarat	-	
Alur Utama	Aktor	System
	1. Aktor membuka aplikasi	2. System menampilkan menu awal
	3. Aktor memilih button Sign-Up	4. System menampilkan menu option pilihan daftar sebagai pengguna
	5. Aktor memilih daftar sebagai Pengguna Bengkel	6. System menampilkan menu isi identitas

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

	7. Aktor mengisi menu identitas dan menekan tombol Sign-Up	8. System akan memvalidasi proses pendaftaran
	9. Use case berakhir	
Altrenatif	Apabila pada menu isi identitas actor tidak lengkap dalam pengisian data diri, maka akan muncul pop-up "masukkan identitas dengan lengkap" kemudian kembali ke menu isi data Pengguna Bengkel	
Kondisi akhir sukses	Pendaftaran pengguna bengkel sukses dilakukan.	
Kondisi akhir gagal	Pendaftaran pengguna bengkel tidak dapat dilakukan.	

Nama Use Case	Mendaftar sebagai Pemilik Bengkel	
Aktor	Pemilik Bengkel	
Deskripsi Singkat	Use case ini memungkinkan pemilik bengkel melakukan pendaftaran agar memiliki akses pada aplikasi BengkelQ sebagai pemilik bengkel.	
Pra-syarat	-	
Alur Utama	Aktor	System
	1. Aktor membuka aplikasi	2. System menampilkan menu awal
	3. Aktor memilih button Sign-Up	4. System menampilkan menu option pilihan daftar sebagai pengguna
	5. Aktor memilih daftar sebagai Pemilik Bengkel	6. System menampilkan menu isi identitas diri & identitas bengkel
	7. Aktor mengisi menu identitas dan menekan tombol Sign-Up	8. System akan memvalidasi proses pendaftaran
	9. Use case berakhir	
	Altrenatif	jika identitas yang dimasukkan tidak lengkap maka akan muncul pop up "masukkan identitas dengan lengkap" kemudian kembali ke menu isi data Pemilik Bengkel
Kondisi akhir sukses	Pendaftaran pemiliki bengkel sukses dilakukan.	
Kondisi akhir gagal	Pendaftaran pemilik bengkel tidak dapat dilakukan.	

Nama Use Case		Mencari bengkel yang sesuai
Aktor	Pengguna Bengkel	

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Deskripsi Singkat	Use case ini memungkinkan pengguna bengkel dapat mencari bengkel yang sesuai dengan kendaraan yang dimiliki dan layanan yang dikehendaki .	
Pra-syarat	Login	
Alur Utama	Aktor	System
		1. System menampilkan menu jenis kendaraan
	2. Aktor memilih button jenis kendaraan	3. System menampilkan menu layanan bengkel
	4. Aktor memilih jenis layanan yang akan digunakan	5. System menampilkan pilihan filter pencarian yaitu sesuai dengan lokasi terdekat atau nama bengkel
	6. Aktor memilih dari filter pencarian bengkel tersebut	7. System akan mencari bengkel sesuai dengan yang dicari aktor.
	8. Use case berakhir	
Altrenatif	Jika bengkel di tempat terdekat anda tidak ada, maka akan muncul pop-up "Maaf, bengkel di tempat terdekat anda sekarang tidak tersedia. Silahkan masukkan tempat terdekat yang lain." Kemudian akan kembali ke menu pencarian bengkel.	
Kondisi akhir sukses	Pengguna Bengkel dapat dapat mencari & memilih bengkel yang sesuai.	
Kondisi akhir gagal	Pengguna Bengkel tidak menemukan bengkel yang sesuai.	

Nama Use Case		Menghubungi bengkel terkait
Aktor	Pengguna Bengkel	
Deskripsi Singkat	Use case ini memungkinkan pengguna bengkel menghubungi pemilik bengkel untuk melakukan fiksasi mengenai pelayanan bengkel yang akan digunakan pengguna bengkel.	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor memilih bengkel yang sesuai	2. System menampilkan profile bengkel dan contact person nya
	3. Aktor menghubungi pemilik bengkel via telepon	4. Use case berakhir
Altrenatif	Jika bengkel yang dipilih tutup, maka akan muncul pop-up berupa "Bengkel yang anda pilih tutup, silahkan pilih bengkel yang lain" kemudian akan kembali ke menu pilihan bengkel.	

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Kondisi akhir sukses	Pengguna Bengkel dapat menghubungi pemilik bengkel dan mendapatkan fiksasi pelayanan bengkel.
Kondisi akhir gagal	Pengguna Bengkel tidak dapat menghubungi pemilik bengkel.

Nama Use Case		Memesan layanan bengkel
Aktor	Pengguna Bengkel	
Deskripsi Singkat	Use case ini memungkinkan Pengguna Bengkel memesan layanan bengkel setelah melakukan fiksasi dengan pemilik bengkel.	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor memilih button pesan layanan pada menu profile bengkel	2. System akan menampilkan form yang isinya kolom no booking & jenis layanan yang dipilih
	3. Aktor mengisi form dan menekan button “Pesan”	4. Use case berakhir
Altrenatif	Jika Pengguna Bengkel tidak jadi melanjutkan ke tahap pemesanan, maka akan muncul popup “Apakah anda yakin tidak melanjutkan ke menu pemesanan?” kemudian terdapat button “Yes” dan “No”. Apabila Pengguna Bengkel menekan button Yes maka akan kembali ke menu pilihan bengkel	
Kondisi akhir sukses	Pemesanan Layanan bengkel oleh Pengguna Bengkel sukses	
Kondisi akhir gagal	Pemesanan Layanan bengkel oleh Pengguna Bengkel gagal	

Nama Use Case		Memberikan rating bengkel
Aktor	Pengguna Bengkel	
Deskripsi Singkat	Use case ini memungkinkan Pengguna Bengkel untuk memberikan rating pada bengkel yang telah digunakan.	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor sudah selesai menerima layanan dari bengkel	2. System menampilkan pop-up berupa pemberian rating untuk bengkel
	3. Aktor memilih skala bintang 1-5 untuk bengkel	4. Use case berakhir

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Altrenatif	Apabila aktor tidak ingin memberikan rating dengan car menekan tombol back, makaa akan muncul pop-up “Apakah anda tidak ingin memberikan rating untuk bengkel?” disertai button yes & no, maka aktor pilihlah button yes.
Kondisi akhir sukses	Aktor berhasil memberikan rating untuk bengkel
Kondisi akhir gagal	Aktor tidak memberikan rating untuk bengkel

Nama Use Case		Update data bengkel	
Aktor	Pemilik Bengkel		
Deskripsi Singkat	Use case ini memungkinkan pemilik bengkel melakukan update data bengkel yang dimiliki berupa update layanan, tempat maupun waktu buka.		
Pra-syarat	Login		
Alur Utama	Aktor	System	
	1. Aktor memilih button update data	2. System akan menampilkan menu isi identitas diri & bengkel	
	3. Aktor meng-update data yang dibutuhkan & menekan button OK	4. Use case berakhir	
Altrenatif	jika identitas yang dimasukkan tidak lengkap maka akan muncul pop up "masukkan identitas dengan lengkap" kemudian kembali ke menu isi data Pemilik Bengkel		
Kondisi akhir sukses	Pendaftaran pemiliki bengkel sukses dilakukan.		
Kondisi akhir gagal	Pendaftaran pemilik bengkel tidak dapat dilakukan.		

Nama Use Case		Menerima Pesanan Layanan	
Aktor	Pemilik Bengkel		
Deskripsi Singkat	Use case ini memungkinkan pemilik bengkel untuk menerima layanan yang dipesan oleh pengguna bengkel		
Pra-syarat	Login		
Alur Utama	Aktor	System	
	1. Aktor menekan ikon notifikasi	2. Menampilkan notifikasi yang masuk	
	3. Actor memilih button terima		
		4. Use case selesai	

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Altrenatif	jika pemilik bengkel tidak menerima pesanan layanan, maka akan muncul pop-up berupa “Apakah anda yakin untuk menolak pesanan” kemudian terdapat button yes dan no. ketika button yes di klik maka akan kembali ke menu notifikasi
Kondisi akhir sukses	Pemilik Bengkel menerima pesanan layanan
Kondisi akhir gagal	Pemilik Bengkel menolak pesanan layanan

Nama Use Case	Menambah data user	
Aktor	Admin	
Deskripsi Singkat	Use case ini memungkinkan admin untuk menambah data user dari aplikasi BengkelQ	
Pra-syarat	Login	
Alur Utama	Aktor	System
	5. Aktor memilih button “Tambah user”	6. System akan menampilkan menu untuk menambahkan user
	7. Aktor mengisi identitas user yang akan ditambahkan	
	8. Aktor menekan button “Tambah”	9. Use case berakhir
Altrenatif	jika identitas yang dimasukkan tidak lengkap maka akan muncul pop up "masukkan identitas dengan lengkap" kemudian kembali ke menu menambahkan user	
Kondisi akhir sukses	Admin berhasil menambahkan user untuk aplikasi BengkelQ	
Kondisi akhir gagal	Admin tidak dapat menambahkan user untuk aplikasi BengkelQ	

Nama Use Case	Menghapus data user	
Aktor	Admin	
Deskripsi Singkat	Use case ini memungkinkan admin untuk menghapus data user dari aplikasi BengkelQ	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor memilih button hapus data user	2. System menampilkan menu hapus data user dengan kolom email user yang akan dihapus

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

	3. Aktor menulis email user pada kolom & menekan button "Hapus"	4. Use case berakhir
Altrenatif	Apabila email yang dimasukkan tidak terdapat pada data user, maka akan muncul pop-up "Email yang anda masukkan tidak tersedia, silahkan masukkan kembali" kemudian system akan menampilkan menu hapus data user	
Kondisi akhir sukses	Admin dapat menghapus data user	
Kondisi akhir gagal	Admin tidak dapat menghapus data user	

Nama Use Case		Melihat data user
Aktor	Admin	
Deskripsi Singkat	Use case ini memungkinkan admin dapat melihat data-data user yang terdaftar di aplikasi BengkelQ	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Admin menekan button "Lihat data user"	2. System menampilkan menu list data user
	3. Use case berakhir	
Altrenatif	-	
Kondisi akhir sukses	Admin dapat melihat data-data user	
Kondisi akhir gagal	Admin tidak dapat melihat data-data user	

Nama Use Case		Menghapus data transaksi
Aktor	Admin	
Deskripsi Singkat	Use case ini memungkinkan admin menghapus data transaksi yang sekiranya perlu untuk dihapus	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor menekan button hapus data transaksi	2. System menampilkan list data transaksi
	3. Aktor memilih data transaksi yang akan dihapus	

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

	4. Aktor menekan button “Hapus” pada data transaksi yang dipilih	5. System menghapus data transaksi yang dipilih aktor
	6. Use case berakhir	
Altrenatif	Jika dalam proses penghapusan admin membatalkan proses tersebut, maka akan muncul pop-up “Apakan anda akan membatalkan penghapusan data transaksi?” dan akan tersedia button yes dan no. kemudian admin menekan yes dan system akan menampilkan list data transaksi	
Kondisi akhir sukses	Admin berhasil melakukan hapus data transaksi	
Kondisi akhir gagal	Admin tidak berhasil menghapus data transaksi	

Nama Use Case		Melihat data transaksi
Aktor	Admin	
Deskripsi Singkat	Use case ini memungkinkan admin untuk melihat data transaksi .	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor memilih button “lihat data transaksi”	2. System menampilkan list data transaksi
	3. Use case berakhir	
Altrenatif	-	
Kondisi akhir sukses	Admin dapat melihat data transaksi	
Kondisi akhir gagal	Admin tidak dapat melihat data transaksi	

Nama Use Case		Menambah data transaksi
Aktor	Admin	
Deskripsi Singkat	Use case ini memungkinkan admin untuk menambahkan data transaksi yang dibutuhkan	
Pra-syarat	Login	
Alur Utama	Aktor	System
	1. Aktor menekan button “Tambah data transaksi”	2. System menampilkan menu tambah transaksi
	3. Aktor mengisi detail transaksi yang ditambahkan	
	4. Aktor menekan button “Tambah”	5. System akan menambahkan data transaksi

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

	6. Use case berakhir
Altrenatif	jika detail transaksi yang dimasukkan tidak lengkap maka akan muncul pop up "masukkan detail transaksi dengan lengkap" kemudian kembali ke menu menambahkan user
Kondisi akhir sukses	Admin berhasil menambahkan data transaksi
Kondisi akhir gagal	Admin gagal menambahkan data transaksi

Nama Use Case		Mendapatkan password baru	
Aktor	Pengguna Bengkel & Pemilik Bengkel		
Deskripsi Singkat	Use case ini memungkinkan aktor untuk mendapatkan password baru ketika aktor lupa password lama		
Pra-syarat	Terdaftar		
Alur Utama	Aktor	System	
	1. Menekan button “Lupa password”	2. System menampilkan menu lupa password dengan kolom email aktor	
	3. Aktor menuliskan email yang terdaftar	4. System mengirimkan password baru ke email	
	5. Use case berakhir		
Altrenatif	Jika email yang dimasukkan tidak terdaftar maka akan muncul pop up “Email tidak terdaftar, silahkan masukkan kembali” kemudian masuk ke menu lupa password		
Kondisi akhir sukses	Aktor berhasil memperoleh password baru		
Kondisi akhir gagal	Aktor tidak mendapatkan memperoleh password baru		

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

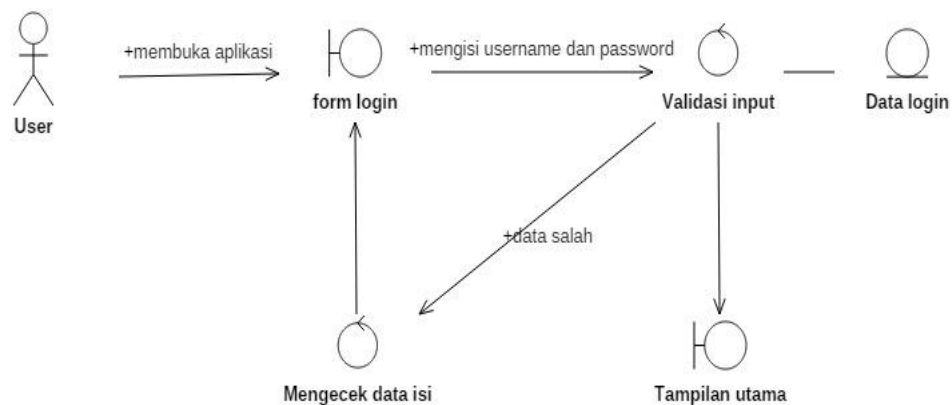
4 Robustness Diagram

Fungsi robustness diagram adalah membantu untuk menjembatani adanya ruang kosong dari analisis desain yang sebenarnya terhadap implementasi kode. Atribut dari robustness Diagram antara lain:

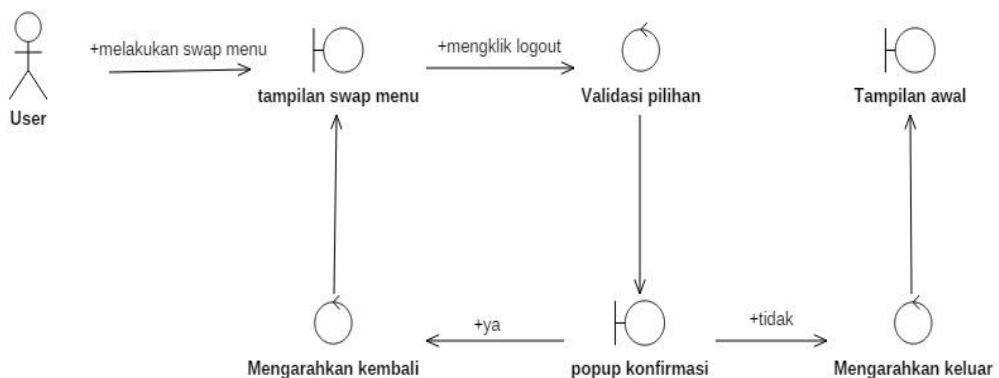
- Obyek Boundary : interface antara sistem dan dunia luar. boundary objek merupakan halaman halaman pada aplikasi BengkelQ
- Obyek entitas : kelas dari domain model
- Obyek Controllers : penghubung antara boundary objek dan objek entitas

Robustness diagram dibentuk dari masing masing use case, berikut diagram robustness BengkelQ :

1. Login Aplikasi BengkelQ

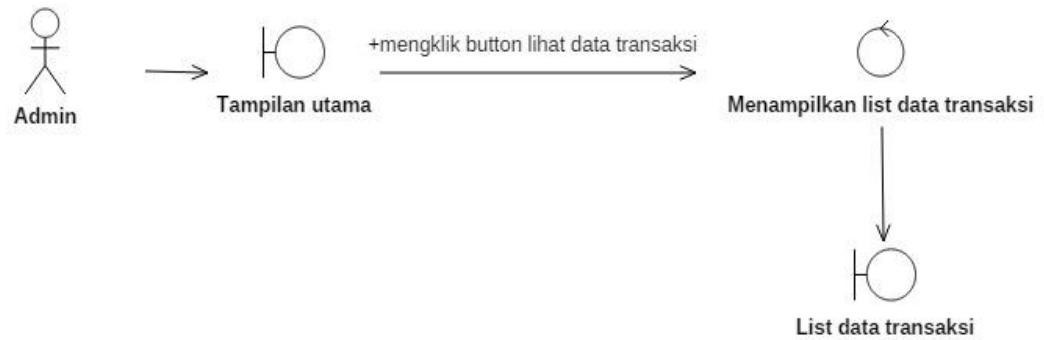


2. Logout Aplikasi BengkelQ



XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

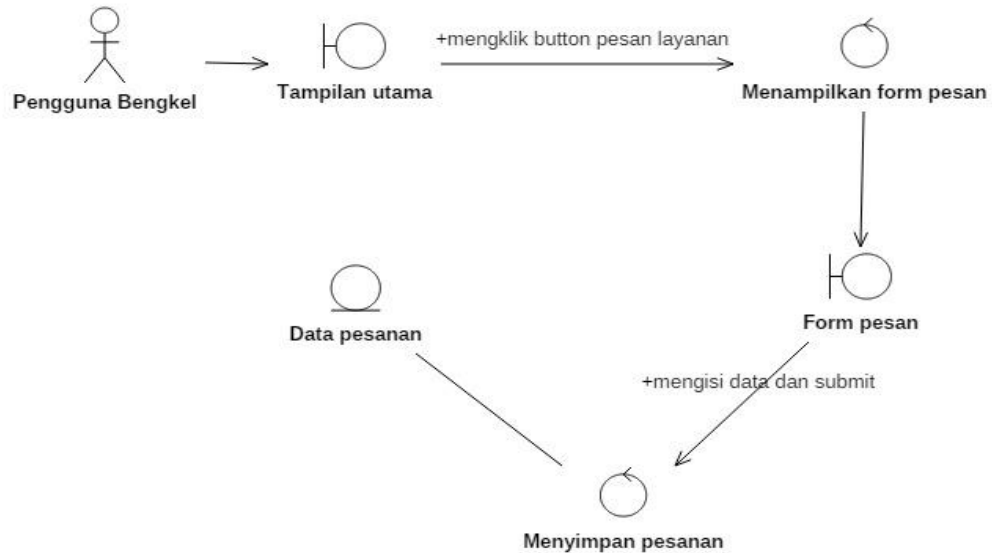
3. Melihat data transaksi



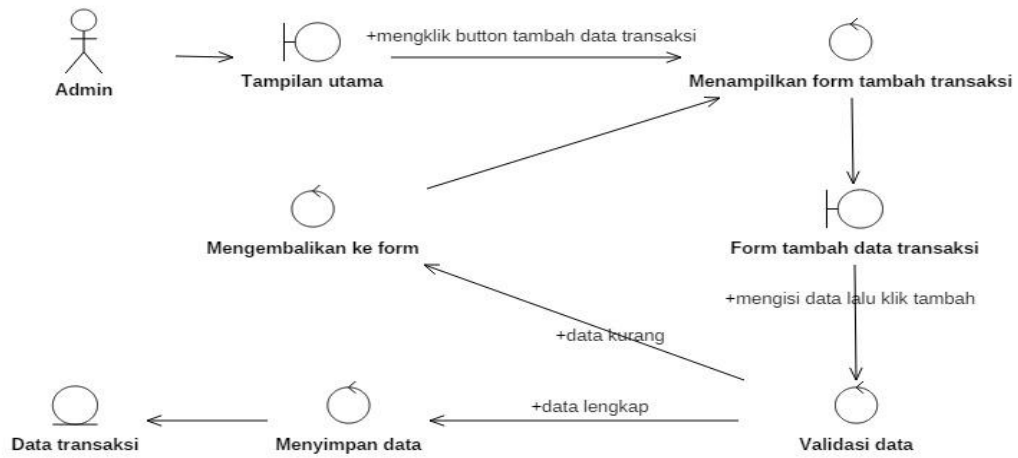
4. Melihat data user



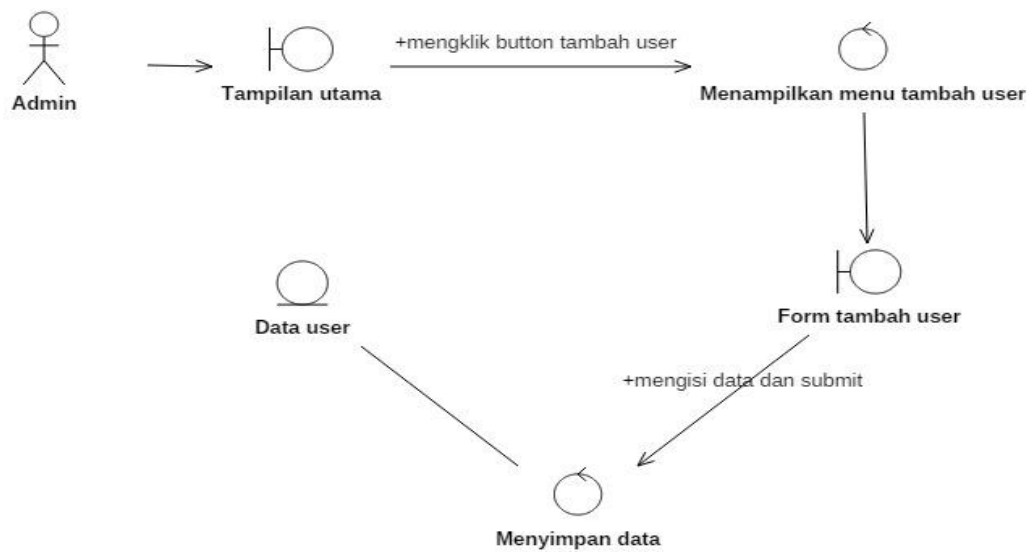
5. Memesan layanan Bengkel



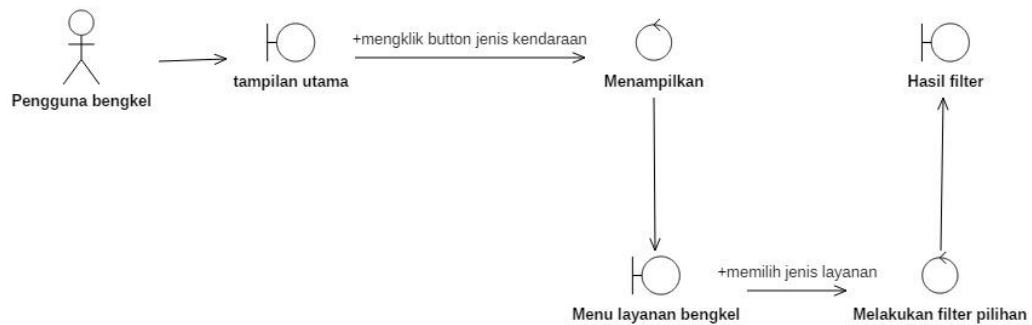
6. Menambah data transaksi



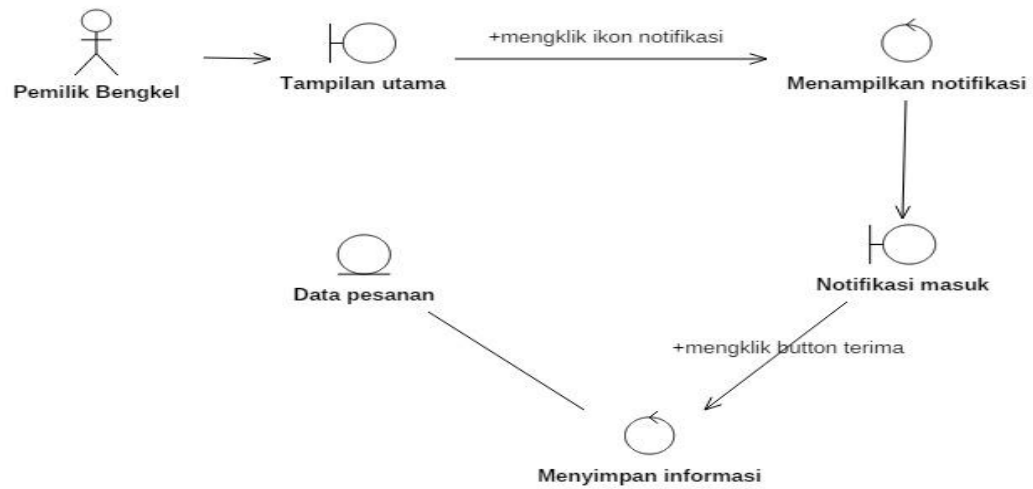
7. Menambah data user



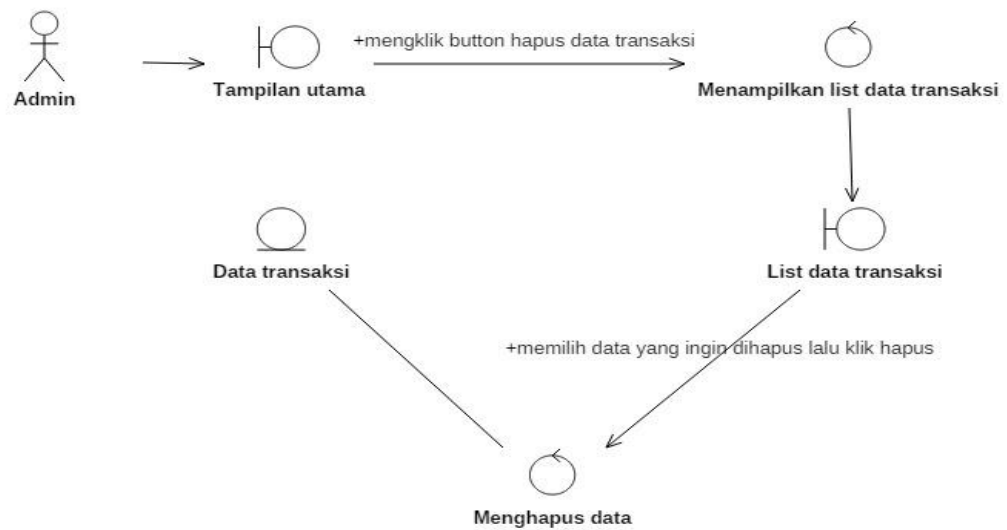
8. Mencari Bengkel yang sesuai



9. Menerima Layanan Pesanan

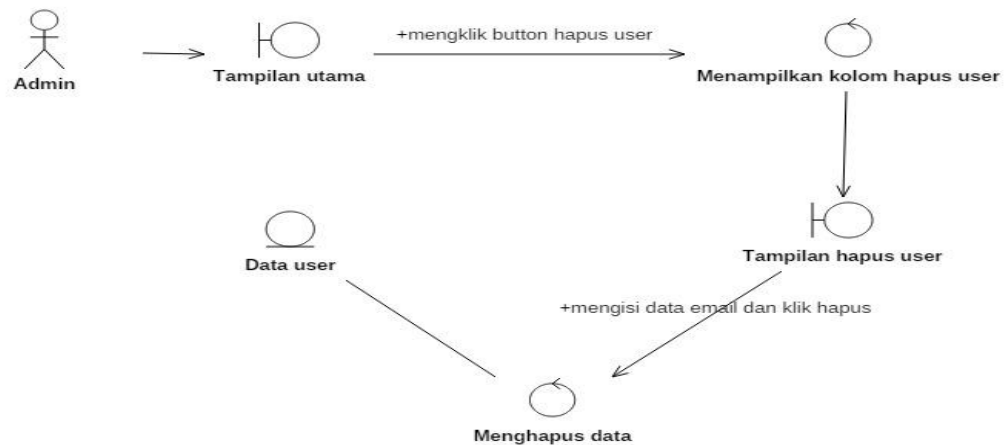


10. Menghapus data transaksi

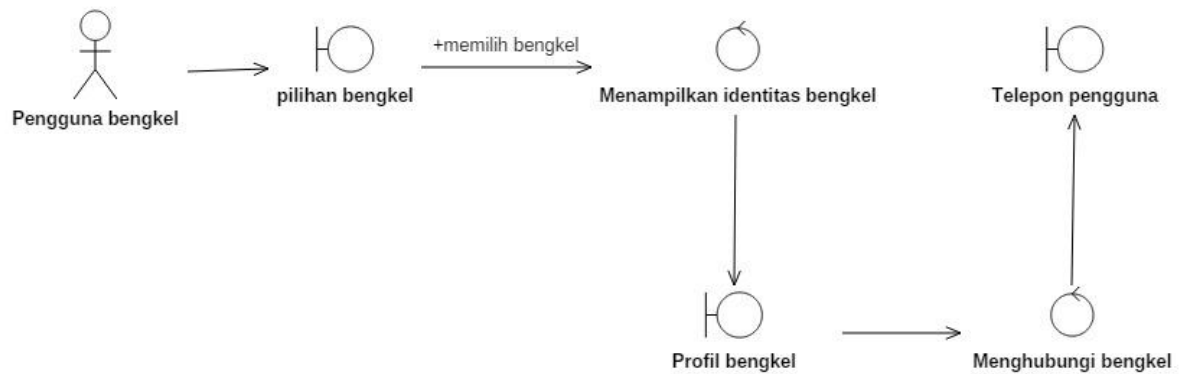


XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

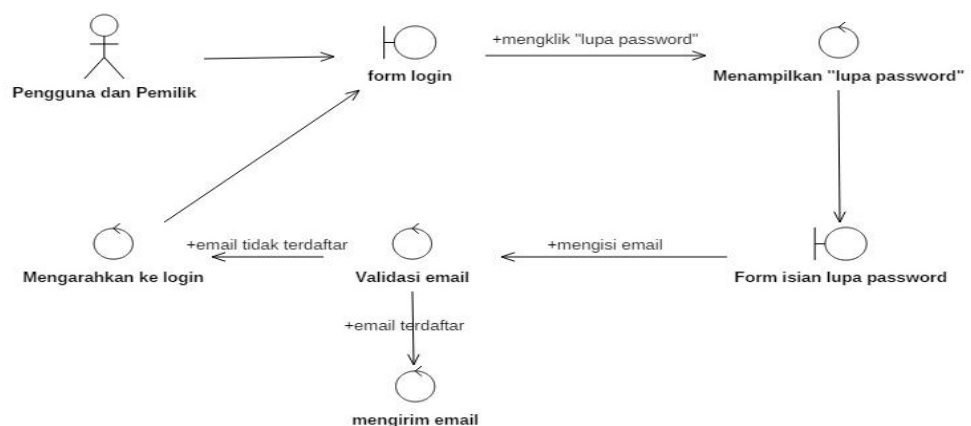
11. Menghapus data user



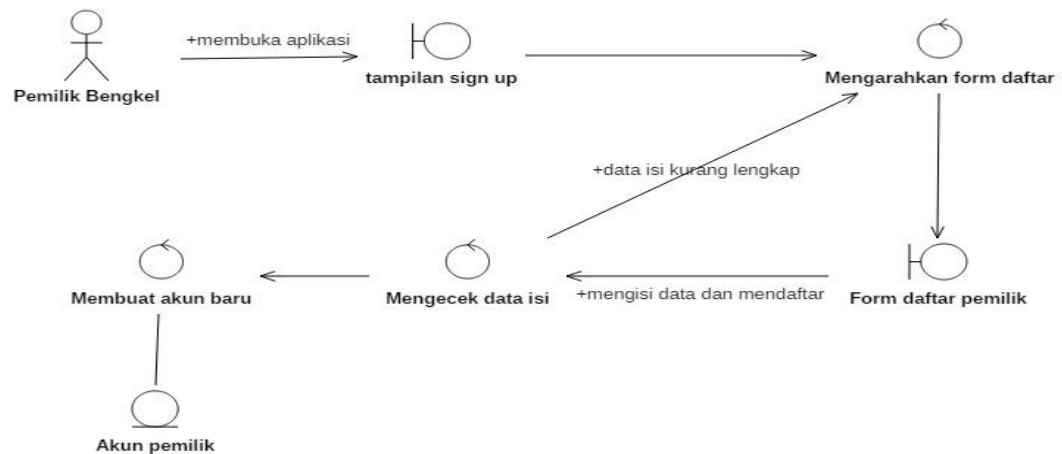
12. Menghubungi Bengkel



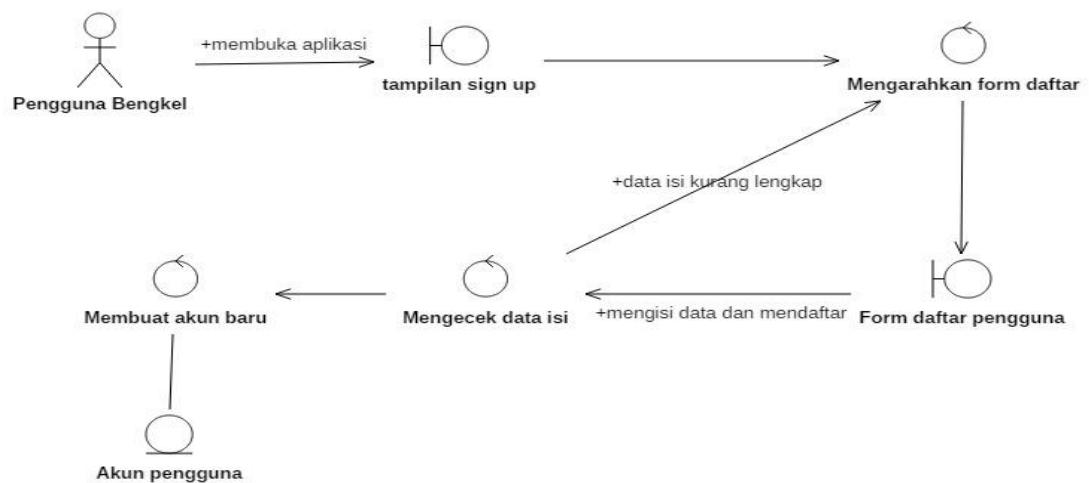
13. Mendapatkan password baru



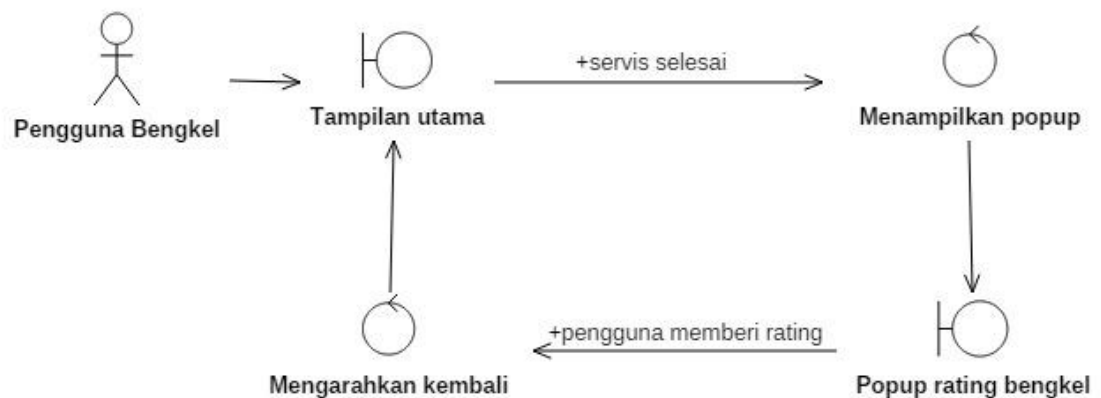
14. Mendaftar pemilik bengkel



15. Mendaftar pengguna bengkel

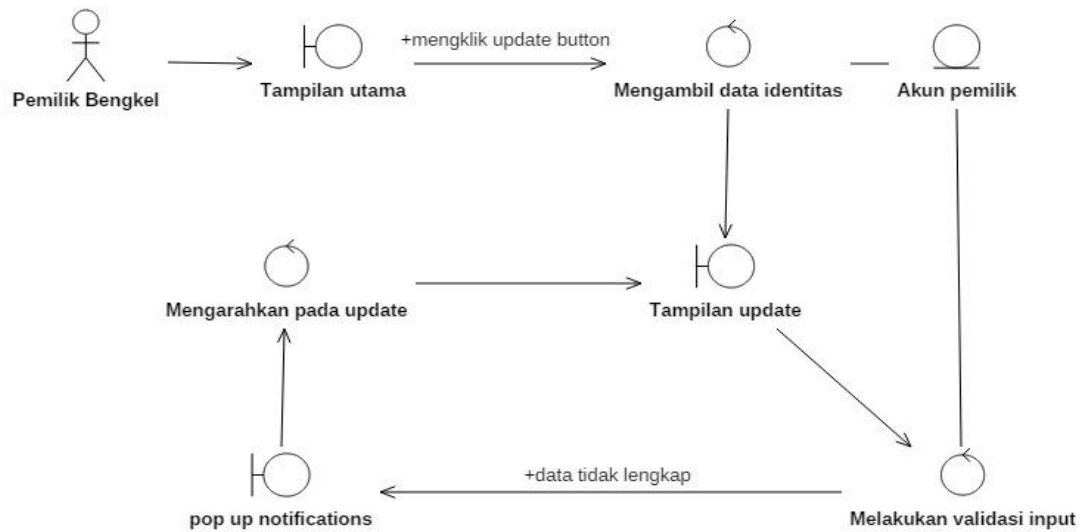


16. Memberi rating bengkel



17. Update data bengkel

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	



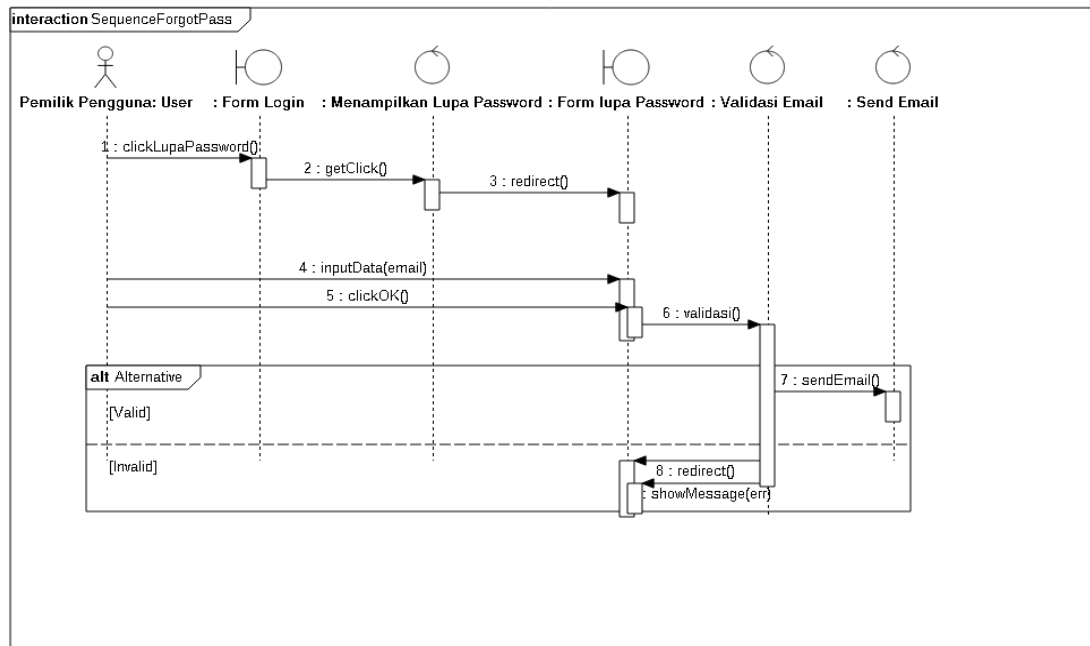
5 Sequence Diagram

Diagrams sequence atau sering disebut sequence diagrams dalam UML (Unified Modeling Language) adalah diagrams yang menggambarkan kelakuan objek pada use case dengan mendiskripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek.

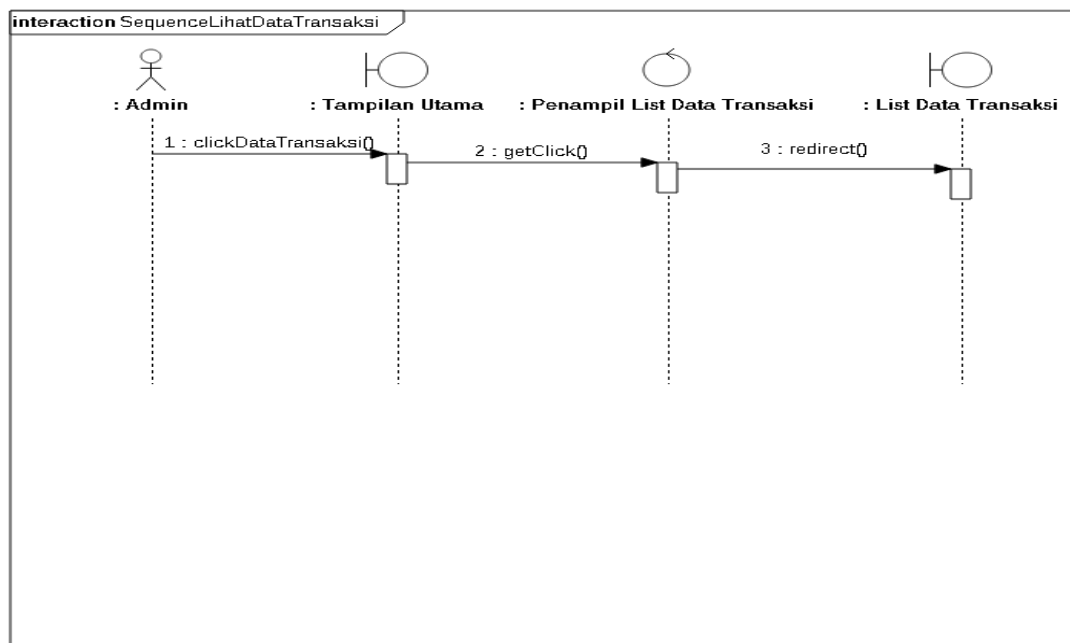
Simbol	Deskripsi	Simbol	Deskripsi
Aktor Nama_aktor Atau Nama_aktor	Orang, proses atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda diawal frase nama aktor	Pesan tipe create <<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat
		Pesan tipe call 1 : nama_metode()	Menyatakan suatu objek memanggil operasi/metode yang ada pada objek lain atau dirinya sendiri.
		Pesan tipe send 1 : masukan	Menyatakan bahwa suatu objek mengirimkan data / masukan / informasi ke objek lainnya
Garis hidup / lifeline 	Menyatakan kehidupan suatu objek.	pesan tipe return 1 : keluaran	Menyatakan bahwa suatu objek yang sudah menjalankan suatu operasi atau metode menghasilkan kembalian ke objek tertentu.
Objek Nama objek : nama kelas	Menyatakan objek yang berinteraksi pesan	Pesan tipe destroy <<destroy>>	Menyatakan suatu objek mengakhiri hidup objek lain, arah panah mengarah pada objek yang di akhiri.
Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang berhubungan dengan waktu aktif ini adalah sebuah tahapan yang dilakukan didalamnya.		

Berikut merupakan Sequence diagram aplikasi BengkelQ:

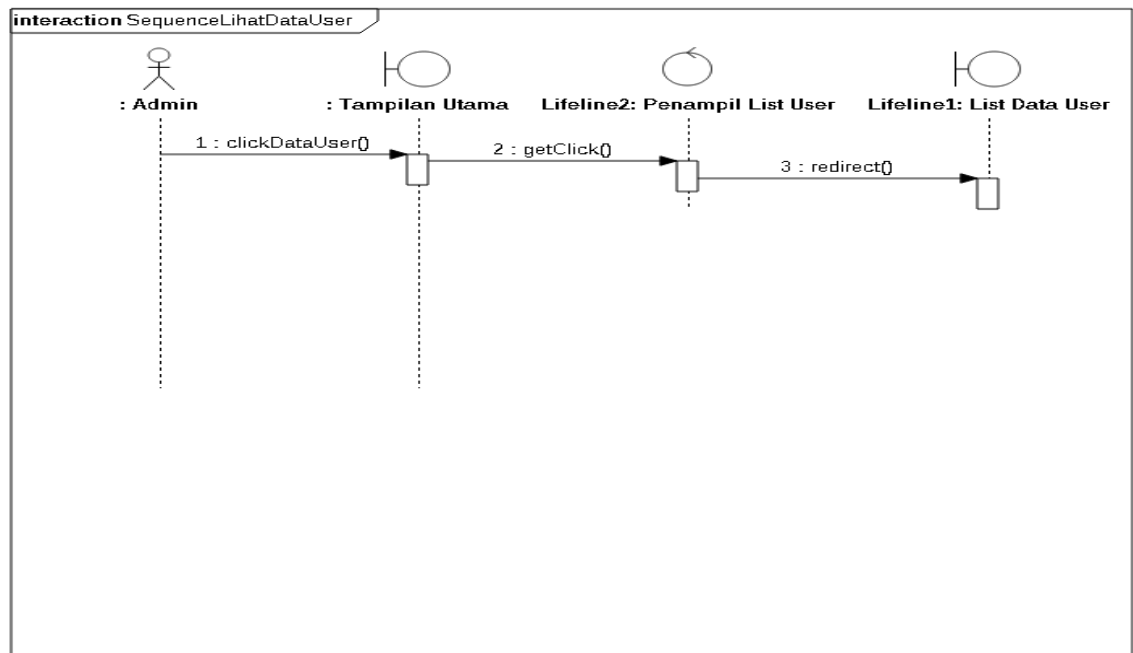
1. Sequence meminta password baru



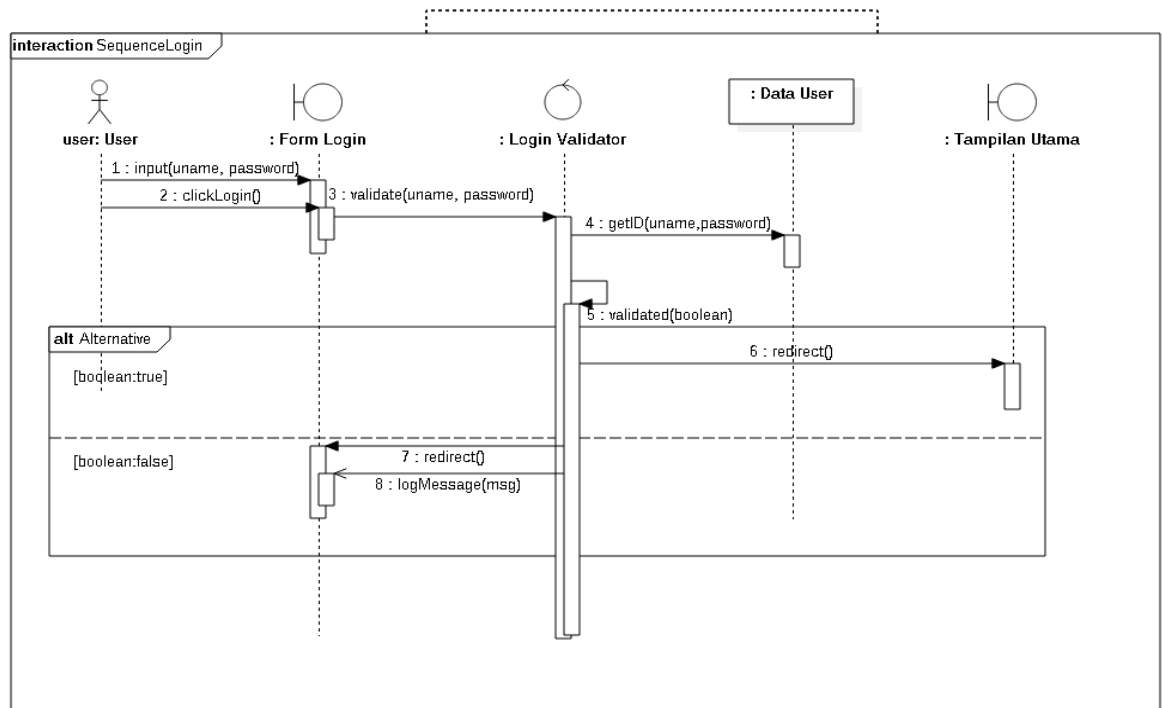
2. Sequence melihat data transaksi



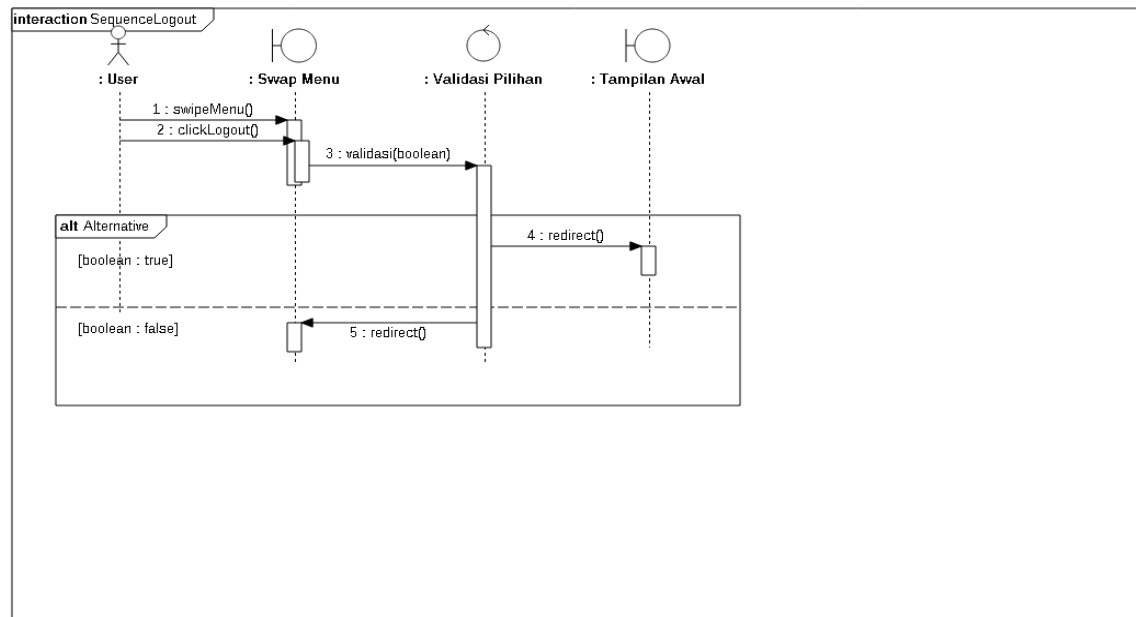
3. Sequence melihat data User



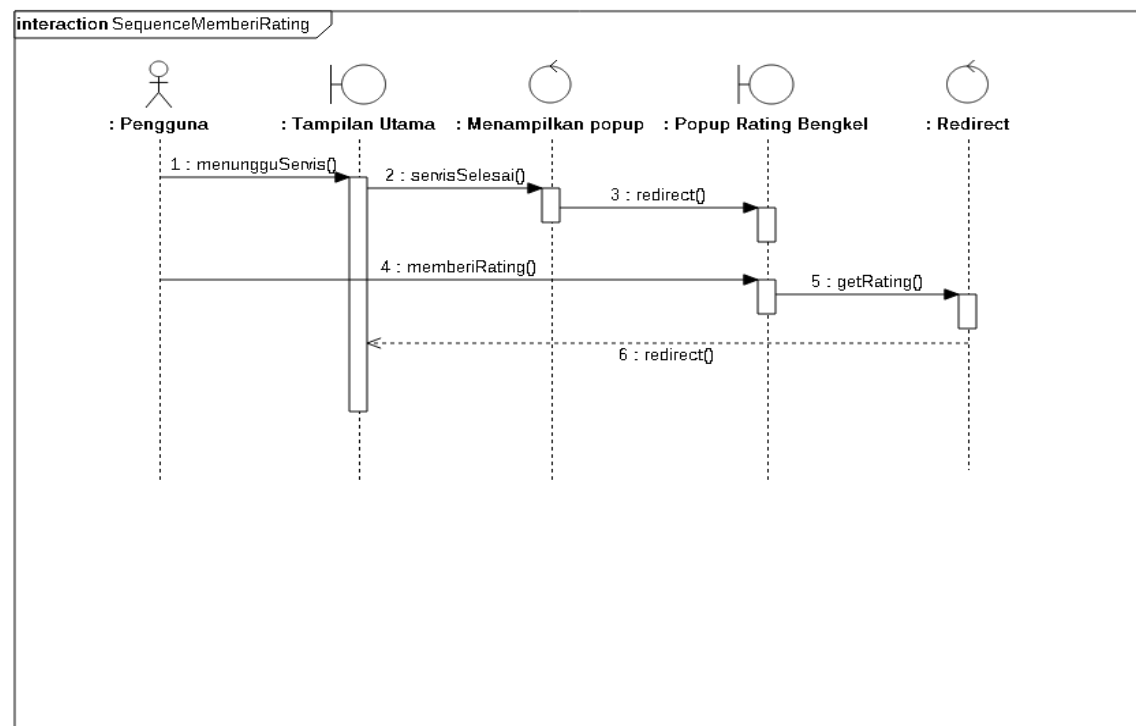
4. Sequence login aplikasi BengkelQ



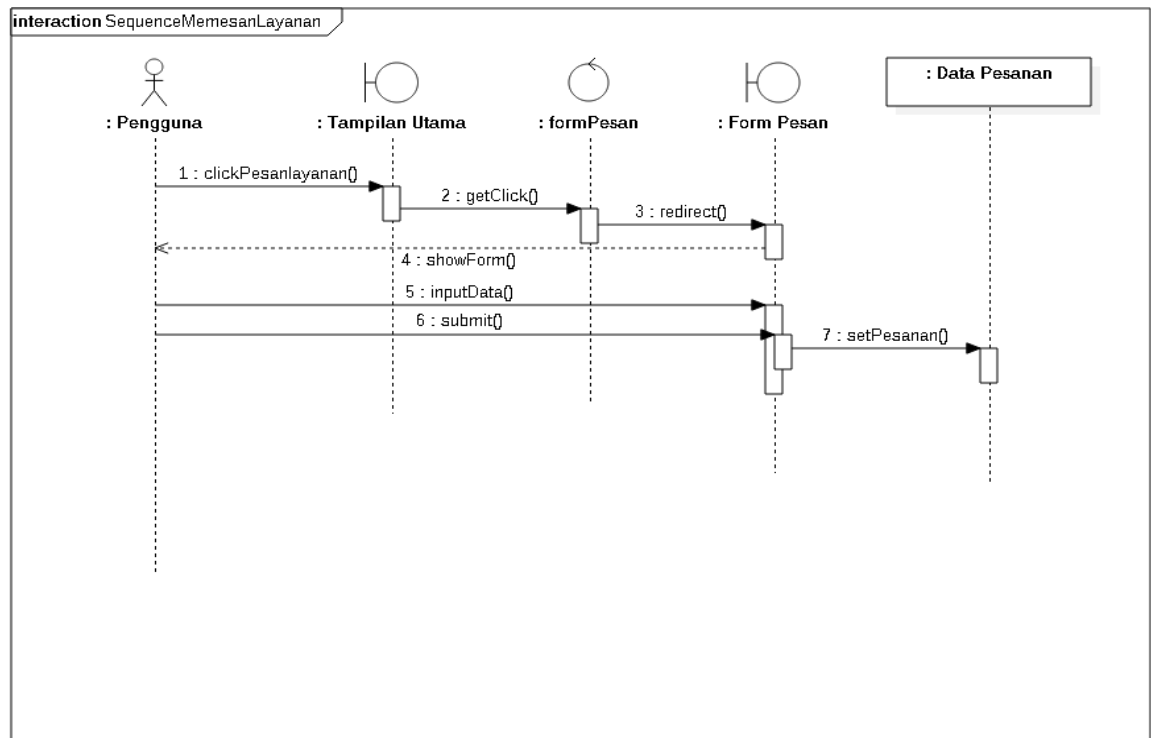
5. Sequence Logout aplikasi BengkelQ



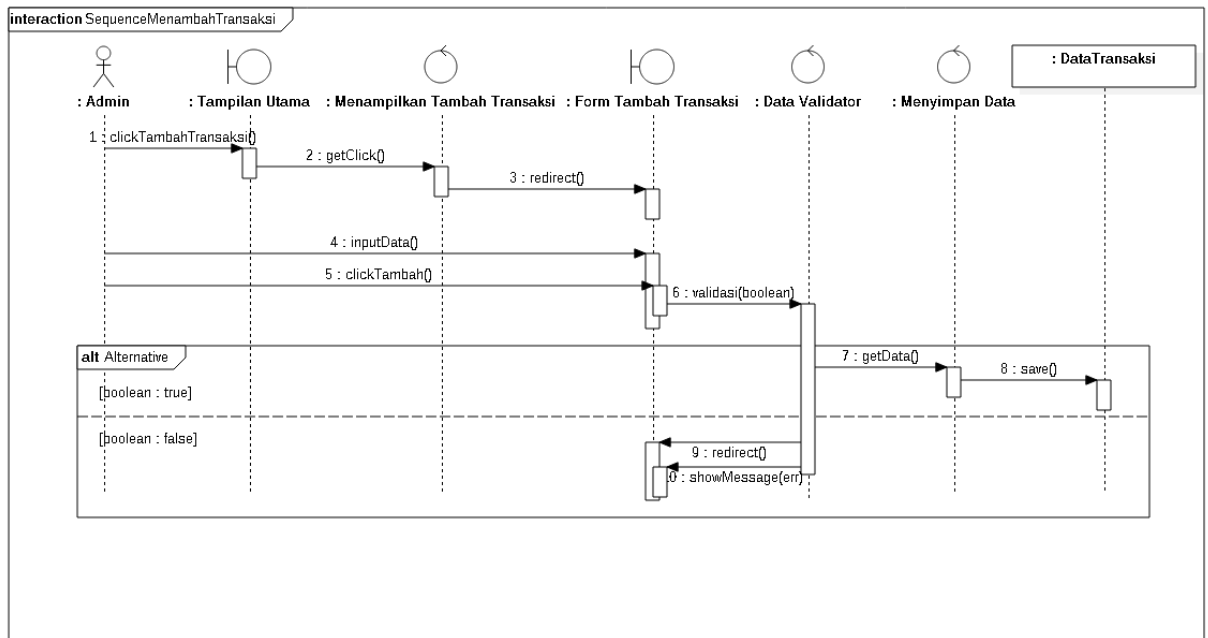
6. Sequence memberi rating bengkel



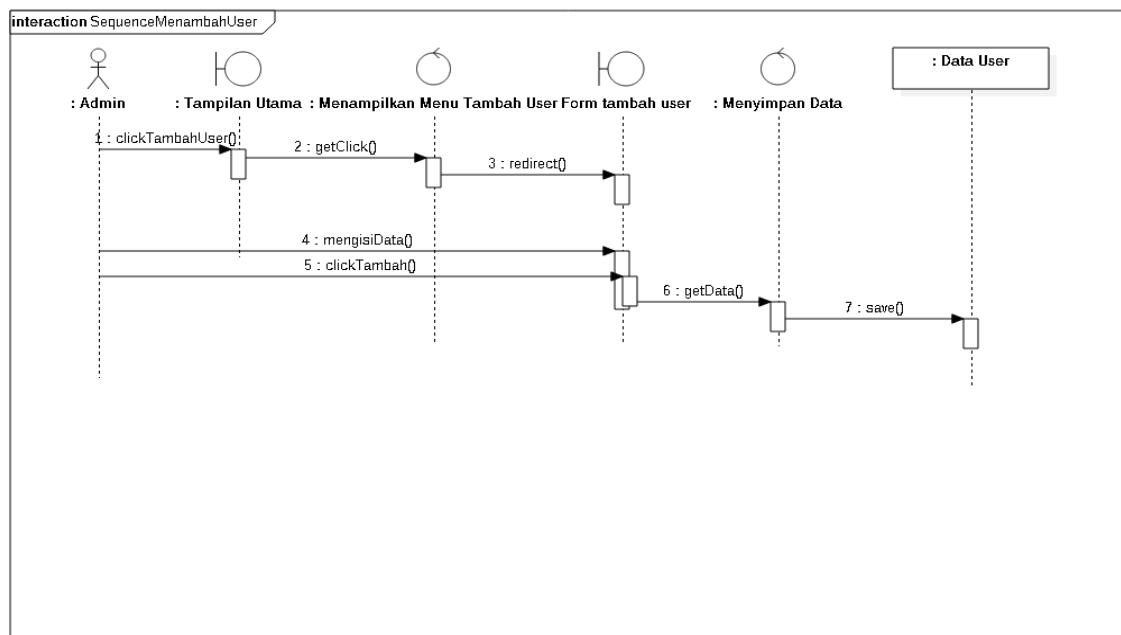
7. Sequence memesan layanan bengkel



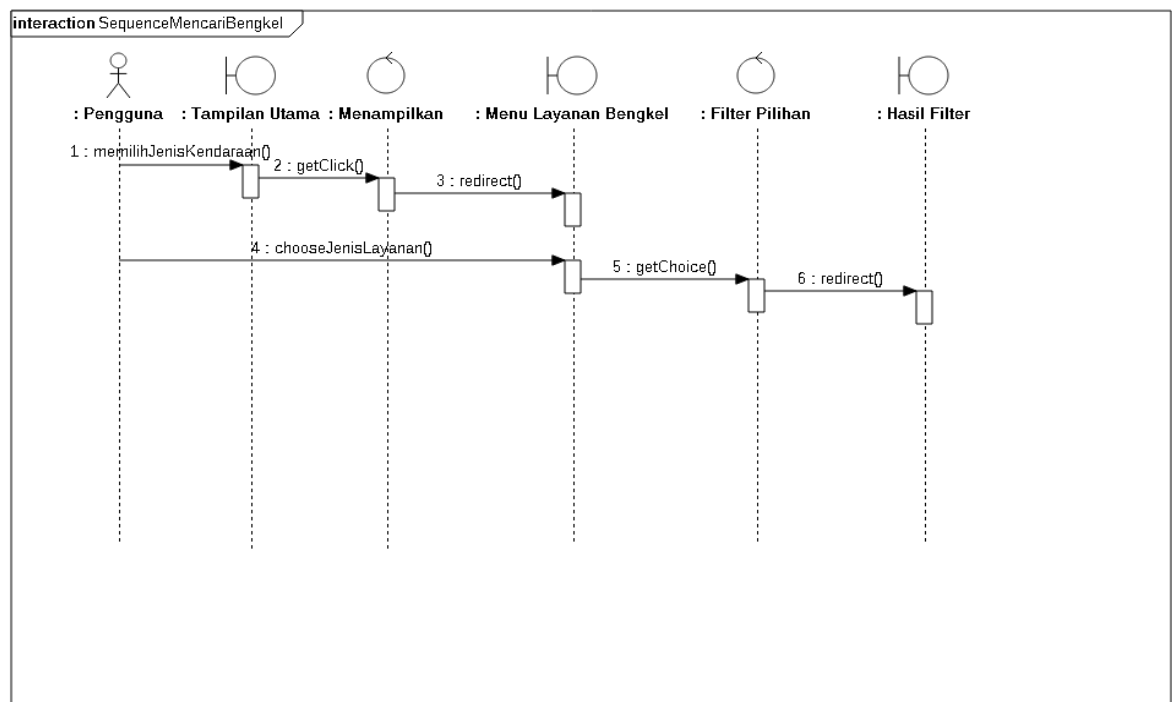
8. Sequence menambah data transaksi



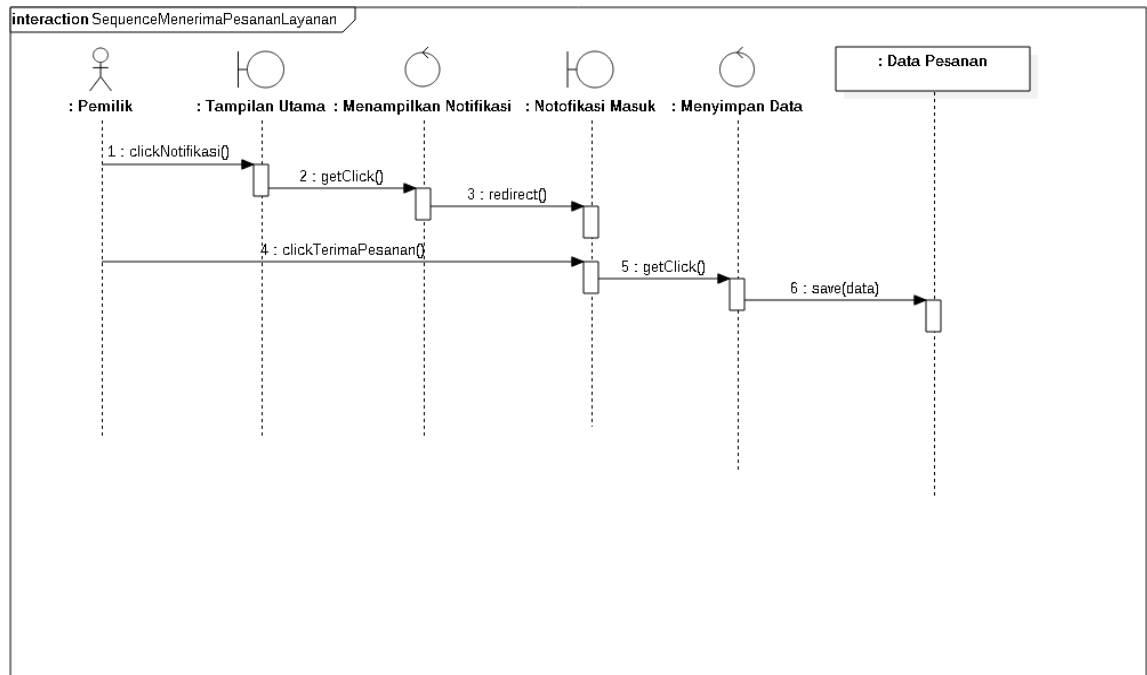
9. Sequence menambah data user



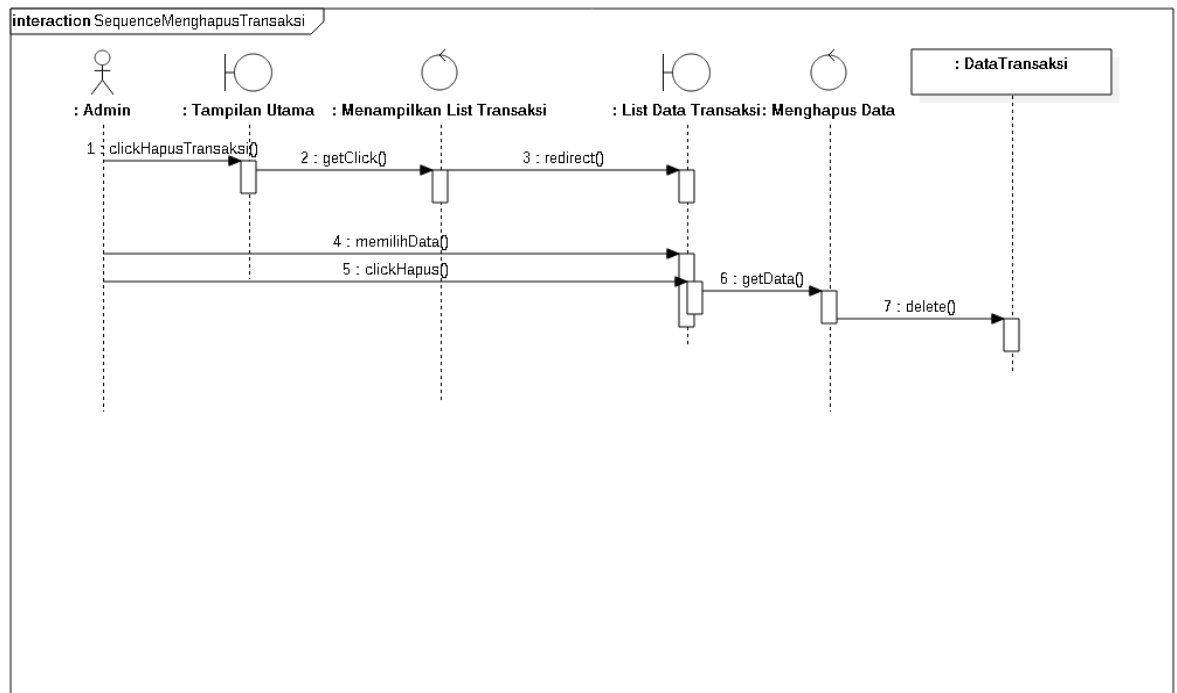
10. Sequence mencari bengkel yang sesuai



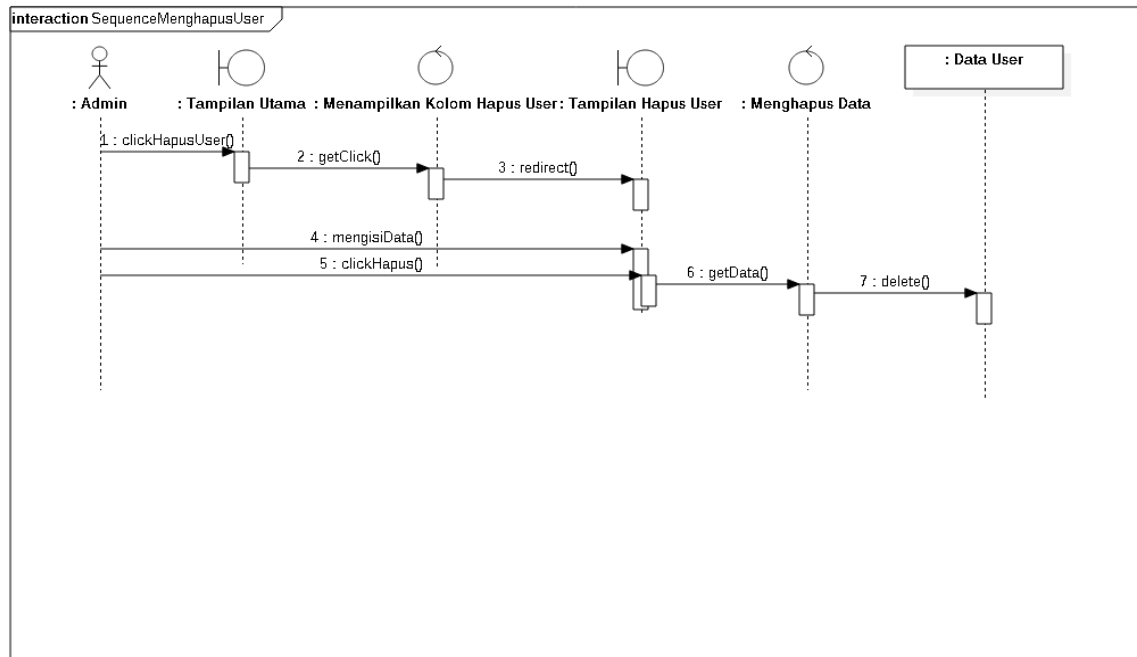
11. Sequence menerima pesanan layanan



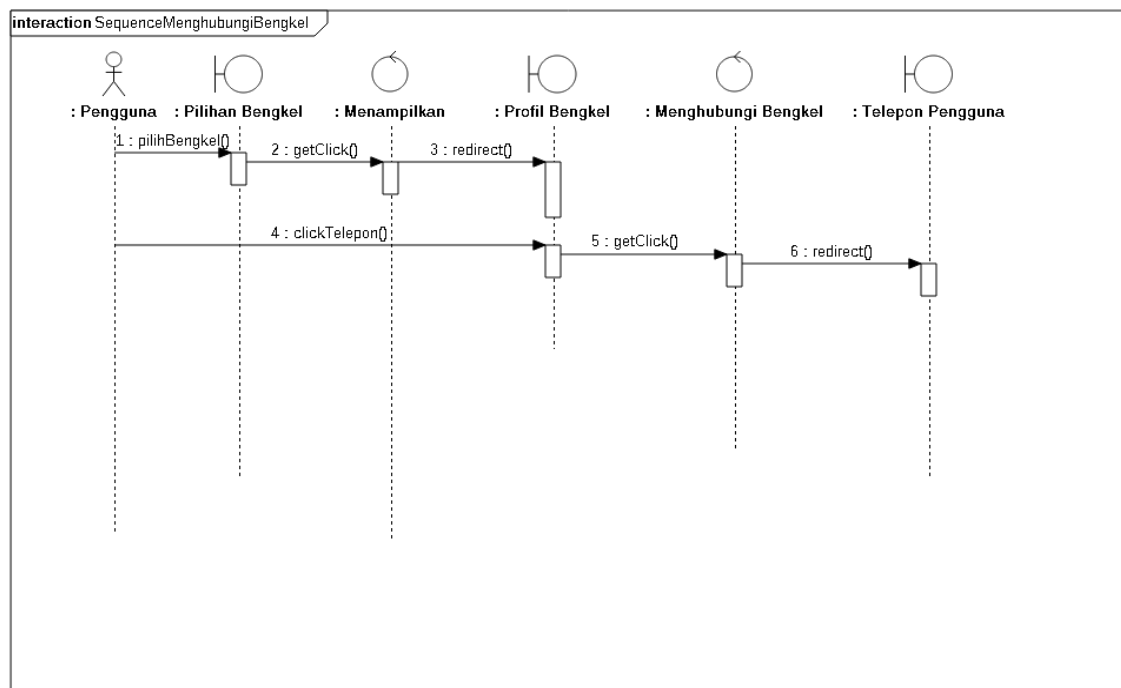
12. Sequence menghapus data transaksi



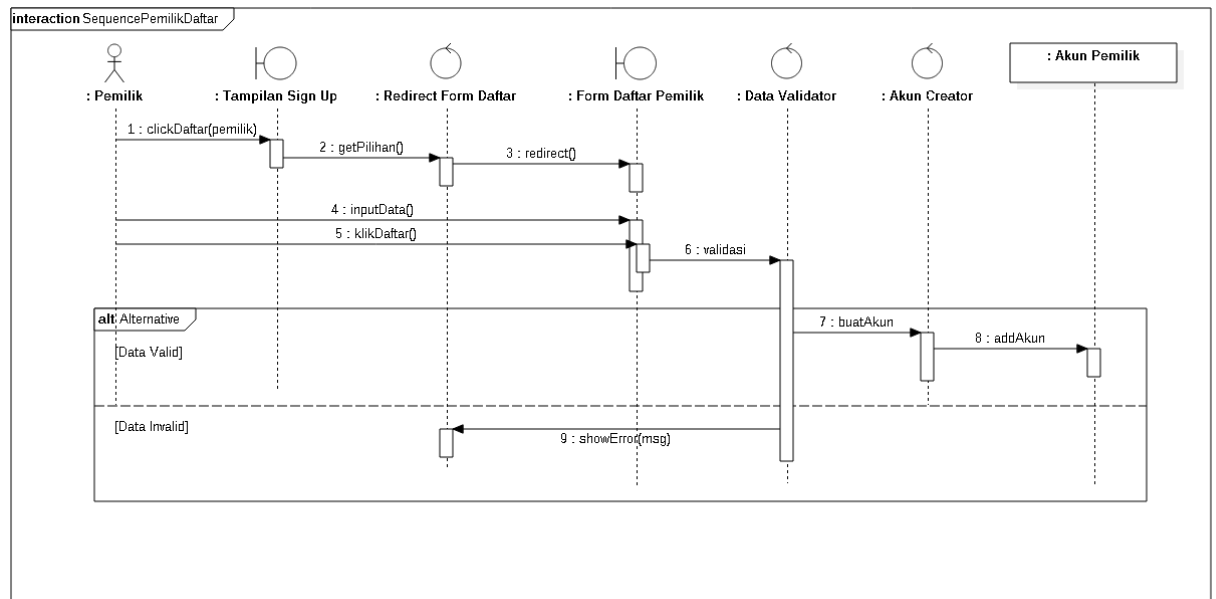
13. Sequence menghapus data user



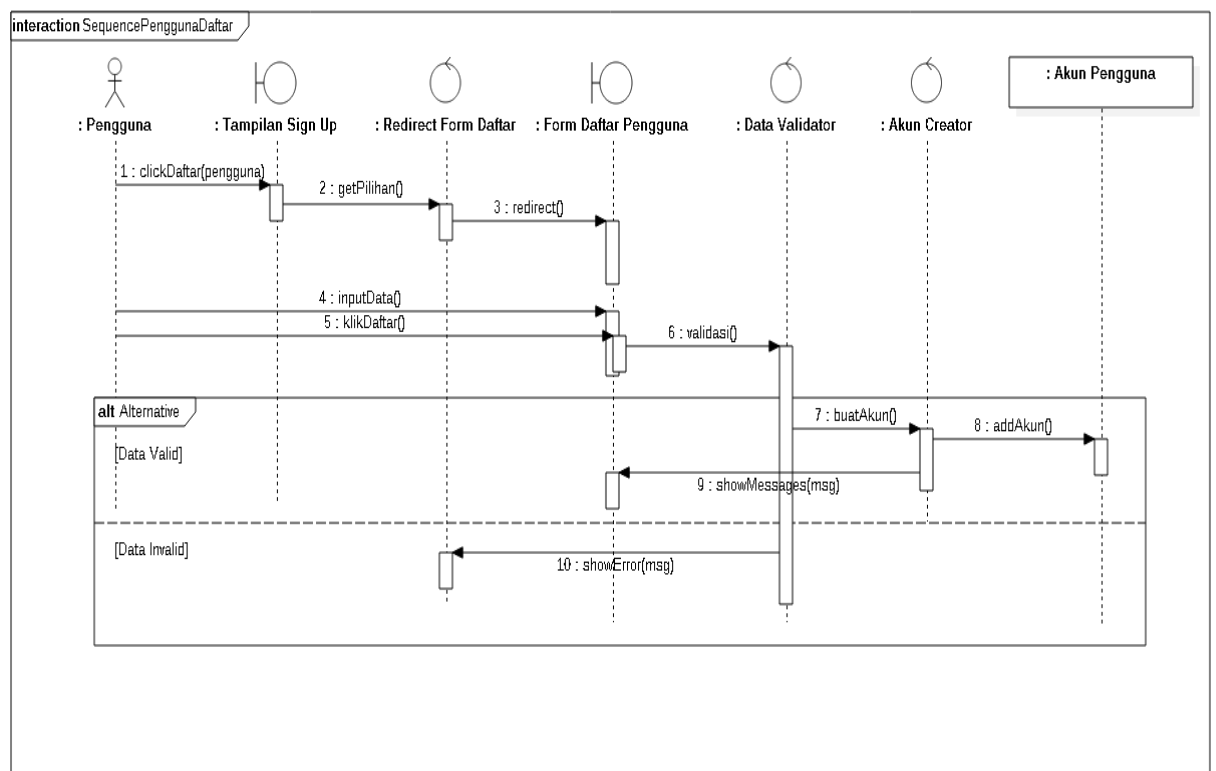
14. Sequence menghubungi bengkel terkait



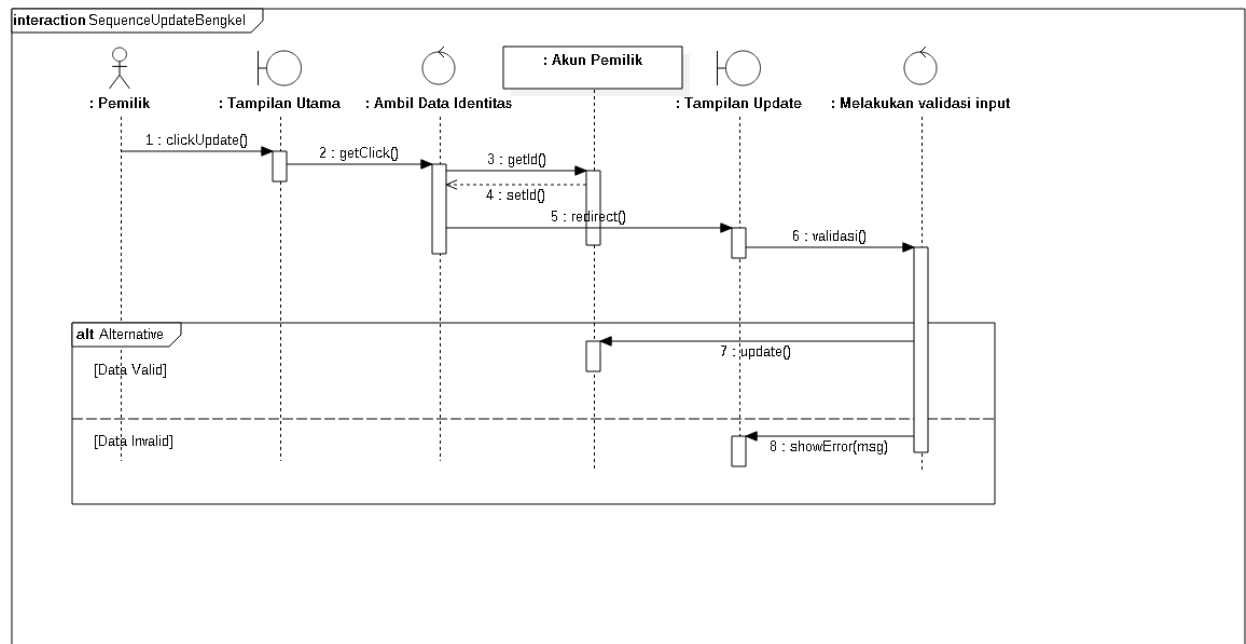
15. Sequence mendaftar pemilik bengkel



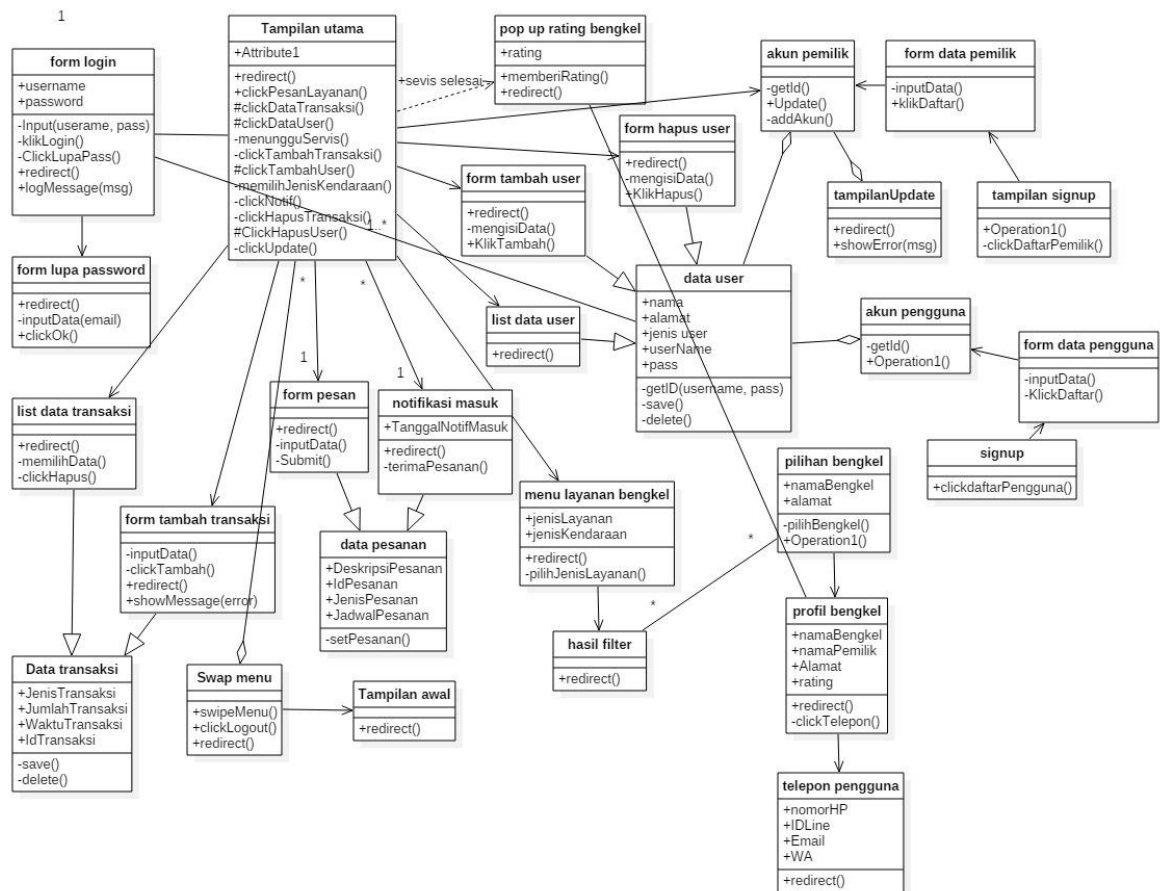
16. Sequence mendaftar pengguna bengkel



17. Sequence update data bengkel



6 Class Diagram

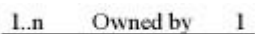
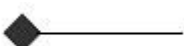

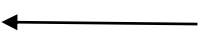


Gambar 8.1 Class Diagram BengkelQ

Class diagram adalah diagram yang digunakan untuk menampilkan beberapa kelas serta atribut dan method yang ada dalam perangkat lunak yang sedang dibangun Class diagram memberi gambaran (diagram statis) tentang sistem/perangkat lunak dan relas-relasi yang ada didalamnya. Berikut komponen komponen Class diagram:

Nama Komponen	Keterangan	Simbol
<i>Class</i>	<i>Class</i> adalah blok - blok pembangun pada pemrograman berorientasi obyek. Sebuah <i>class</i> digambarkan sebagai sebuah kotak yang terbagi atas 3 bagian. Bagian atas adalah bagian nama dari <i>class</i> . Bagian tengah mendefinisikan property/atribut <i>class</i> . Bagian akhir mendefinisikan <i>methodmethod</i> dari sebuah <i>class</i> .	Nama <i>Class</i>
		+ atribut
		+ atribut
		+ atribut
		+ <i>method</i>
		+ <i>method</i>

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

Association	Sebuah asosiasi merupakan sebuah relationship paling umum antara 2 <i>class</i> dan dilambangkan oleh sebuah garis yang menghubungkan antara 2 <i>class</i> . Garis ini bisa melambangkan tipe-tipe <i>relationship</i> dan juga dapat menampilkan hukum-hukum multiplisitas pada sebuah <i>relationship</i> . (Contoh: <i>One-to-one</i> , <i>one-to-many</i> , <i>many-to-many</i>).	
Composition	Jika sebuah <i>class</i> tidak bisa berdiri sendiri dan harus merupakan bagian dari <i>class</i> yang lain, maka <i>class</i> tersebut memiliki relasi <i>Composition</i> terhadap <i>class</i> tempat dia bergantung tersebut. Sebuah <i>relationship composition</i> digambarkan sebagai garis dengan ujung berbentuk jajaran genjang berisi/solid.	
Aggregation	<i>Aggregation</i> mengindikasikan keseluruhan bagian <i>relationship</i> dan biasanya disebut sebagai relasi	
Generalization	Mengindikasikan adanya subclass dan superclass	

XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

7 GUI

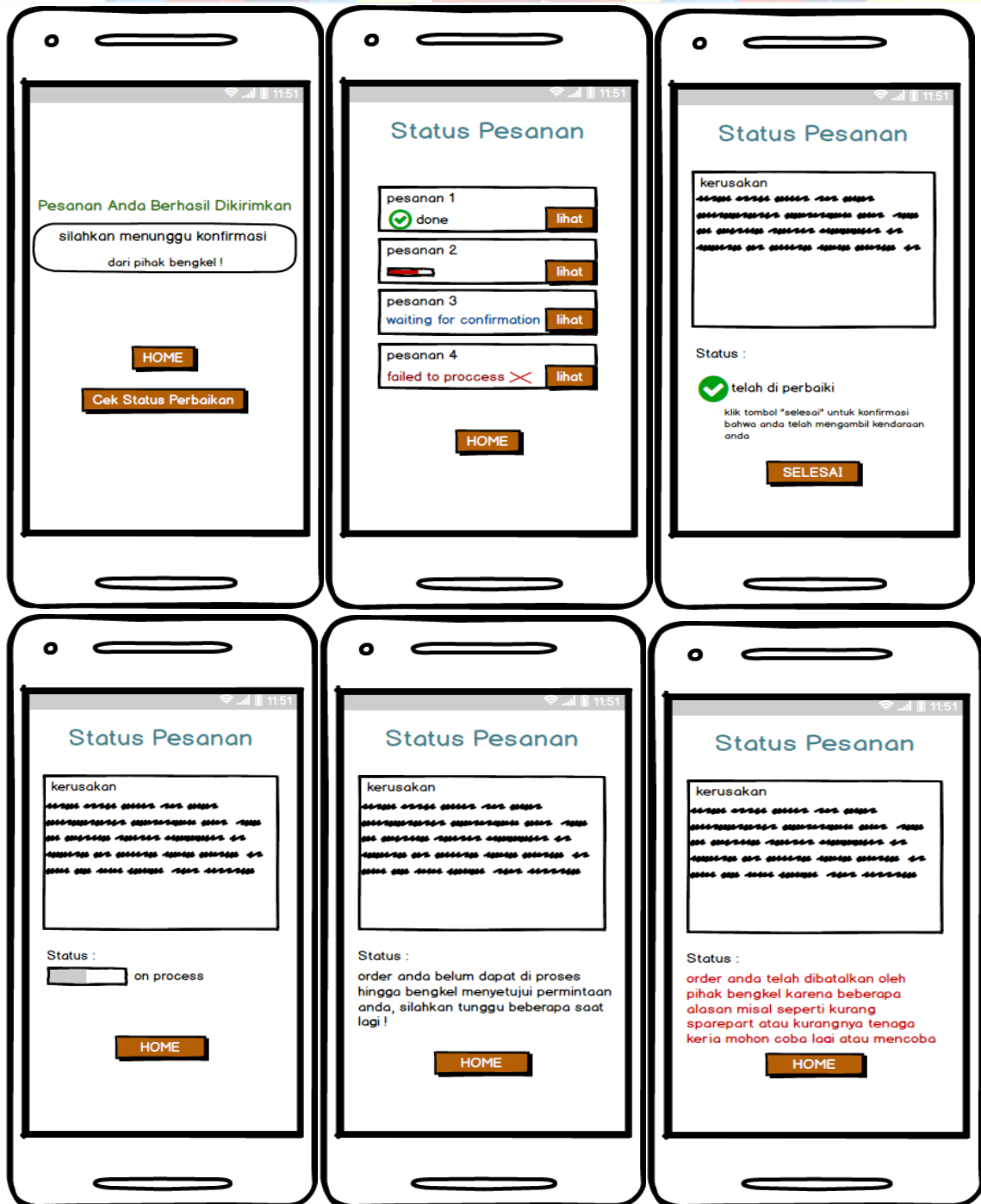
1. User View Pengguna Bengkel

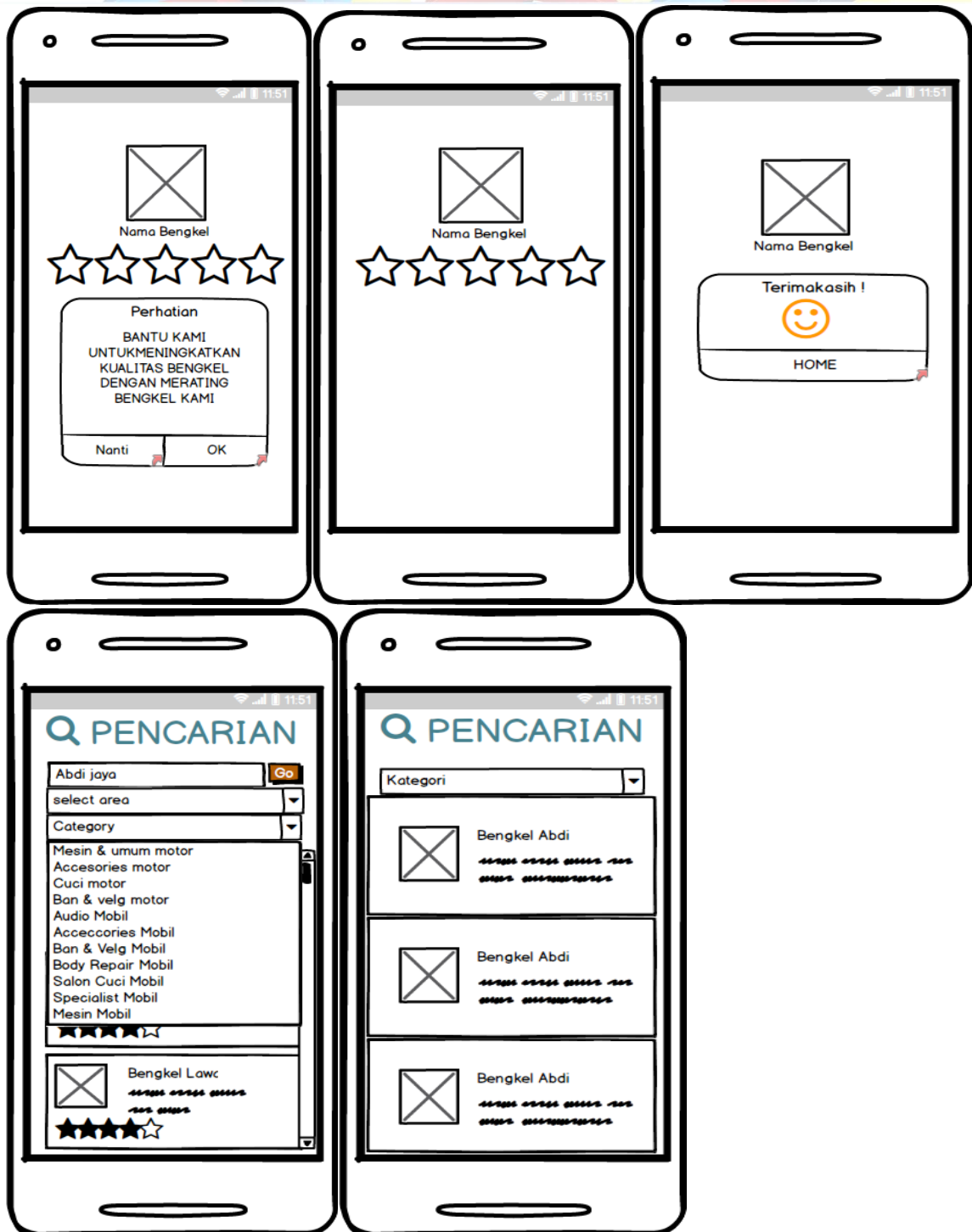


XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	



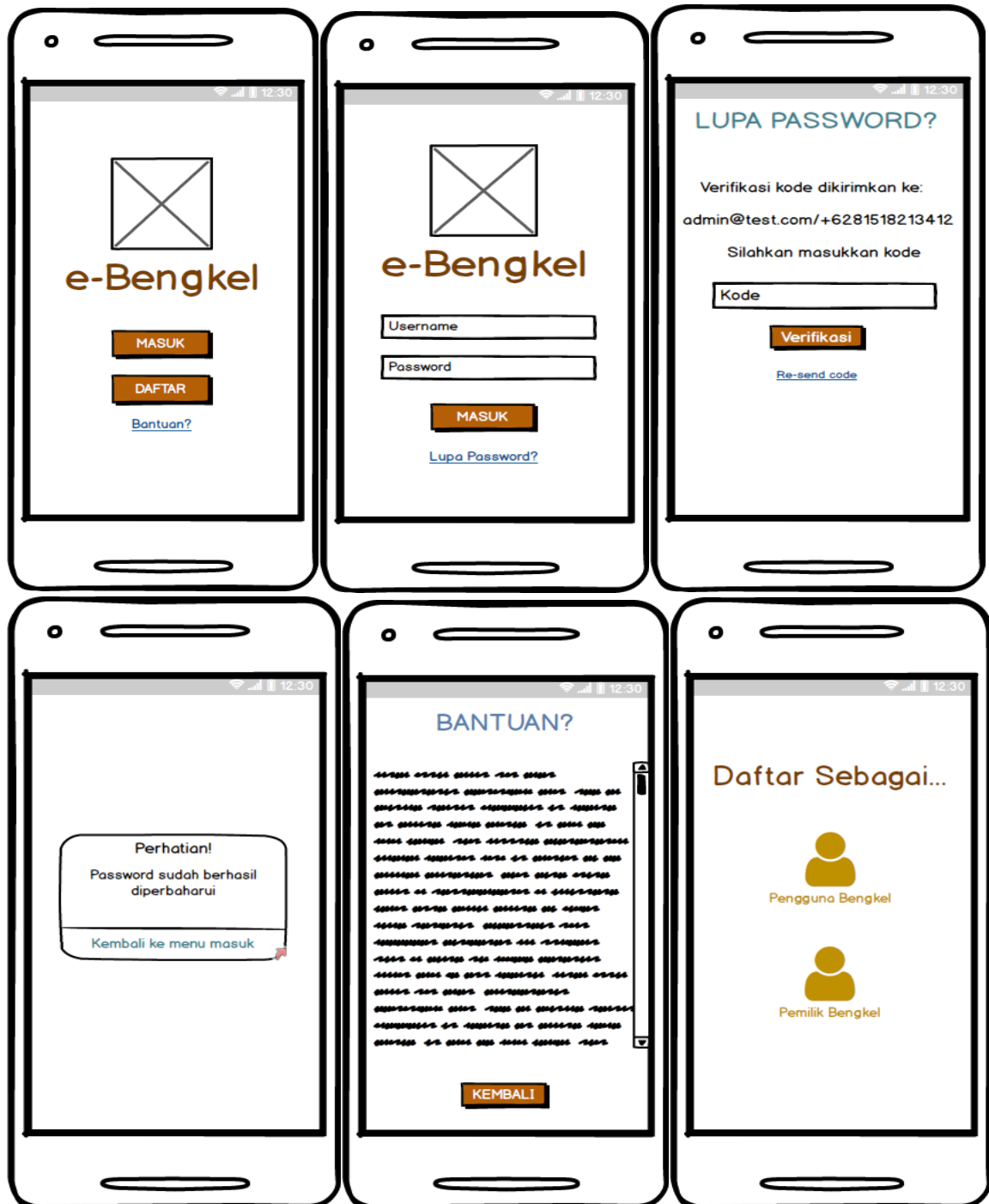






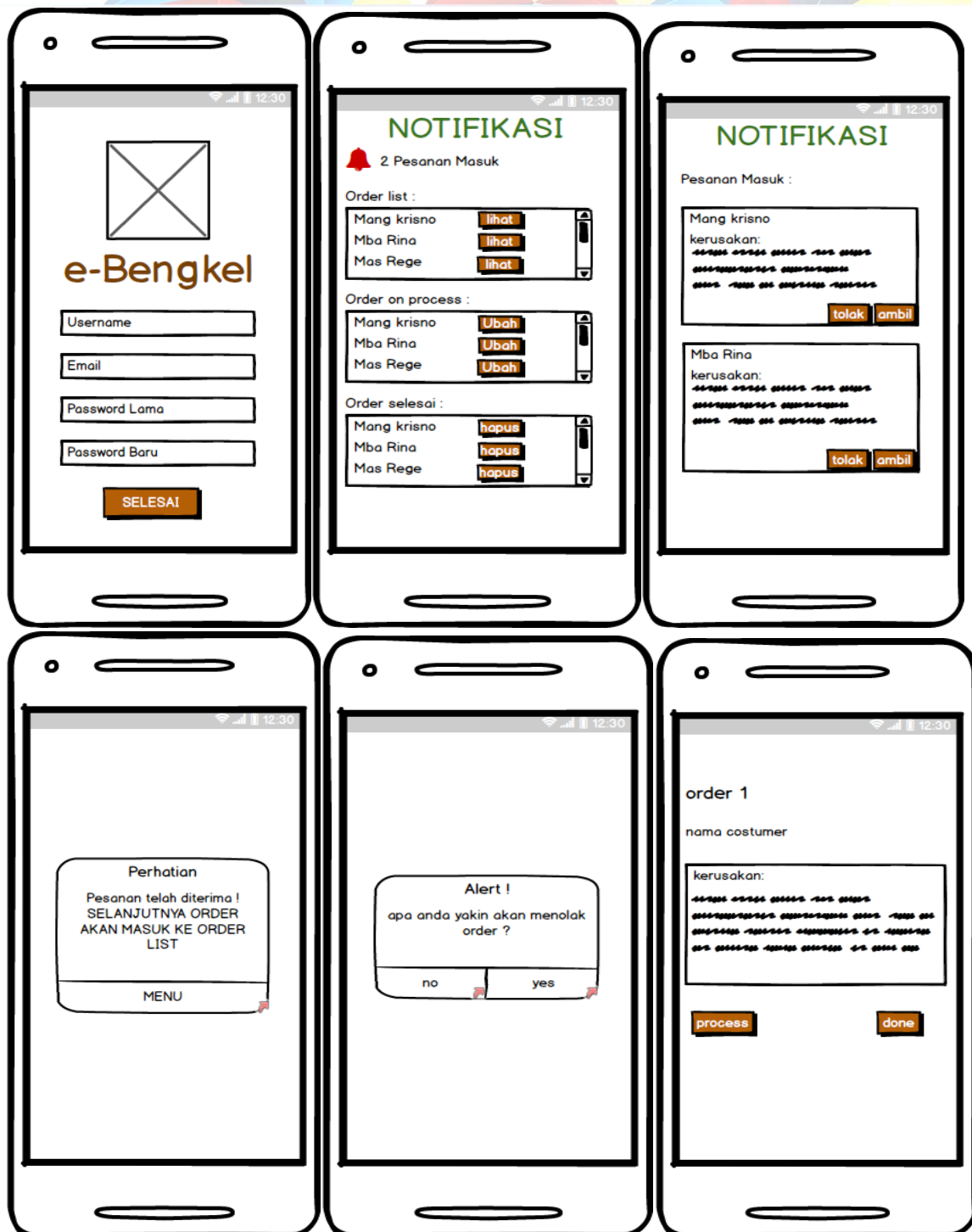
XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

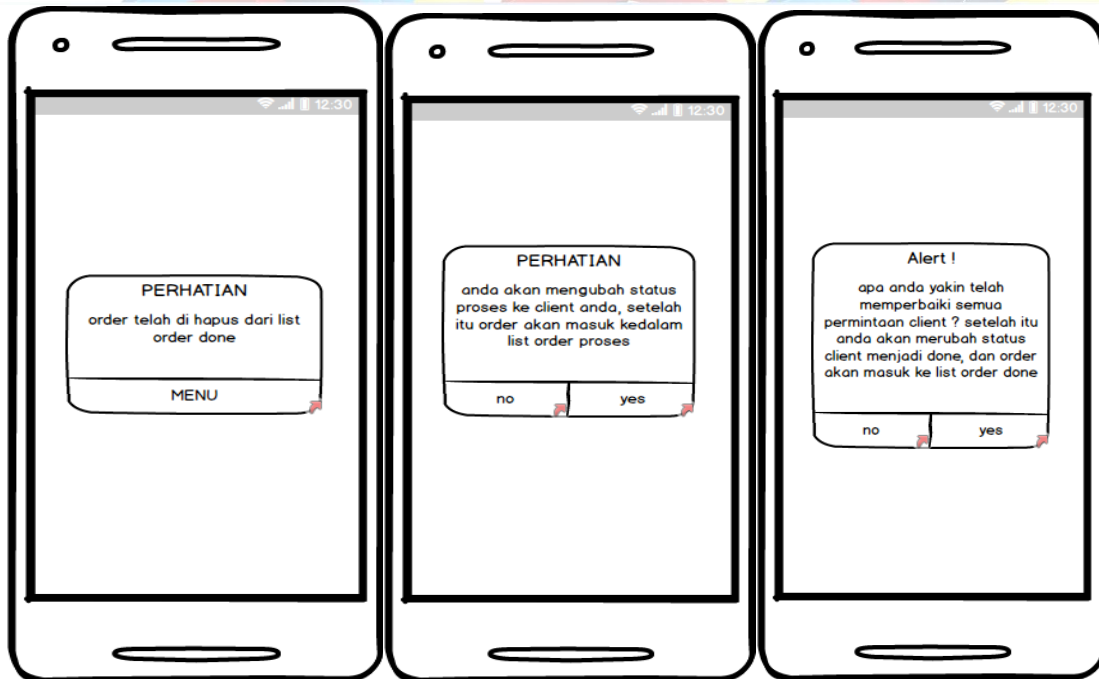
2. User View Pemilik Bengkel



XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	







XML Legal Document Utility	Version: <1.2>
Software Design Document	Date: 2017-05-30
SD-BengkelQ	

8 Kebutuhan Non-Fungsional

- Sistem memiliki jaminan keamanan pada semua data user.
- Sistem harus memiliki tingkat ketersediaan 699/700 atau 99%.
- Sistem dapat diakses oleh lebih dari 700 pengguna dalam waktu yang sama
- Sistem dapat menyimpan data sebesar 70 MB per hari
- Sistem dapat mengupdate dan menyinkronisasikan data yang dimasukkan
- Batas kapasitas yang dapat ditampung oleh sistem sebesar 10 MB per hari
- Sistem harus memiliki backup dari sistem

9 Dokumen Tambahan

1. Tools Used to Create Diagrams

a. UML Modeling Tools

Pembuatan Use Case Diagram, Robustness Diagram, Sequence Diagram dan Class Diagram menggunakan tools Star UML.

2. Tools Used to Craete GUI Story Board

a. GUI Tools

Pembuatan GUI Story Board menggunakan tools Balsamiq Mockups 3.