BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem yang Sedang Berjalan

Analisis merupakan suatu kegiatan yang bertujuan untuk mempelajari serta mengevaluasi suatu bentuk permasalahan yang ada pada sebuah sistem. Dalam analisis sistem akan ditemukan masalah yang mungkin akan mempengaruhi kerja sistem. Agar sistem yang dirancang dapat berjalan sebagaimana mestinya, perlu dilakukan analisis terhadap kinerja sistem yang pada akhirnya bertujuan untuk pengembangan sistem.

Adapun sistem yang sedang berjalan pada bengkel Ditech Injection sampai saat ini yaitu sistem informasi pengolahan data yang berjalan pada perusahaan tersebut masih ditulis secara manual. Pengarsipan data pelanggan, pengarsipan daftar suku cadang, hingga pada proses transaksi masih dilakukan pencatatan manual, baik itu transaksi penjualan pembelian suku cadang dan pelayanan perbaikan masih dicatat manual dengan proses penghitungan menggunakan kalkulator.

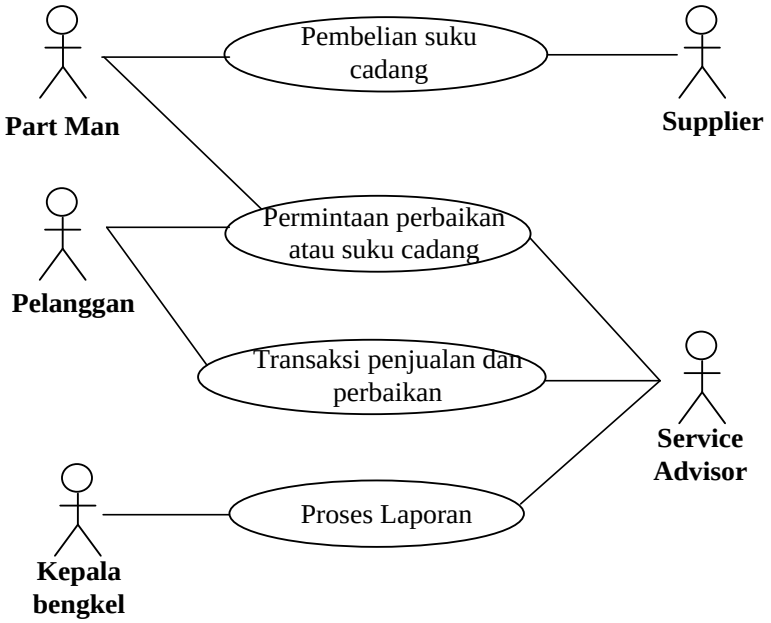
4.1.1 Analisis Prosedur yang Sedang Berjalan

Analisis prosedur merupakan proses analisa terhadap kegiatan-kegiatan yang dilakukan oleh bengkel Ditech Injection dalam proses pelayanan perbaikan dan penjualan pembelian suku cadang. Analisis terhadap prosedur yang sedang berjalan hendaknya perlu dilakukan terlebih dahulu sebelum dilakukannya

perancangan terhadap sistem informasi yang akan diusulkan. Harus diketahui hal-hal yang menjadi tujuan pemakai sehingga masalah tersebut dapat didefinisikan secara jelas.

4.1.1.1 *Use case* Diagram

Diagram *use case* bisnis memperlihatkan hubungan-hubungan antara aktor-aktor bisnis, *use case* bisnis, dan pekerja-pekerja bisnis untuk suatu organisasi atau perusahaan. Diagram ini memberikan model lengkap tentang apa yang dilakukan perusahaan, siapa yang ada di dalam perusahaan, serta siapa yang berada di luar perusahaan. Berikut adalah gambar model diagram *use case* bisnis yang berjalan :



Gambar 4. 1 *Use case* sistem yang sedang berjalan

1. Identifikasi *Use case*

   Identifikasi *use case* adalah *use case* yang bekerja pada sistem dan *use case* yang bekerja pada sistem adalah sebagai berikut :

Tabel 4. 1 Identifikasi *use case* sistem berjalan

Kode	*Use case*	Deskripsi
A	Pembelian suku cadang	Fungsionalitas untuk pembelian suku cadang ke *supplier*
B	Permintaan perbaikan dan suku cadang	Fungsionalitas untuk melakukan permintaan perbaikan suku cadang
C	Transaksi penjualan dan perbaikan	Fungsionalitas untuk melakukan transaksi permintaan perbaikan dan suku cadang
D	Proses Laporan	Fungsionalitas untuk proses laporan

2. Identifikasi Aktor

   Aktor yang berperan dalam sistem adalah sebagai berikut

Tabel 4. 2 Identifikasi aktor sistem berjalan

No	Aktor	Deskripsi
1	*Service Advisor*	Aktor yang berperan mengatur proses administrasi dan transaksi di bengkel
2	*Part Man*	Aktor yang berfungsi mengatur persediaan suku cadang.
3	Kepala bengkel	Aktor yang berfungsi memeriksa semua laporan dan bertanggung jawab penuh dengan seluruh kegiatan proses bisnis di bengkel
4	*Supplier*	Aktor yang berada dil luar sistem bengkel yang menjual suku cadang.

4.1.1.2 Skenario *Use case*

Skenario *Use case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Adapun tahapan dari skenario *use case* pada sistem yang sedang berjalan adalah sebagai berikut :

4.1.1.2.1 Skenario *use case* pembelian suku cadang

Berikut skenario use case pembelian suku cadang yang berjalan :

Tabel 4. 3 Skenario *use case* pembelian suku cadang

Identifikasi	
Judul	Pembelian suku cadang
Kode *Use case*	A
Deskripsi *Use case*	Melakukan pembelian untuk kebutuhan gudang
Aktor	*Part man*
Trigger	*Part man* meminta barang
Kondisi awal	*Part man* siap melakukan pembelian
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. *Part man* melakukan pembelian suku cadang dari *supplier* dan pembayaran dengan cara tunai	
	2. Menyimpan suku cadang di gudang

4.1.1.2.2 Skenario *use case* permintaan perbaikan atau suku cadang

Berikut skenario use case permintaan perbaikan atau suku cadang yang berjalan :

Tabel 4. 4 Skenario *use case* permintaan perbaikan atau suku cadang

Identifikasi	
Judul	Permintaan perbaikan atau suku cadang
Kode *Use case*	B
Deskripsi *Use case*	Pelanggan ingin melakukan perbaikan atau suku cadang
Aktor	Pelanggan, *service advisor, part man*
Trigger	Pelanggan datang untuk perbaikan atau membeli suku cadang
Kondisi awal	*Service advisor* siap menerima permintaan pelanggan
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. Pelanggan datang membeli suku cadang	
	2. *Service advisor* menerima permintaan perbaikan atau pembelian suku cadang
	3. *Service advisor* memeriksa persediaan suku cadang
	4. *Part man* mengkonfirmasi ketersediaan suku cadang

4.1.1.2.3 Skenario *use case* transaksi penjualan dan perbaikan

Berikut skenario use case transaksi penjualan dan perbaikan yang berjalan :

Tabel 4. 5 Skenario *use case* transaksi penjualan dan perbaikan

Identifikasi	
Judul	Transaksi penjualan dan perbaikan
Kode *Use case*	C
Deskripsi *Use case*	Transaksi perbaikan dan pembelian suku cadang
Aktor	*Service Advisor*
Trigger	Proses perbaikan dan pembelian suku cadang selesai
Kondisi awal	Menunggu proses perbaikan dan pembelian suku cadang selesai
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. *Service advisor* membuat nota perbaikan dan penjualan suku cadang	
	2. Menyimpan nota perbaikan dan penjualan suku cadang
3. Pelanggan melakukan pembayaran biaya perbaikan dan pembelian suku cadang dan menerima nota pembelian dan perbaikan	

4.1.1.2.4 Skenario *use case* proses laporan

Berikut skenario use case proses laporan yang berjalan :

Tabel 4. 6 Skenario *use case* proses laporan

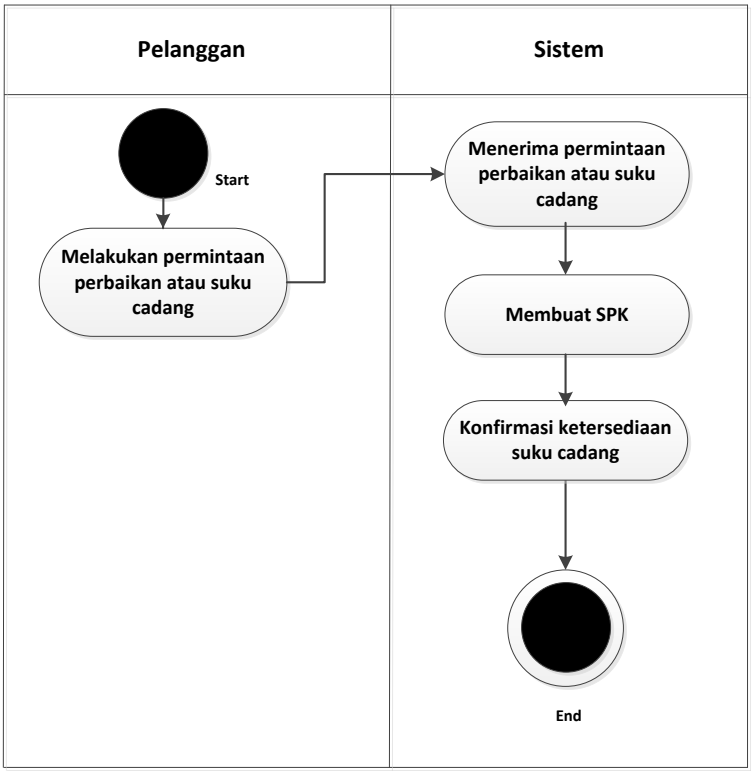
Identifikasi	
Judul	Proses laporan
Kode *Use case*	C
Deskripsi *Use case*	Membuat laporan penjualan, laporan pembelian, dan laporan perbaikan
Aktor	*Service advisor*
Trigger	Kepala bengkel membutuhkan laporan dari *service advisor*
Kondisi awal	*Service advisor* siap untuk membuat laporan
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. *Service advisor* membuat laporan	
2. *Service advisor* menyerahkan laporan kepada kepala bengkel	
	3. Kepala bengkel memeriksa laporan
	4. Kepala bengkel menyimpan arsip laporan

4.1.1.3 *Activity Diagram*

Activity diagram adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Berikut *activity diagram* proses perbaikan pada sistem yang berjalan :

4.1.1.3.1 *Activity diagram* permintaan perbaikan dan suku cadang

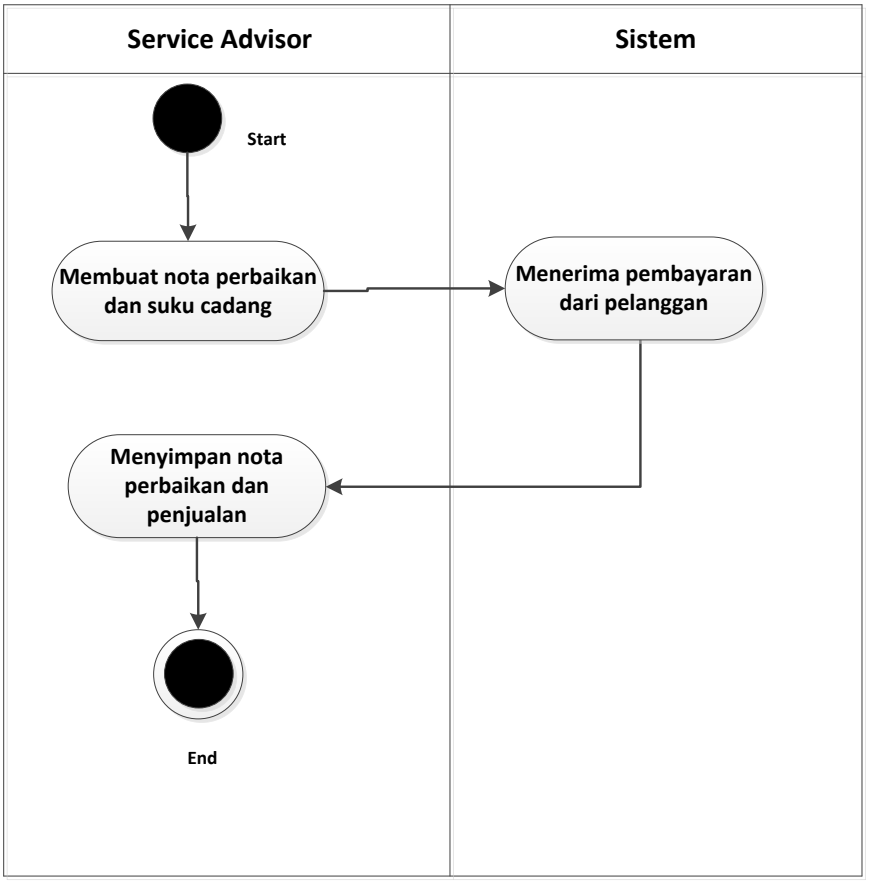
Activity diagram permintaan perbaikan dan suku cadang



Gambar 4. 2 *Activity diagram* permintaan perbaikan dan suku cadang

4.1.1.3.2 *Activity diagram* transaksi perbaikan dan penjualan

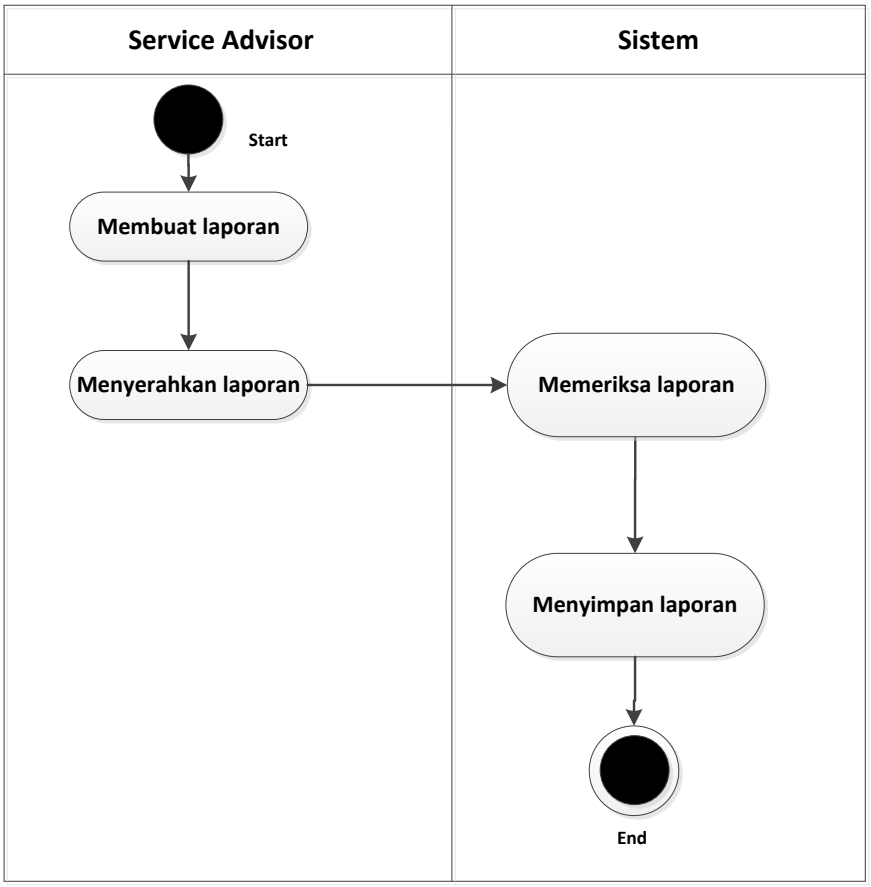
Activity diagram pada transaksi perbaikan dan penjualan



Gambar 4. 3 *Activity diagram* transaksi perbaikan dan penjualan

4.1.1.3.3 *Activity diagram* proses laporan

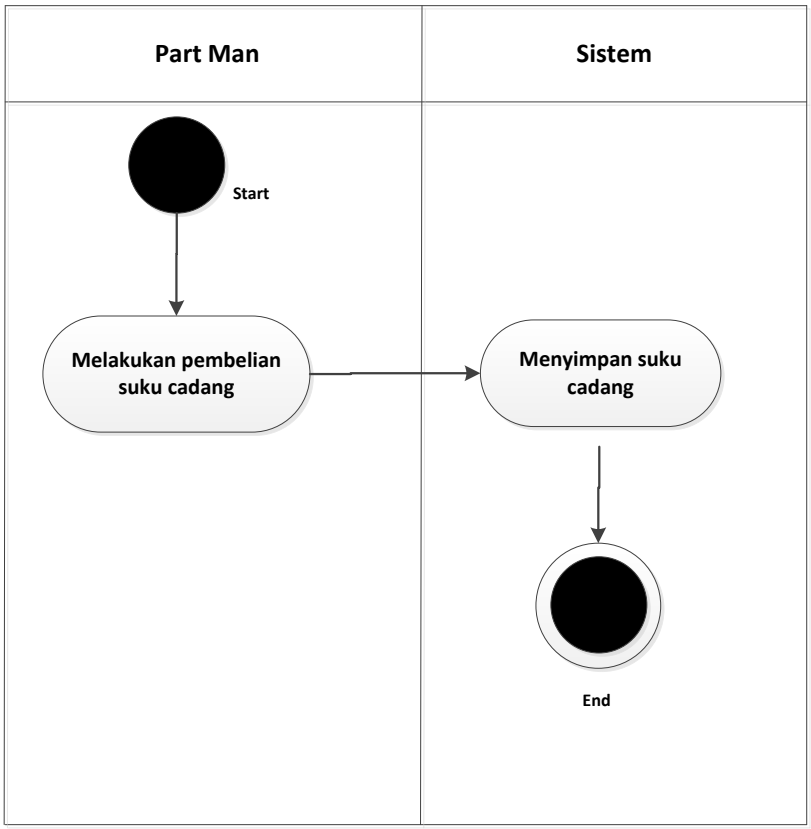
Activity diagram pada proses laporan :



Gambar 4. 4 *Activity diagram* proses laporan

4.1.1.3.4 *Activity Diagram* pembelian suku cadang

Activity diagram proses pembelian suku cadang :



Gambar 4. 5 *Activity diagram* pembelian suku cadang

4.1.2 Evaluasi Proses Bisnis yang Sedang Berjalan

Setelah dilakukan penelitian pada proses bisnis berjalan di bengkel Ditech Injection, terdapat beberapa kelemahan yang ditemukan pada proses yang sedang berjalan. Kelemahan-kelemahan tersebut adalah sebagai berikut:

Tabel 4. 7 Evaluasi proses bisnis yang berjalan

No	Permasalahan	Penyelesaian
1.	Proses pengelolaan dan penghitungan transaksi pelayanan perbaikan, penjualan pembelian suku cadang masih dicatat secara manual, sehingga sering terjadi kesalahan data transaksi dan membutuhkan waktu yang cukup lama.	Merancang aplikasi agar dapat memberikan kemudahan dalam melakukan proses penghitungan transaksi dengan cepat dan tepat.
2.	Kesulitan dalam pengambilan keputusan untuk menentukan garansi perbaikan yang diberikan oleh bengkel karena tidak terdapatnya *database* pelanggan dan riwayat perbaikan kendaraan sehingga terjadi kesalah pahaman antara pelanggan dan pihak bengkel.	Merancang dan membangun aplikasi yang memiliki dan mengelola *database* secara baik dan terintegrasi sehingga membantu dalam kegiatan proses bisnis dan pengambilan keputusan.
3.	Terdapat perbedaan harga yang cukup signifikan antara perkiraan biaya dan pembayaran di akhir karena belum adanya aplikasi untuk melakukan perkiraan biaya perbaikan secara tepat dan sesuai dengan pembayaran	Merancang dan membangun aplikasi yang dapat menghasilkan perkiraan biaya perbaikan mendekati tepat dengan pembayaran di akhir.

No	Permasalahan	Penyelesaian
4.	Lemahnya pengawasan persediaan suku cadang di gudang dan penyimpanan administrasi data persediaan masih menggunakan buku besar untuk media penyimpanannya, sehingga sering terjadi keterlambatan persediaan, dan membutuhkan waktu lama untuk melakukan pembelian ke *supplier*.	Merancang dan membangun aplikasi persediaan suku cadang yang dilengkapi dengan *database* yang dapat meningkatkan efisiensi waktu, dan meningkatkan produktifitas bengkel.
5.	Kepala bengkel merasa kesulitan dalam memeriksa arsip laporan yang dikarenakan tidak sesuainya data transaksi penjualan pelayanan perbaikan dan penjualan pembelian suku cadang serta sering terjadi kehilangan data dan dokumen karena penyimpanannya masih dalam bentuk arsip.	Merancang dan membangun aplikasi yang dapat memberikan data laporan seluruh kegiatan transaksi dan menyimpannya dalam *database* sehingga mudah untuk mencari dan memeriksanya.

4.2 Perancangan Sistem

Perancangan sistem merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan suatu sistem yang baru, atau memperbaharui sistem yang ada untuk meningkatkan efektifitas kerja agar dapat memenuhi hasil yang digunakan dengan tujuan memanfaatkan teknologi dan fasilitas yang tersedia. Pada bagian ini penyusun akan memberikan usulan yang merupakan sistem informasi secara komputerisasi yang diharapkan akan membantu dan mempermudah pekerjaan.

4.2.1 Tujuan Perancangan Sistem

Perancangan sistem bertujuan untuk memberikan gambaran yang jelas dan rancang bangun yang sesuai dengan kebutuhan pemakai sistem itu sendiri. Perancangan sistem dilakukan apabila tahap analisis sistem telah dilakukan. Maka untuk selanjutnya seorang analis sistem merancang bagaimana membentuk sistem yang baru ataupun memperbaharui sistem yang lama. Tahap inilah yang dinamakan dengan istilah dari perancangan sistem.

Adapun tujuan perancangan sistem yang diusulkan yaitu untuk menghasilkan suatu rancangan sistem yang dapat membantu proses pengolahan data dan transaksi dalam menangani proses penjualan pembelian suku cadang dan perbaikan yang semuanya terintegrasi dengan *database* dan aplikasi tersebut dapat mencapai sasaran kebutuhan dan tujuan dari perusahaan yaitu kepuasan pelanggan.

4.2.2 Gambaran Umum Sistem yang Diusulkan

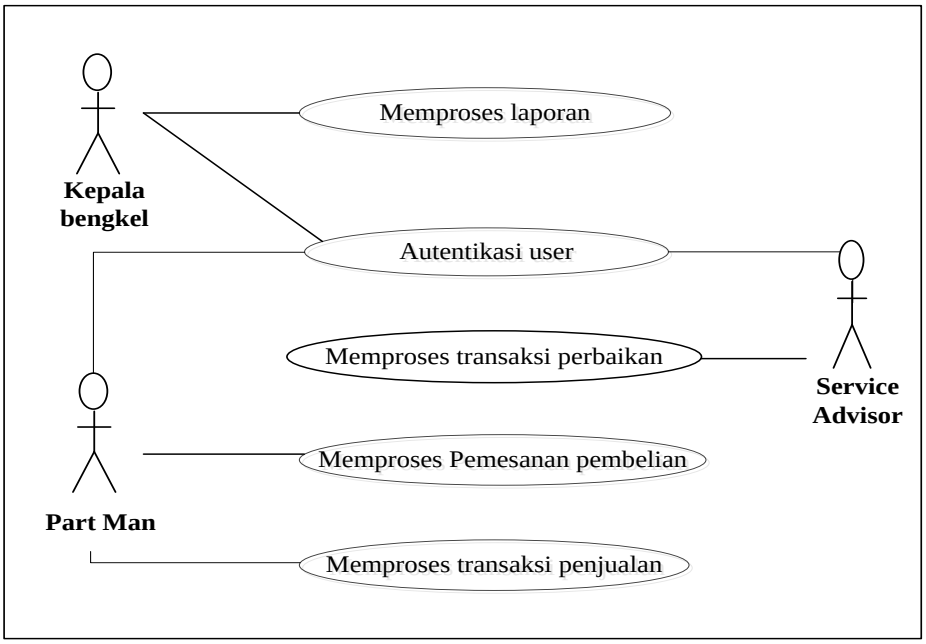
Gambaran umum tentang sistem yang diusulkan pada proses perancangan ini adalah suatu sistem yang merubah seluruh proses bisnis yang manual menjadi terkomputerisasi. Proses bisnis yang dimaksud adalah pengolahan data pelanggan, transaksi penjualan pembelian suku cadang, pelayanan perbaikan, hingga proses laporan kepada kepala bengkel ditangani oleh sistem informasi ini. Sistem ini diharapkan dapat membantu efektifitas dan optimalisasi waktu pada seluruh proses bisnis yang berjalan serta dapat membantu mengatasi permasalahan yang ada dan yang paling penting dapat menghasilkan informasi yang cepat, tepat dan akurat.

4.2.3 Perancangan Prosedur yang Diusulkan

Perancangan prosedur merupakan awal dari pembuatan sistem informasi yang akan dibuat, dimana dapat dilihat proses-proses apa saja yang nantinya diperlukan dalam pembuatan suatu sistem informasi. Sedangkan perancangan prosedur yang diusulkan merupakan tahap untuk memperbaiki atau meningkatkan efisiensi kerja. Tahap perancangan sistem yang digambarkan merupakan tahap perancangan untuk membangun suatu sistem dan mengkonfigurasikan komponen-komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem informasi yang baik. Sistem informasi yang dirancang tersebut menjadi satu komponen. Tahapan perancangan prosedur ini akan dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan *Unified Modeling Language* (UML).

4.2.3.1 *Use case* Diagram

Diagram *use case* bisnis memperlihatkan hubungan-hubungan antara aktor-aktor bisnis, *use case* bisnis, dan pekerja-pekerja bisnis untuk suatu organisasi atau perusahaan. Diagram ini memberikan model lengkap tentang apa yang dilakukan perusahaan, siapa yang ada di dalam perusahaan, serta siapa yang berada di luar perusahaan. Berikut adalah gambar model diagram *use case* bisnis yang diusulkan :



Gambar 4. 6 *Use case* diagram yang diusulkan

1. Identifikasi *Use case*

Identifikasi *use case* yang diusulkan adalah sebagai berikut :

Tabel 4. 8 Identifikasi *use case* sistem yang diusulkan

Kode	*Use case*	Deskripsi
A	Autentikasi *user*	Fungsionalitas untuk menentukan hak akses *user* pada saat *login* ke sistem
B	Memproses transaksi perbaikan	Fungsionalitas untuk mengolah data transaksi perbaikan
C	Memproses pemesanan pembelian	Fungsionalitas untuk melakukan penambahan data transaksi pemesanan pembelian
D	Memproses transaksi penjualan	Fungsionalitas untuk mengolah data transaksi penjualan
E	Memproses laporan	Fungsionalitas untuk melakukan pemeriksaan dan pencetakan laporan.

2. Identifikasi Aktor

Aktor yang berperan dalam menjalankan sistem adalah sebagai berikut :

Tabel 4. 9 Identifikasi aktor sistem diusulkan

No	Aktor	Deskripsi
1	*Service Advisor*	Aktor yang berperan mengatur proses administrasi dan transaksi perbaikan di bengkel
2	*Part Man*	Aktor yang berfungsi mengatur proses pemesanan pembelian, penjualan dan persediaan suku cadang
3	Kepala bengkel	Aktor yang berfungsi memeriksa semua laporan dan bertanggung jawab penuh dengan seluruh kegiatan proses bisnis di bengkel

4.2.3.2 Skenario *Use case*

Skenario *Use case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Adapun tahapan dari skenario *use case* pada sistem yang diusulkan adalah sebagai berikut :

4.2.3.2.1 Skenario *use case* autentikasi *user*

Berikut skenario *use case* autentikasi *user* yang diusulkan :

Tabel 4. 10 Skenario *use case* autentikasi *user*

Identifikasi	
Judul	Autentikasi *user*
Kode *Use case*	A
Deskripsi *use case*	Aktor login ke sistem untuk menentukan hak aksesnya
Aktor	*Service advisor, part man*, kepala bengkel
Trigger	Aktor menekan tombol login
Kondisi awal	Aplikasi menampilkan sebuah *form* untuk login
Skenario Utama	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi *field* kosong pada *form* login.	
2. Aktor menekan tombol login.	3. Sistem memeriksa validasi *user* berdasarkan *username* dan *password* pada *database*.
	4. Sistem mengkonfirmasi keabsahan *user*.
Kondisi akhir skenario utama	Sistem menampilkan halaman utama berdasarkan hak akses dari *user*
Skenario Alternatif - 1	
Aksi- aktor	**Reaksi sistem**
1. Aktor menekan tombol login tanpa mengisi *field-field* pada *form*.	2. *Form* login ditampilkan dan ada pesan kesalahan yang menyatakan *username* atau *password* harus diisi.
	3. *Form* utama tidak ditampilkan.
Kondisi akhir skenario alternatif -1	Aplikasi menampilkan *form login.*

4.2.3.2.2 Skenario *use case* memproses transaksi perbaikan

Berikut skenario *use case* memproses transaksi perbaian yang diusulkan :

Tabel 4. 11 Skenario *use case* memproses transaksi perbaikan

Identifikasi	
Judul	Memproses transaksi perbaikan
Kode *Use case*	B
Deskripsi *use case*	Aktor menambah data transaksi perbaikan
Aktor	*Service advisor*
Trigger	Aktor memilih menu untuk menambah transaksi perbaikan
Kondisi awal	Aplikasi menampilkan sebuah *form* transaksi perbaikan

Skenario Utama	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi data jenis perbaikan yang baru	
2. Aktor mengisi data transaksi perbaikan yang baru.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data telah diisi dan sesuai.
	4. Data transaksi perbaikan ditampilkan pada *form* dengan hasil kalkulasi.
	5. Sistem menambahkan data transaksi perbaikan yang baru pada *database*
	6. Sistem memperbaharui data suku cadang dengan data pada transaksi perbaikan jika ada pengeluaran suku cadang
	7. *Form* transaksi perbaikan dicetak untuk kemudian diprint

Kondisi akhir skenario utama	Sistem menyimpan data transaksi perbaikan yang baru pada *database* dan memperbarui data suku cadang
Skenario Alternatif – 1	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi data dengan menekan sebuah *button* di mana ada *field-field* pada *form* yang seharusnya diisi tetapi dikosongkan atau ada data yang tidak sesuai.	2. *Form* pengisian data transaksi perbaikan ditampilkan dengan pesan bahwa ada data *field* yang harus diisi namun masih kosong, atau ada data yang tidak sesuai.
	3. Data perbaikan baru tidak disimpan pada *database.*
Kondisi akhir skenario alternatif -1	*Form* penambahan data transaksi perbaikan ditampilkan

4.2.3.2.3 Skenario *use case* memproses pemesanan pembelian

Berikut skenario *use case* memproses pemesanan pembelian yang diusulkan :

Tabel 4. 12 Skenario *use case* memproses pemesanan pembelian

Identifikasi	
Judul	Memproses pemesanan pembelian
Kode *Use case*	C
Deskripsi *use case*	Aktor menambah data transaksi pemesanan pembelian
Aktor	*Part man*
Trigger	Aktor memilih menu untuk melakukan transaksi penambahan data pemesanan pembelian
Kondisi awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data pemesanan pembelian

Skenario Utama	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi *field* pada *form* yang data dari transaksi pemesanan pembelian yang ingin ditambahkan.	
2. Aktor mengisi data pemesanan pembelian yang baru.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data telah diisi dan sesuai.
	4. Data pemesanan pembelian ditambahkan atau dirubah pada *database*.
	5. Data suku cadang diperbaharui sesuai dengan data pembelian.
Kondisi akhir skenario utama	Data pemesanan pembelian dan suku cadang ditambahkan atau dirubah pada *database*
Skenario Alternatif – 1	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi data dengan menekan sebuah *button* di mana ada *field-field* pada *form* yang seharusnya diisi tetapi dikosongkan atau ada data yang tidak sesuai.	2. *Form* pengisian data pemesanan pembelian ditampilkan dengan pesan bahwa ada data *field* yang harus diisi namun masih kosong, atau ada data yang tidak sesuai.
	3. Data pembelian baru tidak disimpan pada *database.*
Kondisi akhir skenario alternatif -1	*Form* penambahan data pemesanan pembelian ditampilkan

4.2.3.2.4 Skenario *use case* memproses transaksi penjualan

Berikut skenario *use case* memproses transaksi penjualan yang diusulkan :

Tabel 4. 13 Skenario *use case* memproses transaksi penjualan

Identifikasi	
Judul	Memproses transaksi penjualan
Kode *Use case*	D
Deskripsi *use case*	Aktor menambah data transaksi penjualan.
Aktor	*Service dvisor*
Trigger	Aktor memilih menu untuk menambah transaksi penjualan.
Kondisi awal	Aplikasi menampilkan *form* transaksi penjualan.
Skenario Utama	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi data penjualan yang baru.	
2. Aktor mengisi data penjualan yang baru diisi	3. Sistem memeriksa apakah *field-field* data yang diperlukan telah diisi dan sesuai.
	4. Data transaksi penjualan ditampilkan pada *form* dengan hasil kalkulasi
	5. Sistem menambahkan data transaksi penjualan yang baru pada *database*.
	6. Sistem memperbaharui data suku cadang sesuai dengan data pada transaksi penjualan
	7. *Form* transaksi penjualan dicetak untuk kemudian diprint
Kondisi akhir skenario utama	Sistem menyimpan data transaksi yang baru pada *database* dan memperbaharui data suku cadang

Skenario Alternatif – 1	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi data dengan menekan sebuah *button* di mana ada *field-field* pada *form* yang seharusnya diisi tetapi dikosongkan atau ada data yang tidak sesuai.	2. *Form* pengisian data penjualanditampilkan dengan pesan bahwa ada data *field* yang harus diisi namun masih kosong, atau ada data yang tidak sesuai.
	3. Data penjualan baru tidak disimpan pada *database.*
Kondisi akhir skenario alternatif -1	*Form* penambahan data penjualan ditampilkan

4.2.3.2.5 Skenario *use case* memproses laporan

Berikut skenario *use case* memproses laporan yang diusulkan :

Tabel 4. 14 Skenario *use case* memproses laporan

Identifikasi	
Judul	Memproses laporan
Kode *Use case*	E
Deskripsi *use case*	Aktor mencetak laporan yang terjadi pada range waktu yang diinginkan.
Aktor	Kepala bengkel.
Trigger	Aktor memilih menu untuk mencetak laporan.
Kondisi awal	Aplikasi menampilkan *form* untuk menentukan periode dan jenis laporan yang diinginkan.

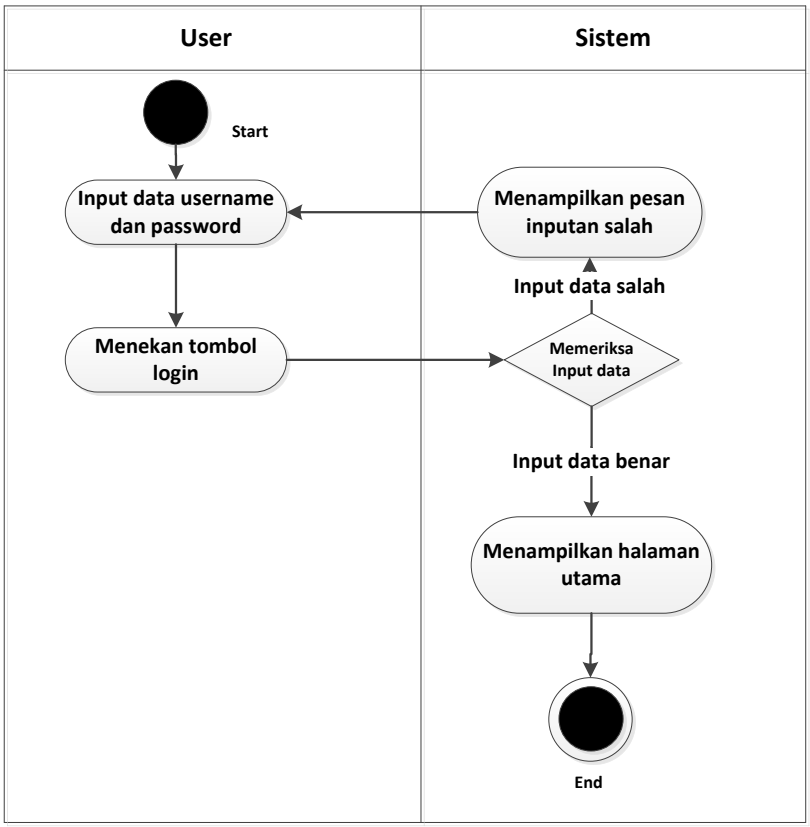
Skenario Utama	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi periode waktu dan jenis laporan yang diinginkan untuk dicetak.	
2. Aktor menekan tombol untuk mencetak laporan	
	3. Sistem mencetak laporan berdasarkan sesuai dengan jenis dan periode laporan yang dipilih.
Kondisi akhir skenario utama	Laporan dicetak
Skenario Alternatif – 1	
Aksi- aktor	**Reaksi sistem**
1. Aktor memilih periode yang tidak memiliki data penjualan samasekali.	
2. Aktor menekan tombol untuk mencetak laporan	
	3. Sistem menampilkan pesan bahwa transaksi pada periode tersebut kosong.
Kondisi akhir skenario alternatif -1	Sistem menampilkan *form* untuk melakukan pengisian periode laporan yang diinginkan.

4.2.3.3 *Activity Diagram*

Activity diagram (diagram aktivitas) adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Berikut *activity diagram* autentikasi *user* pada sistem yang diusulkan :

4.2.3.3.1 *Activity Diagram* autentikasi *user*

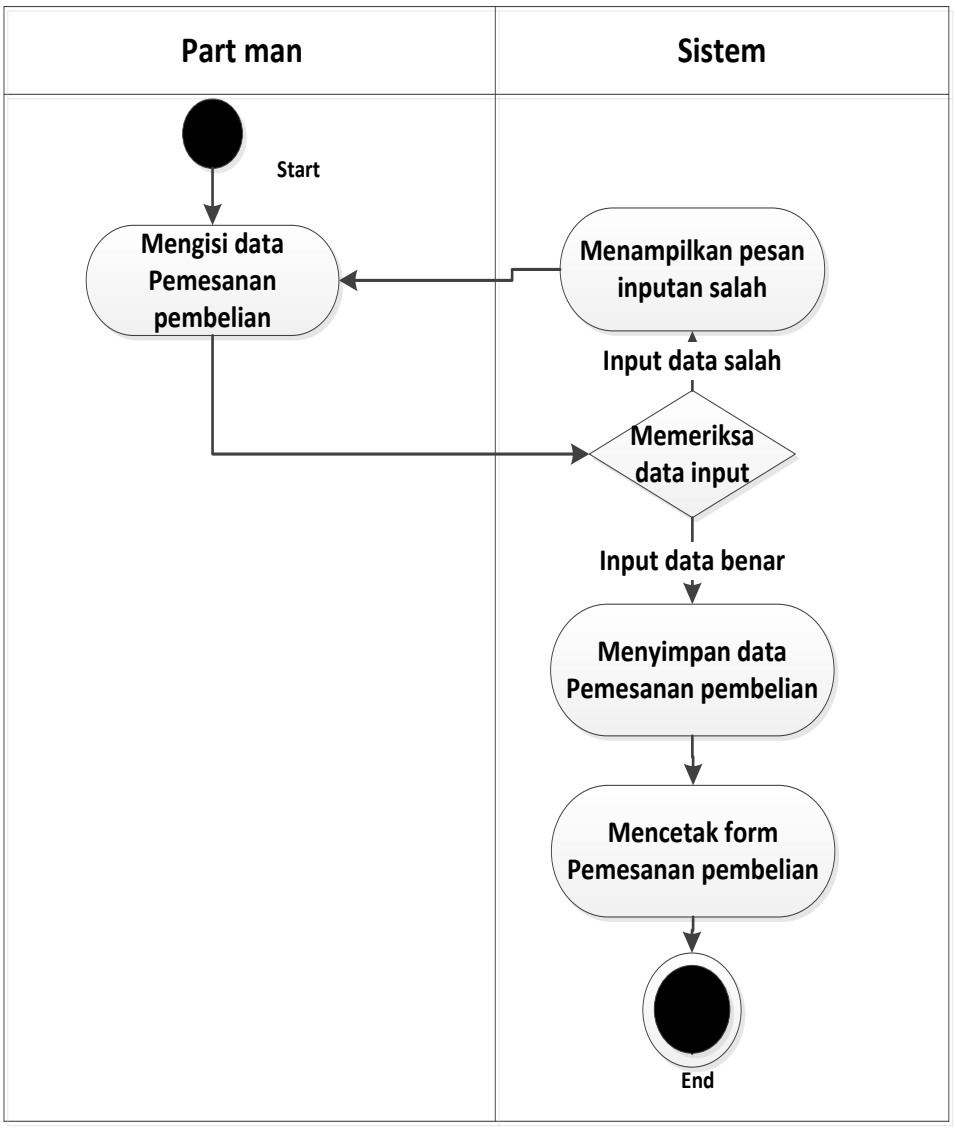
Berikut *activity diagram* autentikasi *user* yang diusulkan :



Gambar 4. 7 *Activity diagram* autentikasi *user*

4.2.3.3.2 *Activity Diagram* memproses pemesanan pembelian

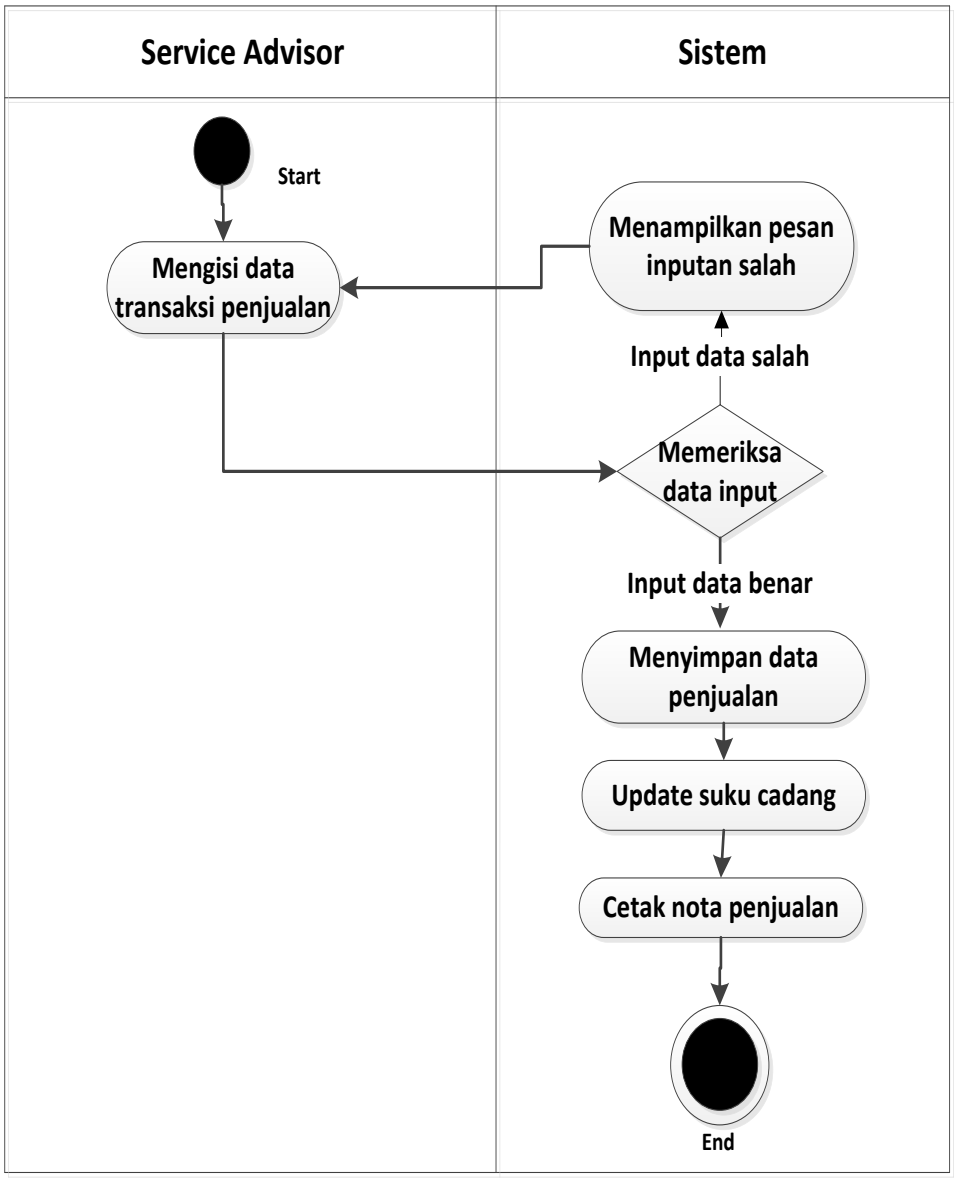
Berikut *activity diagram* memproses pemesanan pembelian yang diusulkan :



Gambar 4. 8 *Activity diagram* memproses pemesanan pembelian

4.2.3.3.3 *Activity Diagram* memproses transaksi penjualan

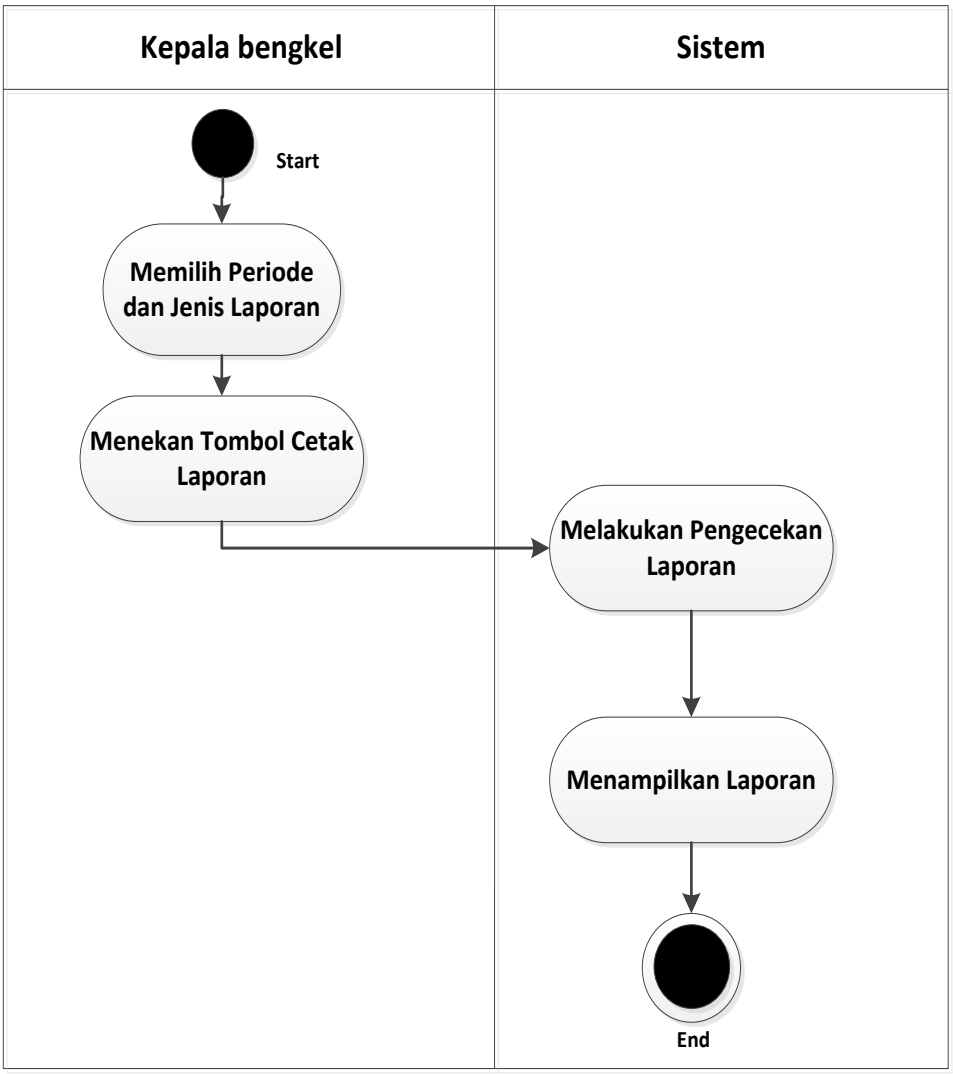
Berikut *activity diagram* memproses transaksi penjualan yang diusulkan :



Gambar 4. 9 *Activity diagram* memproses transaksi penjualan

4.2.3.3.4 *Activity Diagram* memproses laporan

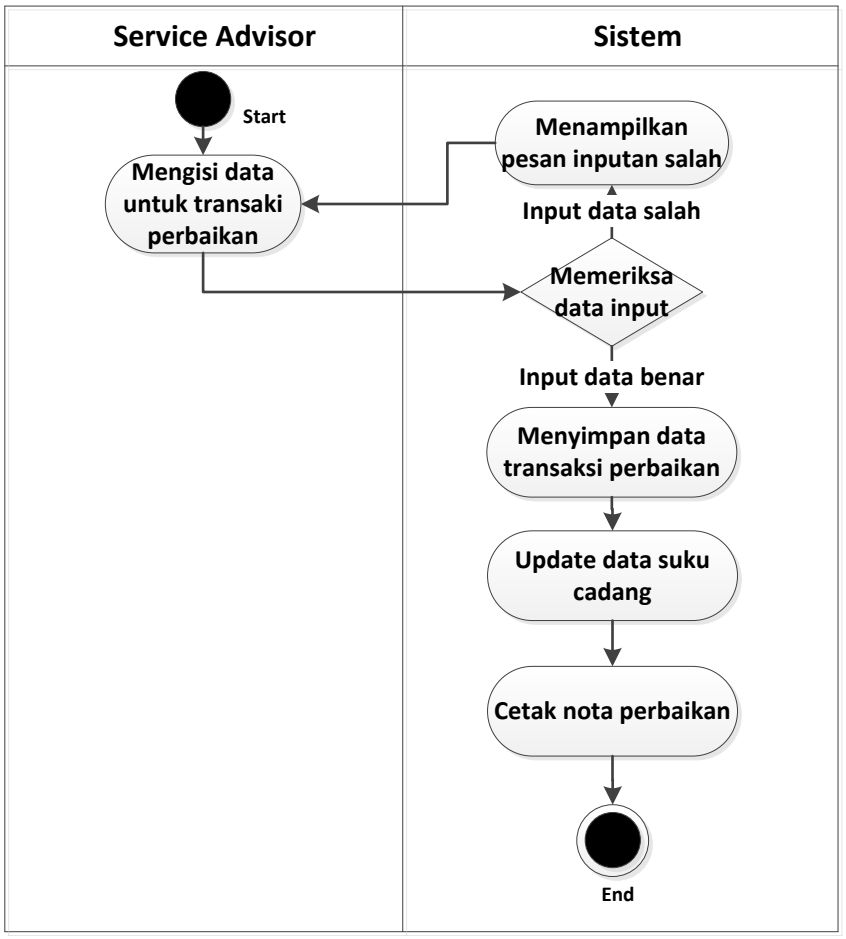
Berikut *activity diagram* memproses laporan yang diusulkan :



Gambar 4. 10 *Activity diagram* memproses laporan

4.2.3.3.5 *Activity Diagram* memproses transaksi perbaikan

Berikut *activity diagram* memproses transaksi perbaikan yang diusulkan :



Gambar 4. 11 *Activity diagram* memproses transaksi perbaikan

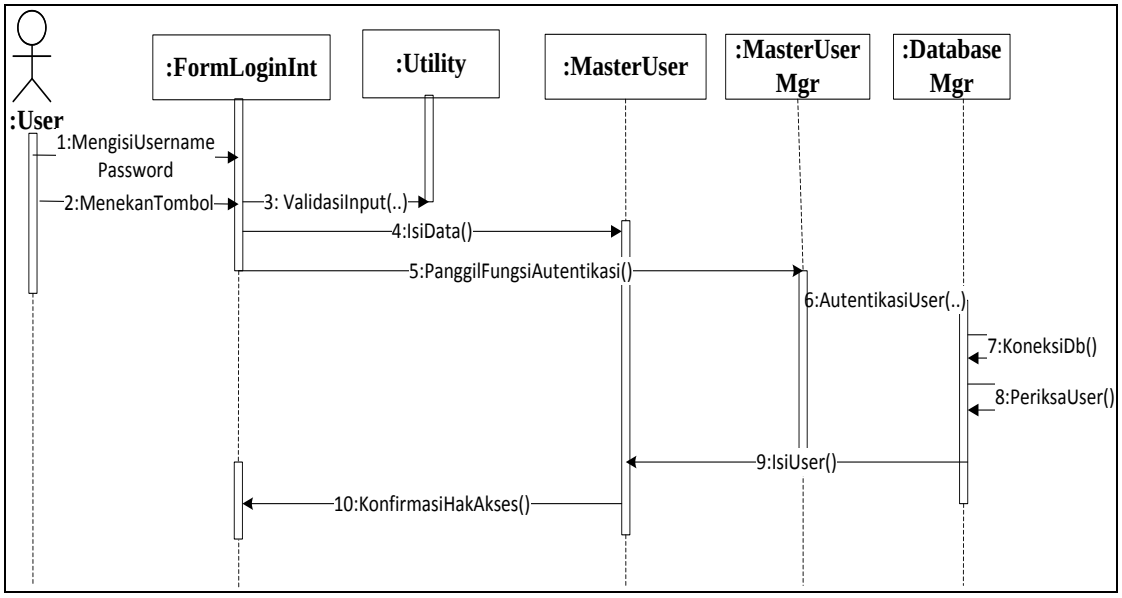
4.2.3.4 *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya *sequence diagram* digunakan dalam lapisan abstraksi model objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antar object, juga interaksi antar objek, dan menunjukkan sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama *sequence diagram* terdiri atas objek yang dituliskan dengan kotak segiempat

bernama, pesan diwakili oleh garis dengan tanda panah, dan waktu yang ditunjukkan dengan proses vertikal. Berikut adalah *sequence diagram* yang diusulkan :

4.2.3.4.1 *Sequence Diagram* autentikasi *user*

Berikut *sequence diagram* autentikasi *user* yang diusulkan :



Gambar 4. 12 *Sequence diagram* autentikasi *user*

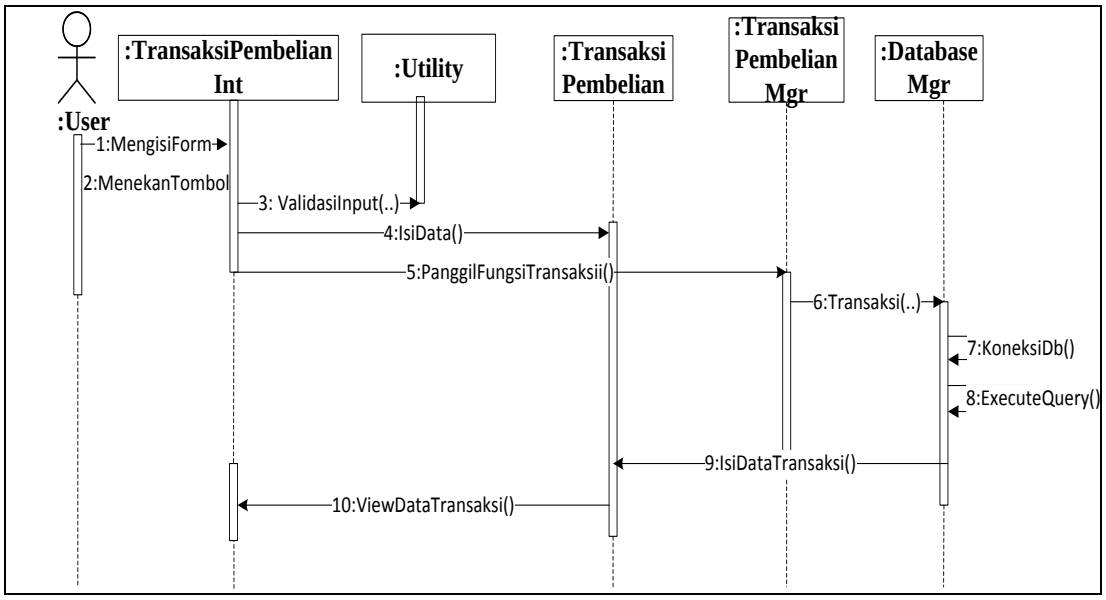
Keterangan:

Pada gambar 4.12 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. FormLoginInt
2. Utility
3. MasterUser
4. MasterUserMgr
5. DatabaseMgr

4.2.3.4.2 *Sequence Diagram* memproses pemesanan pembelian

Berikut *sequence diagram* memproses pemesanan pembelian yang diusulkan :



Gambar 4. 13 *Sequence diagram* memproses pemesanan pembelian

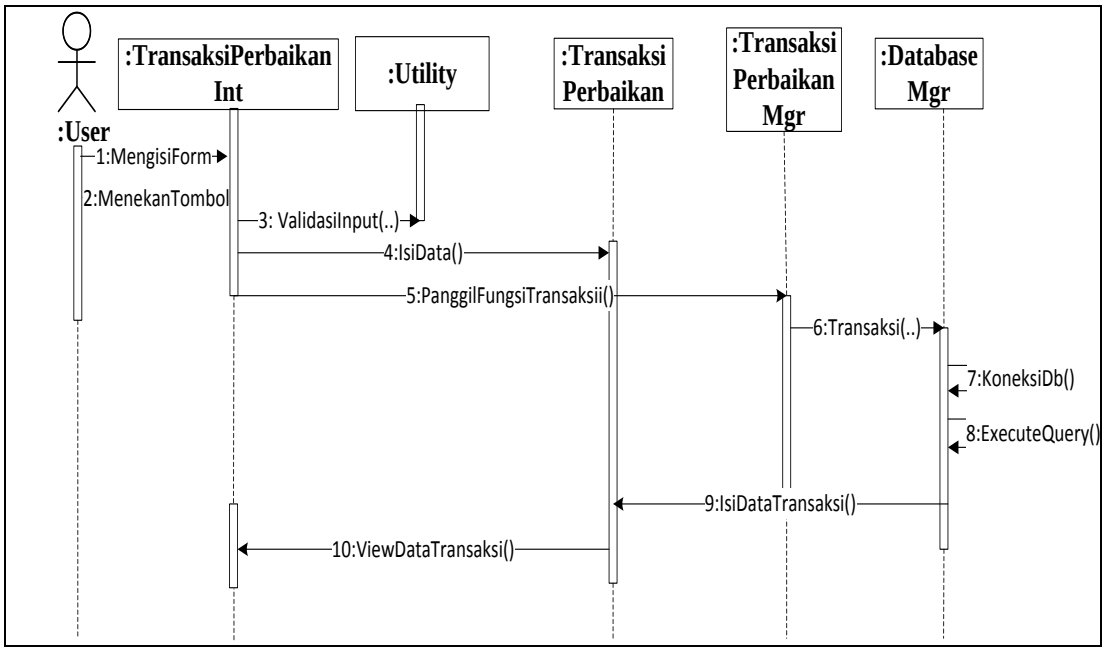
Keterangan:

Pada gambar 4.13 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. TransaksiPembelianInt
2. Utility
3. TransaksiPembelian
4. TransaksiPembelianMgr
5. DatabaseManager

4.2.3.4.3 *Sequence Diagram* memproses transaksi perbaikan

Berikut *sequence diagram* memproses transaksi perbaikan yang diusulkan :



Gambar 4. 14 *Sequence diagram* memproses transaksi perbaikan

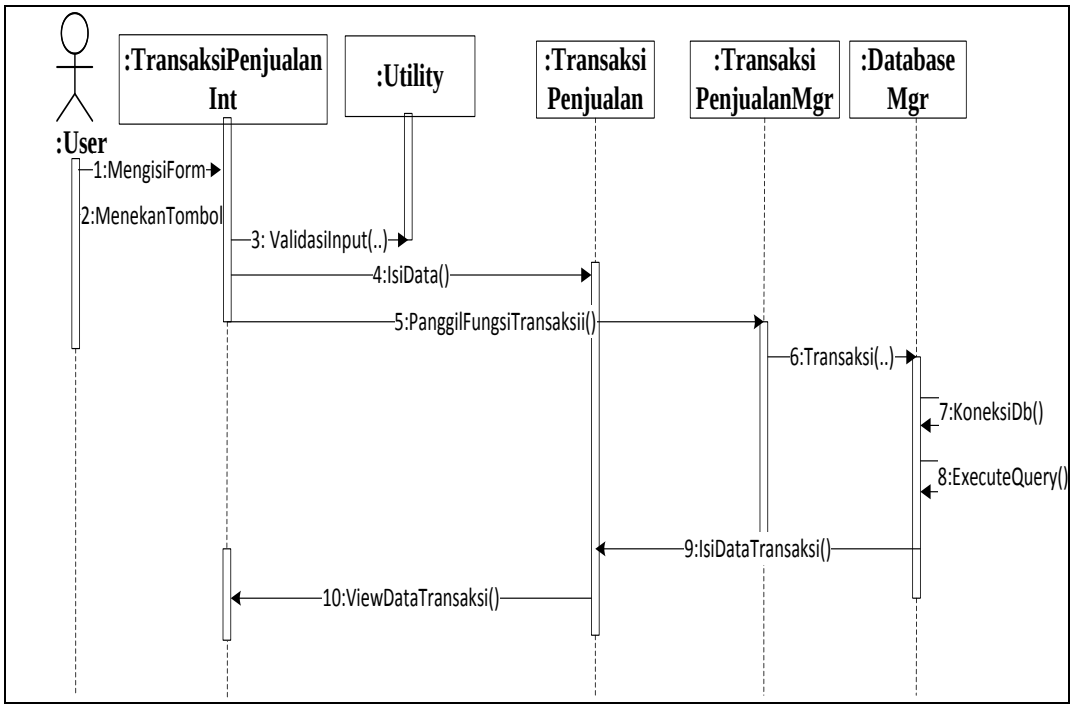
Keterangan:

Pada gambar 4.14 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. TransaksiPerbaikanInt
2. Utility
3. TransaksiPerbaikan
4. TransaksiPerbaikanMgr
5. DatabaseMgr

4.2.3.4.4 *Sequence Diagram* memproses transaksi penjualan

Berikut *sequence diagram* memproses transaksi penjualan yang diusulkan :



Gambar 4. 15 *Sequence diagram* memproses transaksi penjualan

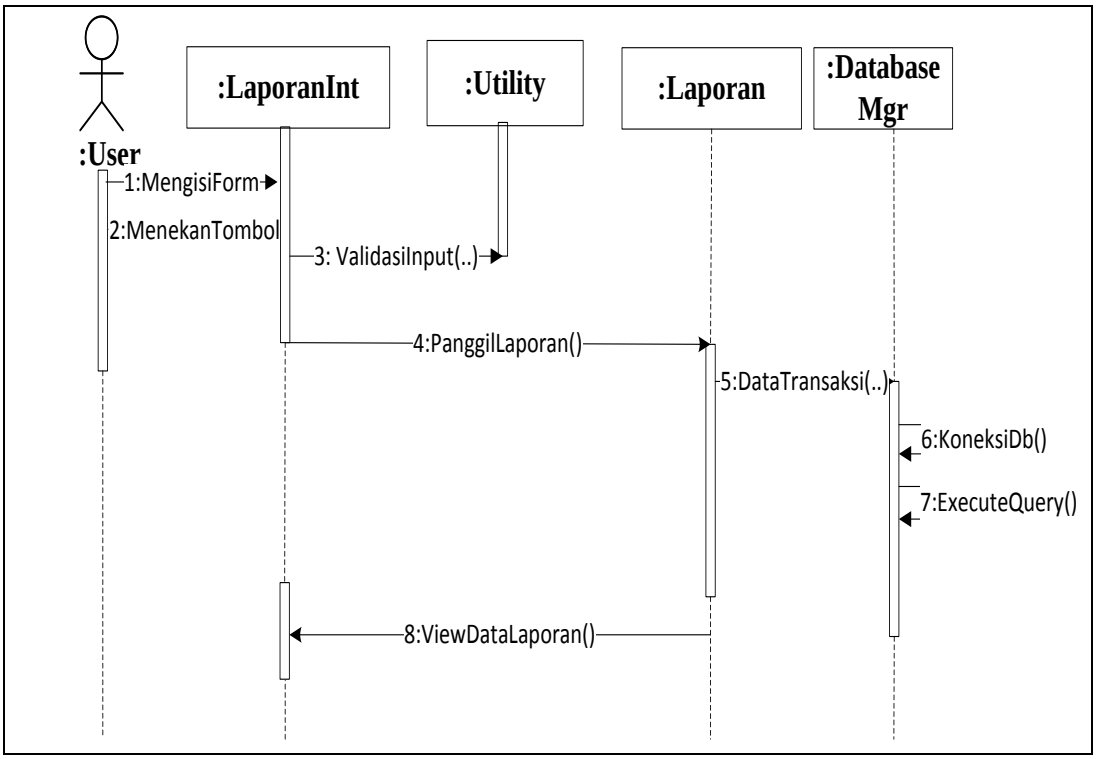
Keterangan:

Pada gambar 4.15 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. TransaksiPenjualanInt
2. Utility
3. TransaksiPenjualan
4. TransaksiPenjualanMgr
5. DatabaseMgr

4.2.3.4.5 *Sequence Diagram* memproses laporan

Berikut *sequence diagram* memproses laporan yang diusulkan :



Gambar 4. 16 *Sequence diagram* memproses laporan

Keterangan:

Pada Gambar 4.16 dapat dilihat bahwa ada 4 kelas yang saling berinteraksi, yakni:

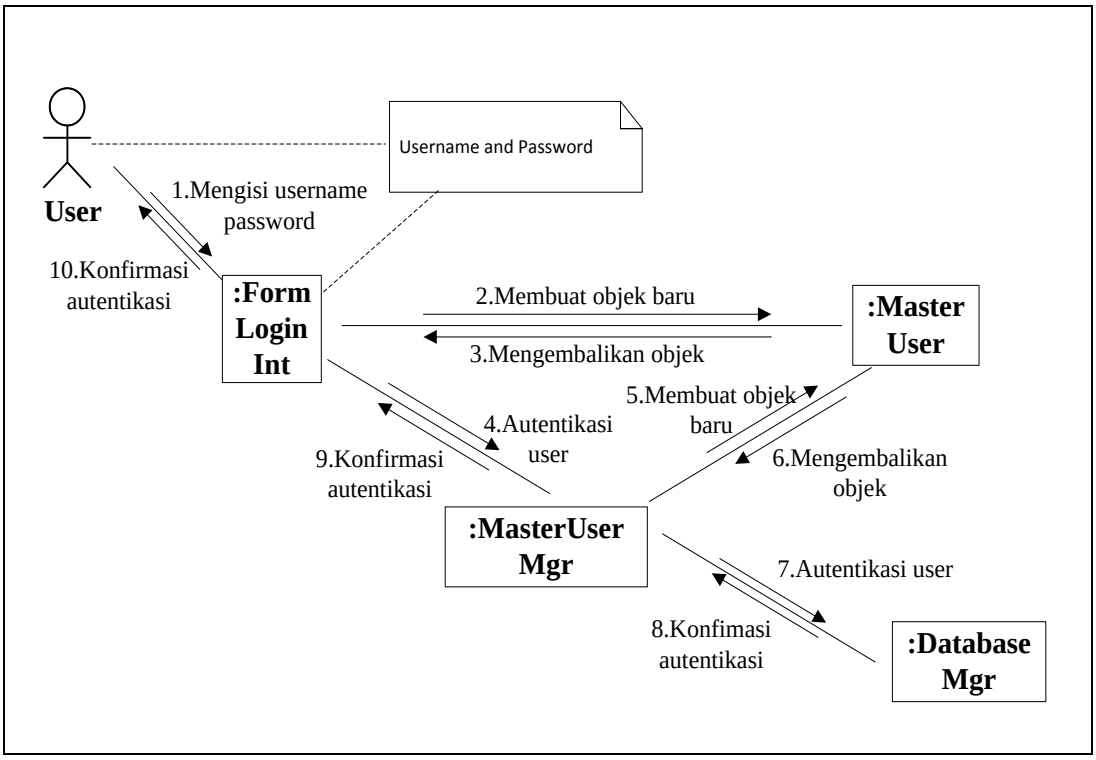
1. LaporanInt
2. Utility
3. Laporan
4. DatabaseMgr

4.2.3.5 *Collaboration Diagram*

Collaboration diagram adalah suatu diagram yang memperlihatkan / menampilkan pengorganisasian interaksi yang terdapat disekitar objek (seperti halnya *sequence diagram*) dan hubungannya terhadap yang lainnya. *Collaboration diagram* lebih menekankan kepada peran setiap objek dan bukan pada waktu penyampaian pesan/*message*. Berikut ini adalah *collaboration diagram* yang dibutuhkan :

4.2.3.5.1 *Collaboration Diagram* autentikasi *user*

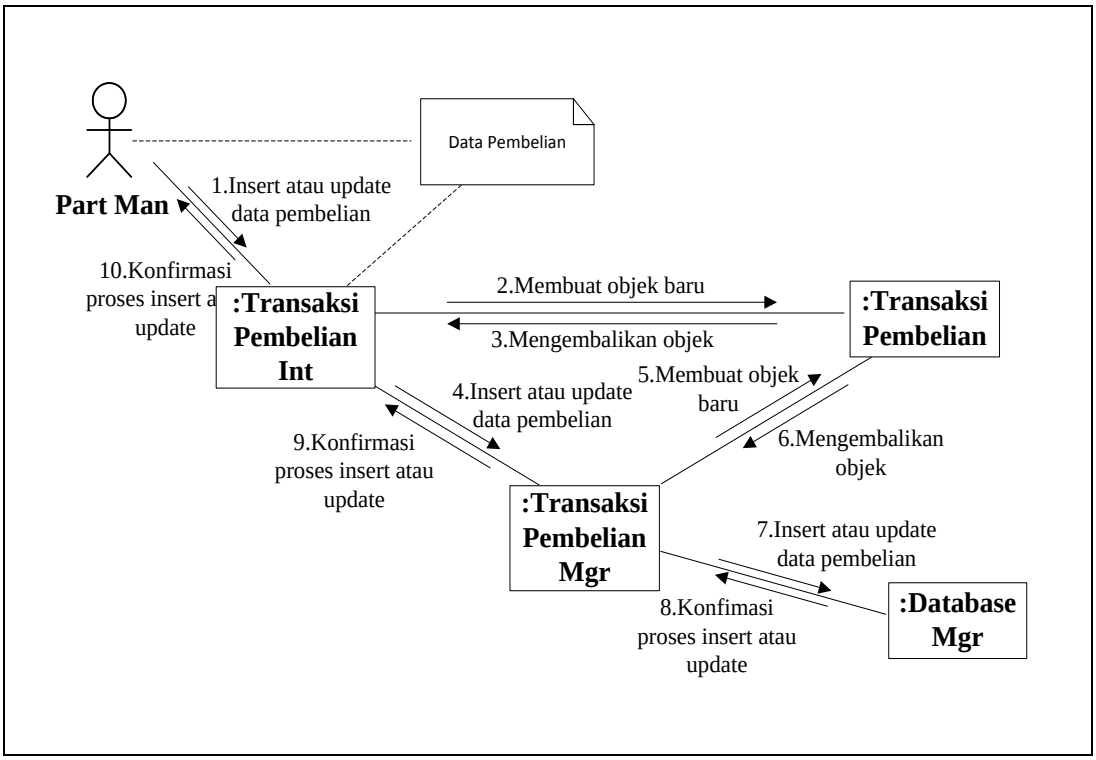
Berikut *collaboration diagram* autentikasi *user* yang diusulkan :



Gambar 4. 17 *Collaboration diagram* autentikasi *user*

4.2.3.5.2 *Collaboration Diagram* memproses pemesanan pembelian

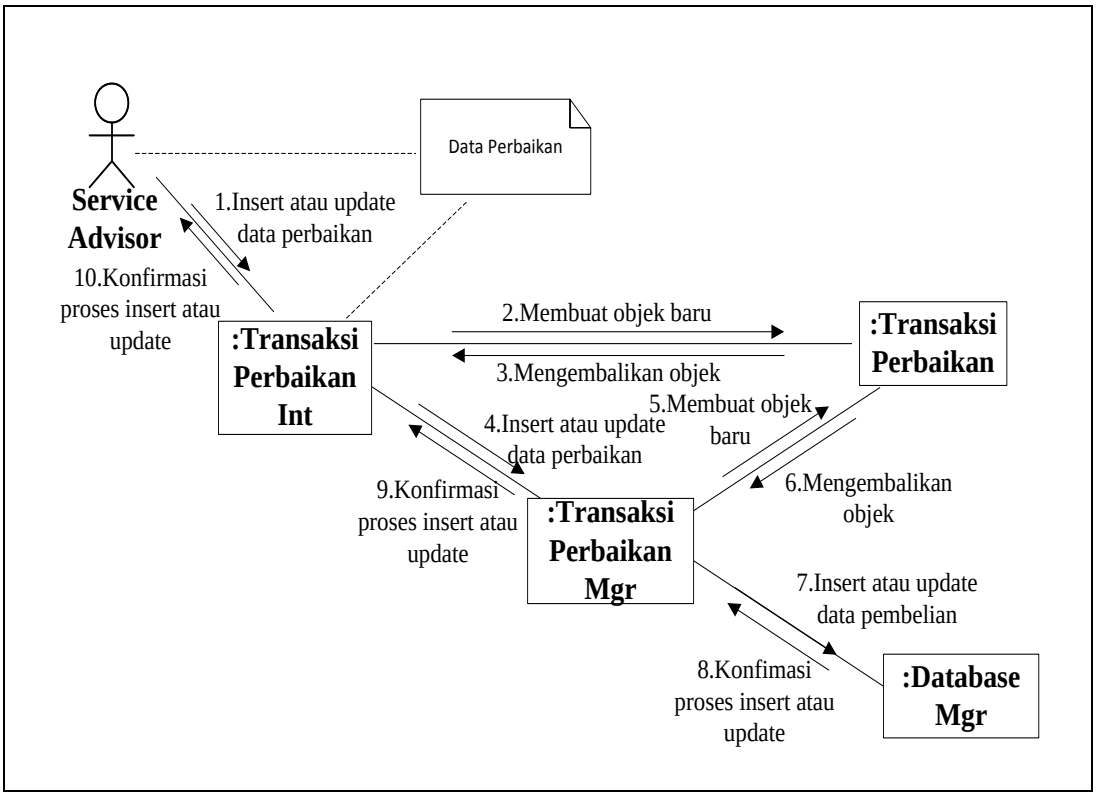
Berikut *collaboration diagram* memproses pemesanan pembelian yang diusulkan



Gambar 4. 18 *Collaboration diagram* memproses pemesanan pembelian

4.2.3.5.3 *Collaboration Diagram* memproses transaksi perbaikan

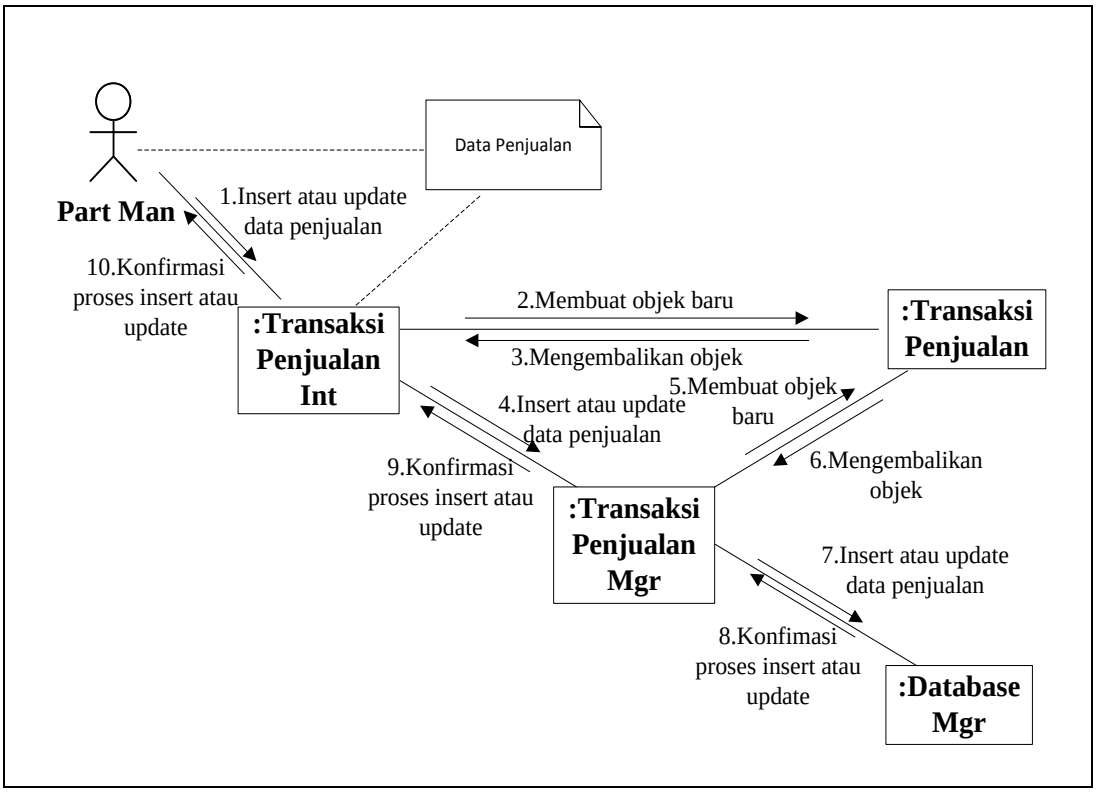
Berikut *collaboration diagram* memproses transaksi perbaikan yang diusulkan



Gambar 4. 19 *Collaboration diagram* memproses transaksi perbaikan

4.2.3.5.4 *Collaboration Diagram* memproses transaksi penjualan

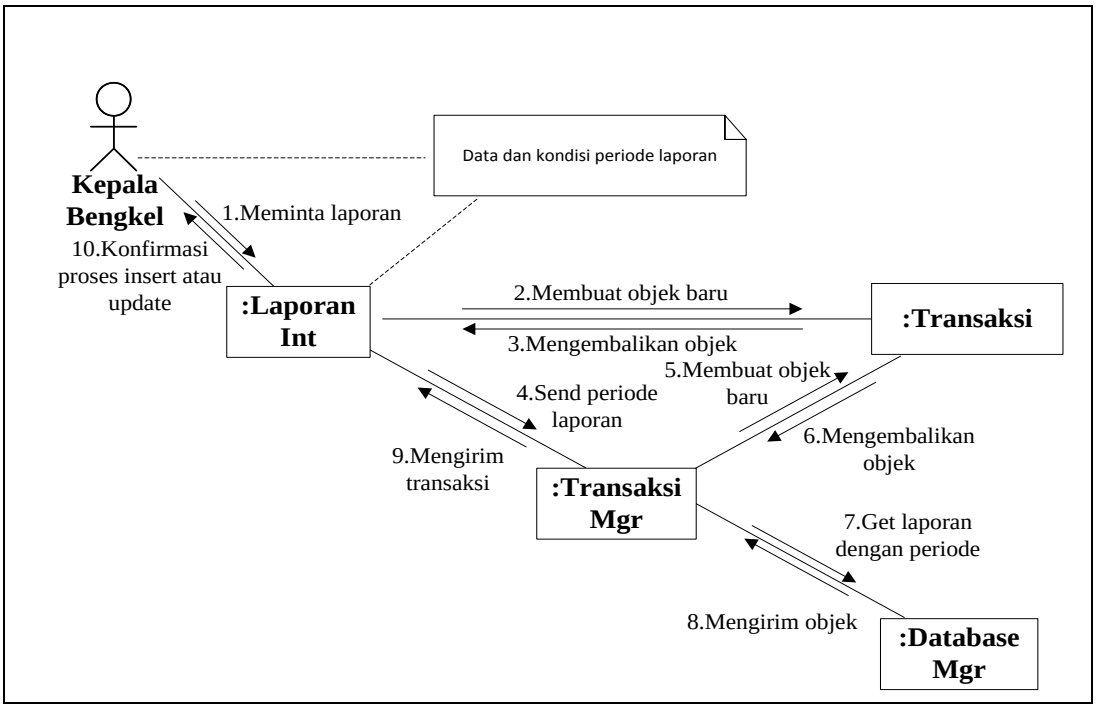
Berikut *collaboration diagram* memproses transaksi penjualan yang diusulkan



Gambar 4. 20 *Collaboration diagram* memproses transaksi penjualan

4.2.3.5.5 *Collaboration Diagram* memproses laporan

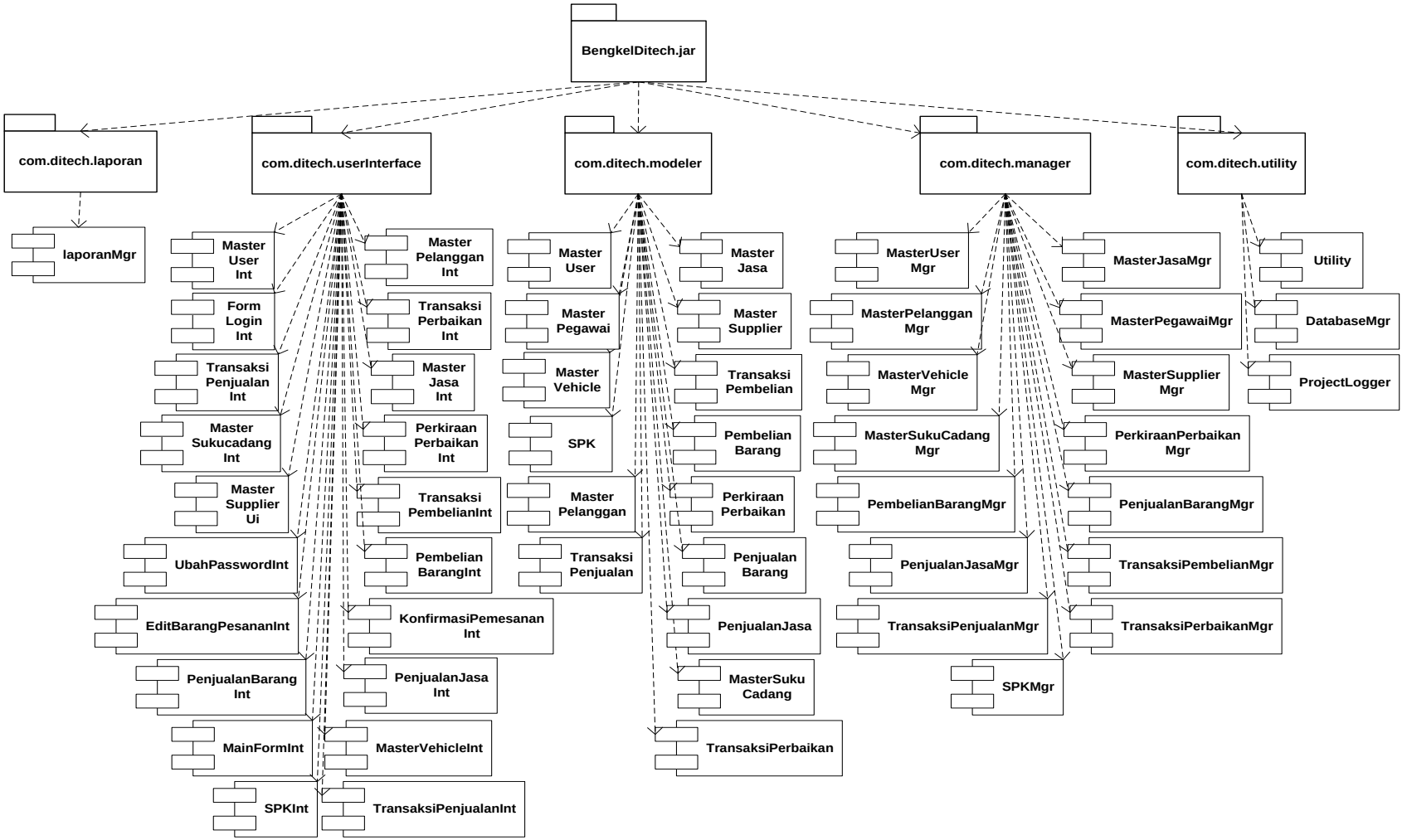
Berikut *collaboration diagram* memproses laporan yang diusulkan



Gambar 4. 21 *Collaboration diagram* memproses laporan

4.2.3.6 *Component Diagram*

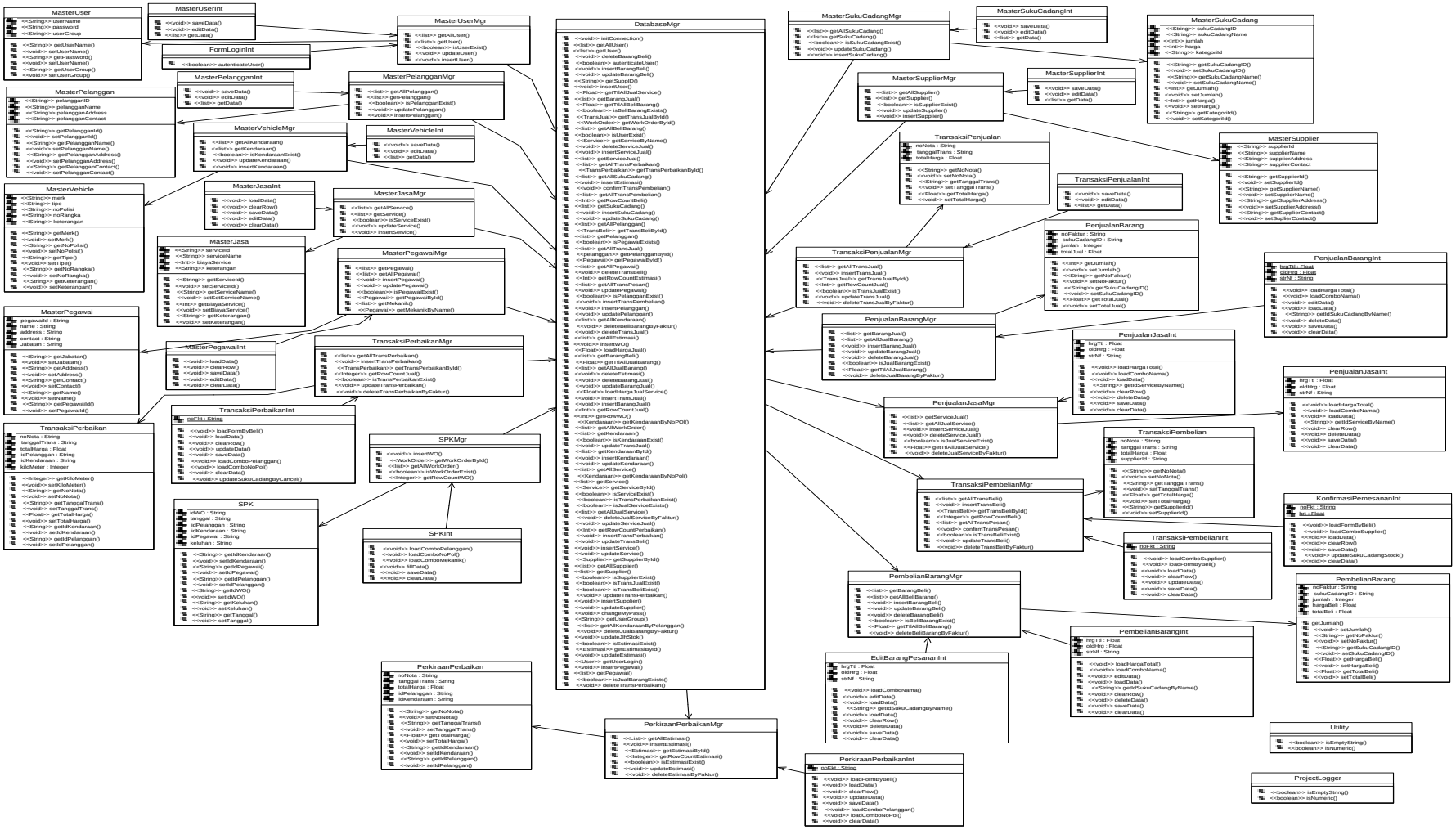
Component diagram menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan (*dependency*) di antaranya. *Component* piranti lunak adalah modul berisi *code*, baik *berisi source code* maupun *binary code*, baik *library* maupun *executable*, baik yang muncul pada *compile time, link time,* maupun *run time.* Umumnya komponen terbentuk dari beberapa *class* dan *package*, tapi dapat juga dari komponen-komponen yang lebih kecil. Berikut ini adalah *component diagram* yang dibutuhkan :



Gambar 4. 22 *Component diagram*

4.2.3.7 *Class Diagram*

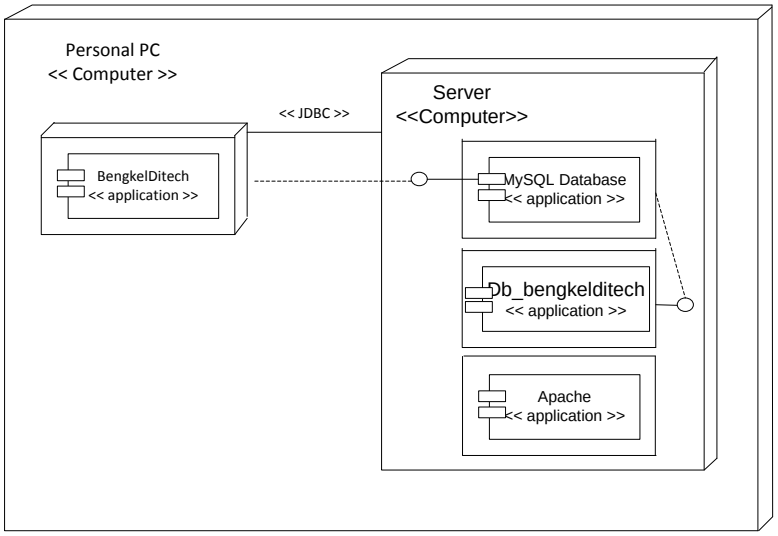
Menggambarkan struktur statis *class* di dalam sistem. *Class* merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem pemasaran produk dari bagian penjualan beserta proses-proses yang terjadi, maka dapat dibuat *class diagram* Berikut *class diagram* yang diusulkan pada bengkel Ditech Injection:



Gambar 4. 23 ***Class diagram***

4.2.3.8 *Deployment Diagram*

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. *Deployment diagram* berhubungan dengan diagram komponen dimana *deployment diagram* memuat satu atau lebih komponen-komponen.



Gambar 4. 24 *Deployment diagram* sistem yang diusulkan

4.2.4 Perancangan Antar Muka

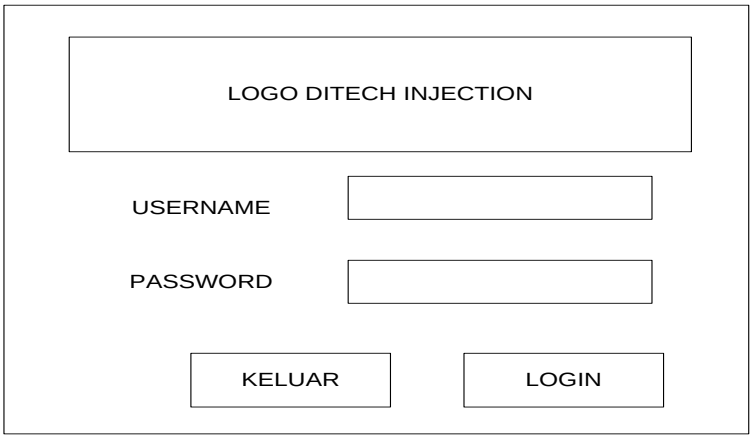
Perancangan antar muka dirancang untuk memudahkan pemakai (*user*) dalam mengakses informasi-informasi yang dibutuhkan. Perancangan antar muka meliputi perancangan tampilan input dan output. Adapun perancangan antar muka sistem informasi penjualan pembelian suku cadang dan perbaikan pada bengkel Ditech Injection adalah sebagai berikut :

4.2.4.1 Perancangan Input

Perancangan input merupakan salah satu bentuk tampilan untuk *user* berinteraksi dengan program. Adapun perancarang input yang diusulkan adalah sebagai berikut :

1. Rancangan *login user*

   Halaman ini merupakan halaman login *user*, sebelum masuk dan mengakses ke sistem terlebih dahulu *user* melakukan login. Disini terdapat 3 *user* yang dapat menggunakan sistem informasi ini yaitu kepala bengkel, *service advisor* dan *part man*. Ketiga *user* tersebut memiliki hak akses yang berbeda-beda sesuai dengan proses bisnis yang dirancang.



Gambar 4. 25 Desain tampilan untuk *user* login

Keterangan :

a. *Textbox Username* dan *password* digunakan untuk mengisi data dari *user*.
b. *Button* Login digunakan untuk melakukan proses masuk ke dalam sistem sesuai dengan hak akses masing-masing *user*.

2. Rancangan *form* utama

Halaman ini merupakan *form* utama yang mana terdapat pilihan tombol menu yang dapat dipilih oleh pengguna sesuai dengan hak akses masing-masing *user*. Berikut hak akses masing-masing dari *user* :

a. Kepala bengkel

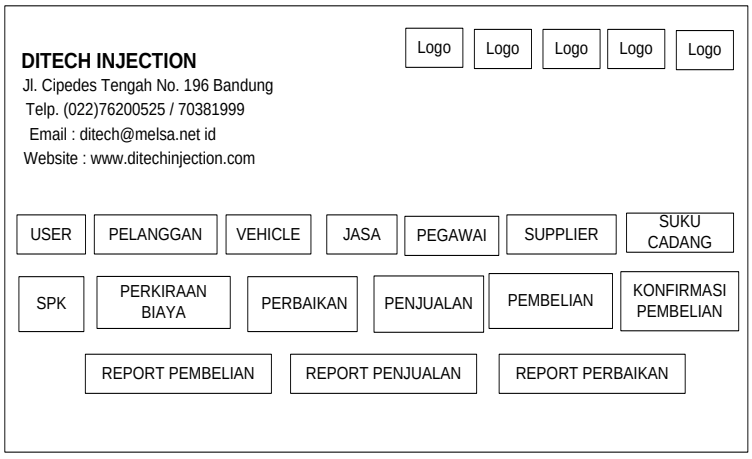
Kepala bengkel dapat mengakses seluruh tombol menu yang terdapat pada *form* utama.

b. *Service advisor*

Service advisor dapat mengakses master pelanggan, kendaraan, jasa, pegawai, SPK, perkiraan biaya, transaksi perbaikan, transaksi penjualan.

c. *Part man*

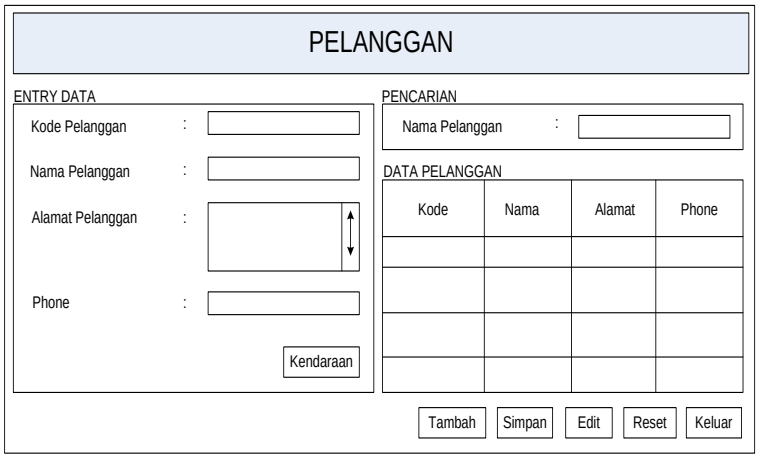
Master suku cadang, master *supplier*, transaksi pembelian, konfirmasi pembelian.



Gambar 4. 26 Desain tampilan *form* utama

3. Rancangan master pelanggan

Halaman ini merupakan *form* master pelanggan. *Form* ini dapat diakses oleh *service advisor* dan kepala bengkel untuk menambahkan data pelanggan yang belum pernah melakukan perbaikan atau terdaftar di *database* dan dalam *form* ini juga tersedia fasilitas pencarian untuk melakukan pemeriksaan apakah pelanggansudah terdaftar atau belum. Setiap pelanggan dapat memiliki satu atau lebih kendaraan.



Gambar 4. 27 Desain tampilan master pelanggan

Keterangan :

a. *Textbox* Kode pelanggan, Nama Pelanggan, Alamat Pelanggan dan Phone digunakan untuk mengisi data dari pelanggan.
b. *Button* Tambah Kendaraan digunakan untuk menambah kendaraan yang dimiliki oleh pelanggan. Dalam hal ini dimungkinkan satu pemilik memiliki lebih dari satu mobil.
c. *Textbox* Cari digunakan untuk mencari pelanggan berdasarkan nama yang diisi, maka secara otomatis setelah diketik akan muncul nama yang diinginkan.

d. *Button* Tambah digunakan untuk menambah data pelanggan.

e. *Button* Simpan digunakan untuk menyimpan data pelanggan setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.

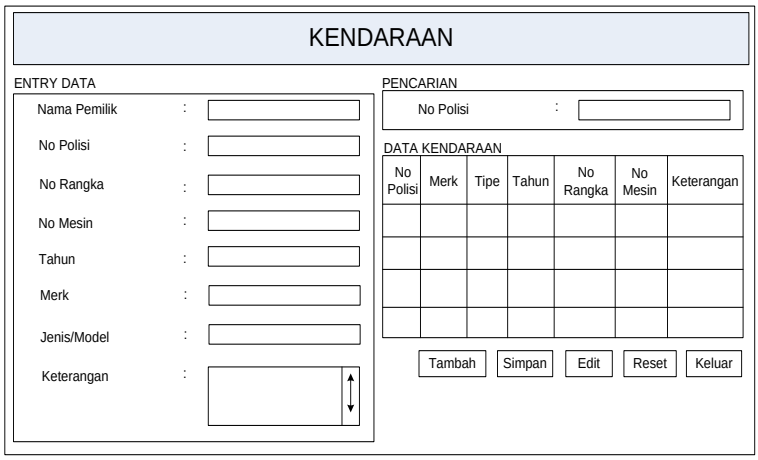
f. *Button* Edit digunakan untuk mengedit data pelanggan yang dipilih dari tabel data pelanggan.

g. *Button* Reset digunakan untuk mereset *input* data yang belum disimpan.

h. *Button* Keluar digunakan untuk kembali ke halaman utama.

4. Rancangan master kendaraan

Halaman ini merupakan *form* master kendaraan. *Form* ini dapat diakses oleh *service advisor* dan kepala bengkel. Data yang diisi pada master ini sesuai dengan Surat Tanda Nomor Kendaaan (STNK) yang dimiliki oleh pelanggan. Tetapi untuk nama pemilik disesuaikan dengan identitas pelanggan yang datang ke bengkel. Fasilitas pencarian juga disediakan dalam *form* ini untuk memudahkan *user* ketika melakukan pencarian dengan mengetikkan nomor polisi.



Gambar 4. 28 Desain tampilan master kendaraan

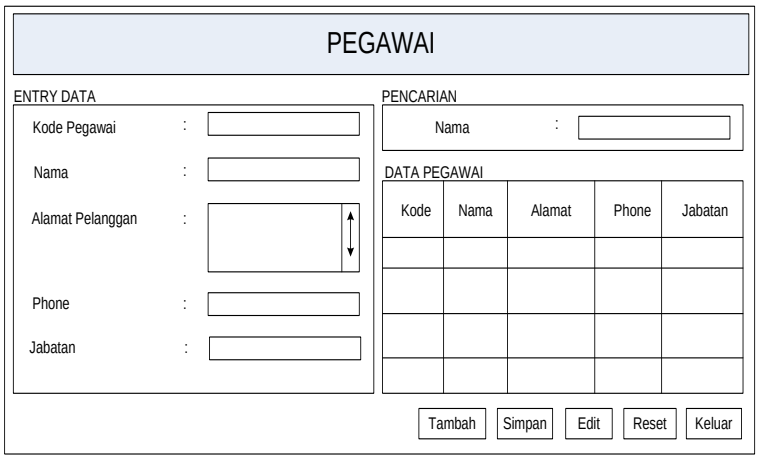
Keterangan :

a. *Textbox* Nama Pemilik akan terisi otomatis sesuai dengan data pelanggan ketika akan menambahkan kepemilikan kendaraan
b. *Textbox* No Polisi, No Rangka, No Mesin, Tahun dan Keterangan digunakan untuk mengisi data kendaraan.
c. *Combobox* Merk digunakan untuk menentukan jenis merk kendaraan
d. *Combobox* Jenis/Model digunakan untuk menentukan jenis/model kendaraan.
e. *Textbox* Cari digunakan untuk mencari kendaraan berdasarkan nomor polisi yang diisi, maka secara otomatis setelah diketik akan muncul nama yang diinginkan.
f. *Button* Tambah digunakan untuk menambah data kendaraan.
g. *Button* Simpan digunakan untuk menyimpan data kendaraan setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.
h. *Button* Edit digunakan untuk mengedit data kendaraan yang dipilih dari tabel data kendaraan.
i. *Button* Reset digunakan untuk mereset *input* data yang belum disimpan.
j. *Button* Keluar digunakan untuk kembali ke halaman utama.

5. Rancangan master pegawai

Halaman ini merupakan *form* master pegawai. Master data ini dapat diakses oleh kepala bengkel. *Form* ini dipergunakan untuk melakukan penambahan atau perubahan data tentang pegawai yang bekerja pada bengkel Ditech

injection. Pencarian juga disediakan berdasarkan nama pegawai untuk memudahkan pencarian data.



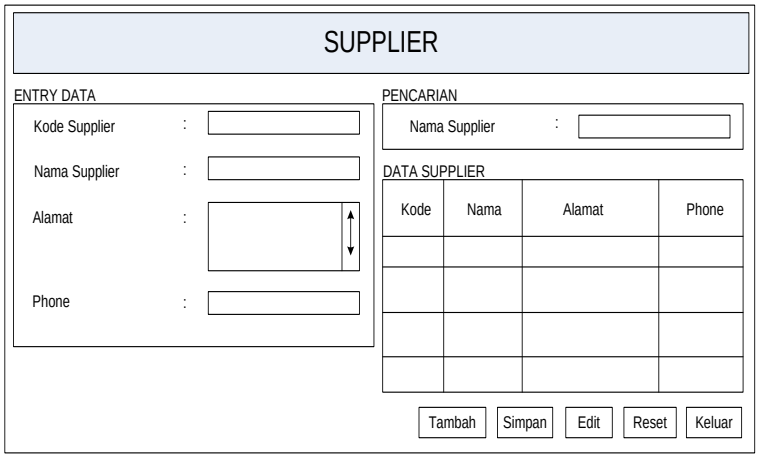
Gambar 4. 29 Desain tampilan master pegawai

Keterangan :

a. *Textbox* Kode Pegawai, Nama Pegawai, Alamat Pegawai dan Phone digunakan untuk mengisi data dari pegawai.
b. *Combobox* Jabatan digunakan untuk menentukan jabatan dari pegawai.
c. *Textbox* Cari digunakan untuk mencari pegawai berdasarkan nama yang diisi, maka secara otomatis setelah diketik akan muncul nama pegawai yang diinginkan.
d. *Button* Tambah digunakan untuk menambah data pegawai.
e. *Button* Simpan digunakan untuk menyimpan data pegawai setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.
f. *Button* Edit digunakan untuk mengedit data pegawai yang dipilih dari tabel data pegawai.
g. *Button* Reset digunakan untuk mereset input data yang belum disimpan.
h. *Button* Keluar digunakan untuk kembali ke halaman utama.

6. Rancangan master *supplier*

Halaman ini merupakan *form* master *supplier*. *Form* ini dapat diakses oleh *part man* dan kepala bengkel. Master data *supplier* digunakan untuk melakukan penambahan atau perubahan data *supplier* yang bekerja sama dengan bengkel Ditech Injection.



Gambar 4. 30 Desain tampilan master *supplier*

Keterangan :

a. *Textbox* Kode *Supplier*, Nama *Supplier*, Alamat *Supplier* dan Phone digunakan untuk mengisi data dari *supplier*.
b. *Textbox* Cari digunakan untuk mencari *supplier* berdasarkan nama yang diisi, maka secara otomatis setelah diketik akan muncul nama supllier yang diinginkan.
c. *Table* digunakan untuk menampung semua data *supplier*.
d. *Button* Tambah digunakan untuk menambah data *supplier*.
e. *Button* Simpan digunakan untuk menyimpan data *supplier* setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.

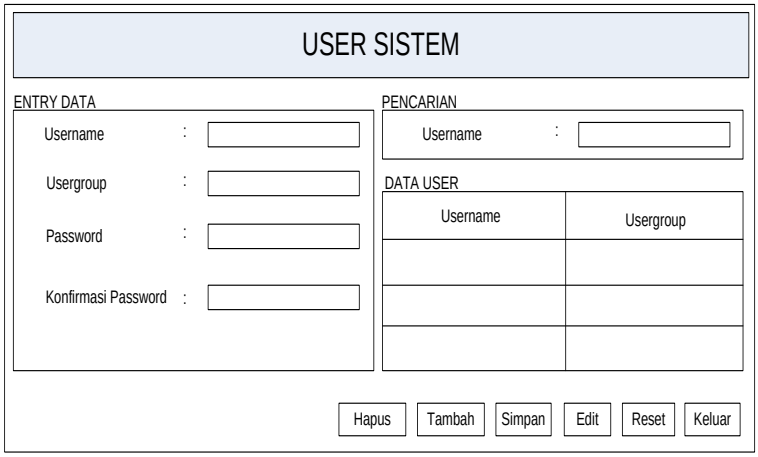
f. *Button* Edit digunakan untuk mengedit data *supplier* yang dipilih dari tabel data *supplier*.

g. *Button* Reset digunakan untuk mereset *input* data yang belum disimpan.

h. *Button* Keluar digunakan untuk kembali ke halaman utama.

7. Rancangan master *user*

Halaman ini merupakan *form* master *user*. Hanya kepala bengkel yang memiliki akses terhadap *form* ini dan digunakan apabila akan menambahkan *user* yang mana karyawan yang dapat mengakses sistem sesuai dengan jabatannya. Setiap *user* akan mendapatkan *username* dan password yang diberikan untuk *login* ke sistem.



Gambar 4. 31 Desain tampilan master *user*

Keterangan :

a. *Textbox Username* digunakan untuk mengisi data dari *user*.

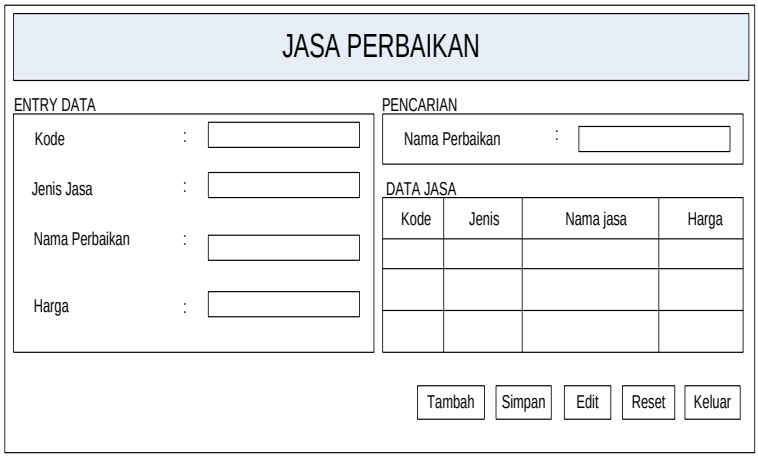
b. *Combobox User* Group digunakan untuk menentukan group dari *user* tersebut

c. *Textbox Password* digunakan untuk input password yang akan digunakan oleh *user*

d. Konfimasi *password* digunakan untuk melakukan konfirmasi *password* yang sesuai dengan diisi di *textbox password*.
e. *Textbox* Cari digunakan untuk mencari *user* berdasarkan huruf yang diketik, maka secara otomatis setelah diketik akan muncul nama *user* yang diinginkan.
f. *Button* Tambah digunakan untuk menambah data *user*.
g. *Button* Simpan digunakan untuk menyimpan data *user* setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.
h. *Button* Edit digunakan untuk mengedit data *user* yang dipilih dari tabel *user*.
i. *Button* Reset digunakan untuk mereset *input* data yang belum disimpan.
j. *Button* Keluar digunakan untuk kembali ke halaman utama.

8. Rancangan master jasa perbaikan

Halaman ini merupakan *form* master jasa perbaikan. *Form* ini dapat diakses oleh *service advisor* dan kepala bengkel. Master data jasa digunakan untuk menambahkan atau merubah jenis perbaikan. Untuk memudahkan pencarian dan mempercepat transaksi disediakan menu pencarian sesuai nama jasa perbaikan



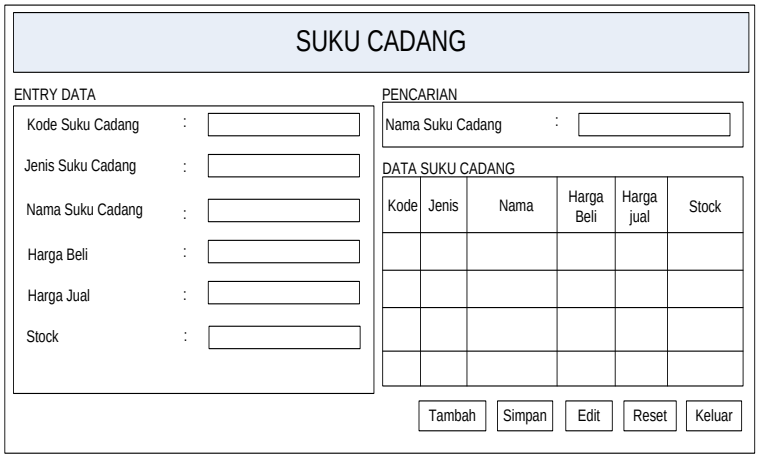
Gambar 4. 32 Desain tampilan master jasa perbaikan

Keterangan :

a. *Textbox* Kode, Nama perbaikan dan Harga digunakan untuk mengisi data dari jasa perbaikan
b. *Combobox* Jenis digunakan untuk menentukan jenis dari jasa.
c. *Textbox* Cari digunakan untuk mencari nama jasa berdasarkan huruf yang diketik, maka secara otomatis setelah diketik akan muncul nama jasa yang diinginkan.
d. *Button* Tambah digunakan untuk menambah data jasa.
e. *Button* Simpan digunakan untuk menyimpan data jasa setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.
f. *Button* Edit digunakan untuk mengedit data jasa yang dipilih dari tabel *jasa*.
g. *Button* Reset digunakan untuk mereset *input* data yang belum disimpan.
h. *Button* Keluar digunakan untuk kembali ke halaman utama.

9. Rancangan master suku cadang

   Halaman ini merupakan *form* master data suku cadang. Master ini dipergunakan untuk menambahkan jenis suku cadang atau melakukan perubahan data mengenai suku cadang tersebut. Untuk memudahkan pencarian terdapat fasilitas pencarian berdasarkan nama suku cadang.



Gambar 4. 33 Desain tampilan master data suku cadang

Keterangan :

a. *Textbox* Kode Suku Cadang, Nama Suku Cadang, Harga Beli, Harga Jual dan Stock digunakan untuk mengisi data dari suku cadang.
b. *Combobox* Jenis digunakan untuk menentukan kategori dari suku cadang tersebut.
c. *Textbox* Cari digunakan untuk mencari nama suku cadang berdasarkan huruf yang diketik, maka secara otomatis setelah diketik akan muncul nama suku cadang yang diinginkan.
d. *Button* Tambah digunakan untuk menambah data suku cadang.
e. *Button* Simpan digunakan untuk menyimpan data suku cadang setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.

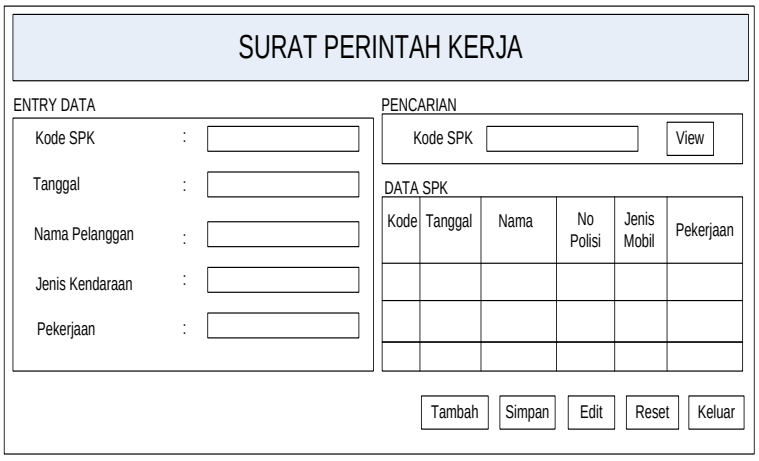
f. *Button* Edit digunakan untuk mengedit data suku cadang yang dipilih dari tabel suku cadang.

g. *Button* Reset digunakan untuk mereset input data yang belum disimpan.

h. *Button* Keluar digunakan untuk kembali ke halaman utama.

10. Rancangan surat perintah kerja (SPK)

Halaman ini merupakan surat perintah kerja. *Form* ini dipergunakan untuk melakukan pembuatan SPK atau surat perintah kerja. Dalam hal ini *service advisor* menanyakan jenis pekerjaan dan keluhan kepada pelanggan. Segala jenis permintaan dan keluhan dimasukkan kedalam kolom pekerjaan/keluhan oleh *service advisor*. Untuk selanjutnya *form* ini dicetak dan kemudian diserahkan kepada mekanik.



Gambar 4. 34 Desain tampilan SPK

Keterangan :

a. *Textbox* Kode SPK dan tanggalotomatis digenerate ketika tombol tambah ditekan.
b. *Combobox* Nama Pelanggan dan No Polisi digunakan untuk memilih pelanggan dan nomor polisi kendaraan yang akan melakukan perbaikan.
c. *TextArea* Keterangan otomatis di generate ketika *combobox* nomor polisi dipilih. Keterangan ini berisi merk dan tipe mobil pelanggan.
d. *TextArea* Pekerjaan diisi dengan jenis pekerjaan dan keluhan berdasarkan permintaan pelanggan.
e. *Button* Tambah digunakan ketika akan melakukan penambahan atau pembuatan SPK baru
f. *Button* Simpan digunakan untuk melakukan penyimpanan ketika proses pengisian selesai, dari tombol ini dihasilkan SPK seperti yang terdapat pada gambar 4.35.
g. *Button* Reset digunakan untuk mereset input data yang belum disimpan.
h. *Button* Keluar digunakan untuk kembali ke halaman utama.

11. Form SPK yang dicetak

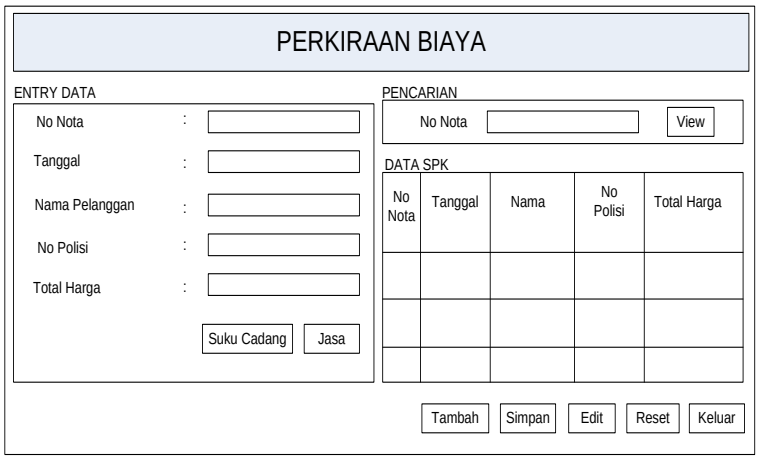
Halaman ini merupakan *form* SPK yang dicetak untuk diserahkan kepada mekanik. *Form* ini mencantumkan seluruh jenis keluhan dan pekerjaan yang masih menggunakan bahasa pelanggan kepada *service advisor*. Untuk itu diperlukan mekanik yang berpengalaman dalam hal ini. Pada *form* ini diperlukan tanda tangan dan nama mekanik yang diisi oleh mekanik yang bersangkutan.



Gambar 4. 35 Desain form SPK

12. Rancangan *form* perkiraan biaya perbaikan

    Halaman ini merupakan *form* perkiraan biaya perbaikan. *Form* ini dapat diakses oleh *service advisor*. Tujuan dari pembuatan *form* ini adalah untuk mengetahui jenis jasa pekerjaan, jenis suku cadang yang diperlukan dan jumlah biaya yang diperlukan. *Form* ini diisi sesuai dengan hasil analisa dan investigasi yang dilakukan oleh mekanik. Tujuan lain dari *form* ini adalah mengurangi terjadinya kesalah pahaman antara pelanggan dan pihak bengkel pada saat pembayaran.



Gambar 4. 36 Desain tampilan *form* perkiraan biaya perbaikan

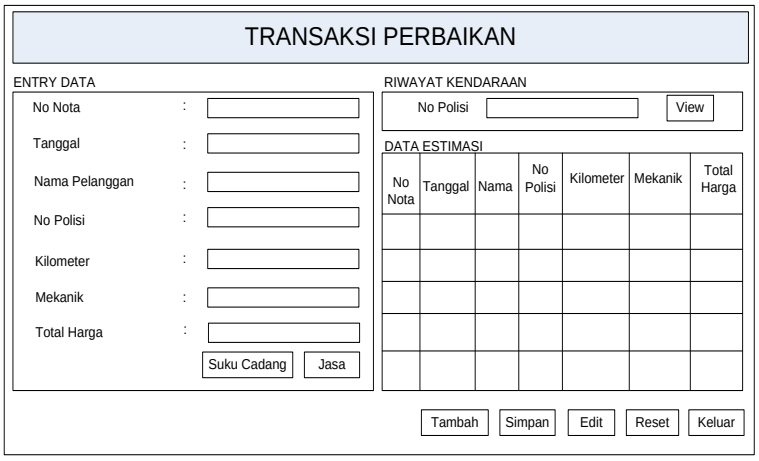
Keterangan :

a. *Textbox* No.Nota dan Tanggal otomatis digenerate ketika tombol tambah ditekan.
b. *Combobox* Nama Pelanggan dan No Polisi digunakan untuk memilih pelanggan dan nomor polisi kendaraan yang akan melakukan perbaikan.
c. *Textbox* Total Harga akan otomatis terisi total harga jumlah perkiraan biaya apabila telah melakukan pemilihan suku cadang dan jasa.
d. *Button* Tambah Suku Cadang digunakan untuk melakukan pemilihan jenis suku cadang apa yang akan digunakan.
e. *Button* Tambah jasa digunakan untuk melakukan pemilihan jenis pekerjaan jasa yang akan dilakukan oleh mekanik.
f. *Button* Tambah digunakan untuk menambah data perkiraan biaya.
g. *Button* Simpan digunakan untuk menyimpan data perkiraan biaya setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem. Dan dari proses penyimpanan ini akan dihasilkan pencetakan *form* perkiraan biaya yang nantinya akan diserahkan kepada pelanggan.
h. *Button* Cancel digunakan untuk membatalkan transaksi.
i. *Button* Reset digunakan untuk mereset input data yang belum disimpan.
j. *Button* Keluar digunakan untuk kembali ke halaman utama.

13. Rancangan *form* transaksi perbaikan

Halaman ini merupakan *form* transaksi perbaikan. Forn ini dapat diakses oleh service advisor dan kepala bengkel. *Form* ini dipergunakan ketika pekerjaan perbaikan yang dilakukan oleh mekanik telah selesai dan telah dilakukan

pengecekan akhir (*final inspection*) oleh kepala bengkel. Dalam *form* ini dapat dilakukan input tambah suku cadang yang akan masuk ke dalam *form* penjualan suku cadang dan input tambah jasa yang akan masuk ke dalam *form* penjualan jasa. Dalam *form* ini juga disertakan fasilitas pencarian riwayat kendaraan *(history car)* untuk memudahkan pelanggan mengetahui daftar riwayat jasa kendaraannya di bengkel Ditech Injection.



Gambar 4. 37 Desain tampilan *form* transaksi perbaikan

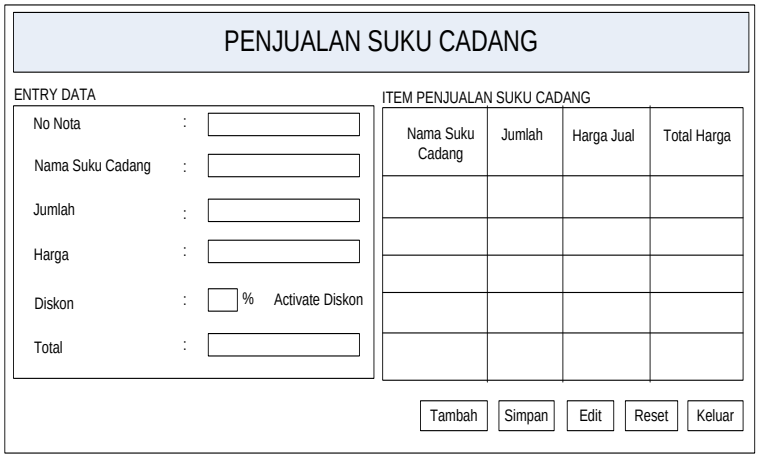
Keterangan :

a. *Textbox* No.Nota dan Tanggal otomatis digenerate ketika tombol tambah ditekan.
b. *Combobox* Nama Pelanggan dan No Polisi digunakan untuk memilih pelanggan dan nomor polisi kendaraan yang akan melakukan perbaikan.
c. *Textbox* Total Harga akan otomatis terisi total harga jumlah perkiraan biaya perbaikan apabila telah melakukan pemilihan suku cadang dan jasa.
d. *Textbox* Kilometer diisi sesuai dengan angka kilometer di spedometer mobil pelanggan pada saat selesai dikerjakan

e. *Combobox* Mekanik digunakan untuk memilih siapa mekanik yang mengerjakan mobil tersebut.
f. *Combobox* History Car digunakan untuk memilih nomor polisi kendaraan pelanggan yang akan melihat riwayat perbaikan kendaraan pelanggan di bengkel.
g. *Button* Tambah Suku Cadang digunakan untuk melakukan pemilihan jenis suku cadang apa yang akan digunakan.
h. *Button* Tambah jasa digunakan untuk melakukan pemilihan jenis pekerjaan jasa yang akan dilakukan oleh mekanik.
i. *Button* Tambah digunakan untuk menambah data transaksi perbaikan.
j. *Button* Simpan digunakan untuk menyimpan data perbaikan setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem. Dan dari proses penyimpanan ini akan dihasilkan pencetakan nota perbaikan yang nantinya akan diserahkan kepada pelanggan.
k. *Button* Cancel digunakan untuk membatalkan transaksi.
l. *Button* Reset digunakan untuk mereset input data yang belum disimpan.
m. *Button* Keluar digunakan untuk kembali ke halaman utama.

14. Rancangan *form* penjualan suku cadang

Halaman ini merupakan *form* penjualan suku cadang. *Form* ini merupakan tampilan dari *button* Tambah Suku Cadang pada *form* perkiraan biaya dan *form* transaksi perbaikan dan dapat dipergunakan untuk melakukan penambahan jenis suku cadang apa saja yang dipergunakan ketika mekanik melakukan perbaikan sesuai dengan perkiraan biaya perbaikan.



Gambar 4. 38 Desain tampilan *form* penjualan suku cadang

Keterangan :

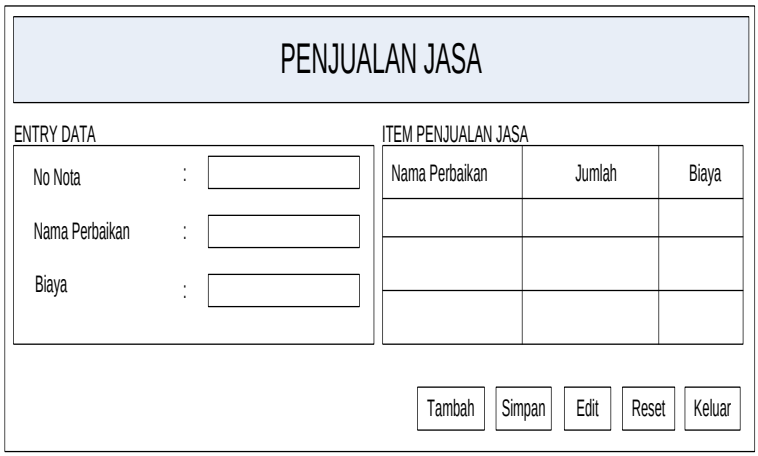
a. *Textbox* No.Nota akan otomatis digenerate ketika *button* tambah transaksi dipilih.
b. *Combobox* Nama Suku Cadang digunakan untuk memilih jenis suku cadang.
c. *Textbox* Harga akan otomatis terisi ketika dilakukan pemilihan pada *combobox* Nama Suku Cadang
d. *Textbox* Total akan terisi secara otomatis hasil penjumlahan harga suku cadang apabila seluruh pemilihan jenis suku cadang telah dipilih dan disimpan dalam tabel.
e. *Textbox* Jumlah digunakan untuk melakukan pengisian jumlah suku cadang yang digunakan pada saat perbaikan kendaraan.
f. *Button* Tambah digunakan untuk menambah data penjualan suku cadang
g. *Button* Simpan digunakan untuk menyimpan data penjualan suku cadang setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.

h. *Button* Reset digunakan untuk mereset input data yang belum disimpan.

i. *Button* Keluar digunakan untuk kembali ke halaman utama.

15. Rancangan *form* penjualan jasa

Halaman ini merupakan *form* penjualan jasa. *Form* ini merupakan tampilan dari *button* Tambah jasa pada *form* perkiraan biaya perbaikan dan *form* transaksi perbaikan. Dalam hal ini dapat ditambahkan jenis perbaikan/jasa apa saja yang telah dilakukan oleh mekanik atau akan dilakukan oleh mekanik



Gambar 4. 39 Desain tampilan *form* penjualan jasa

Keterangan :

a. *Textbox* No.Nota akan otomatis digenerate ketika *button* tambah pada transaksi perbaikan dipilih.

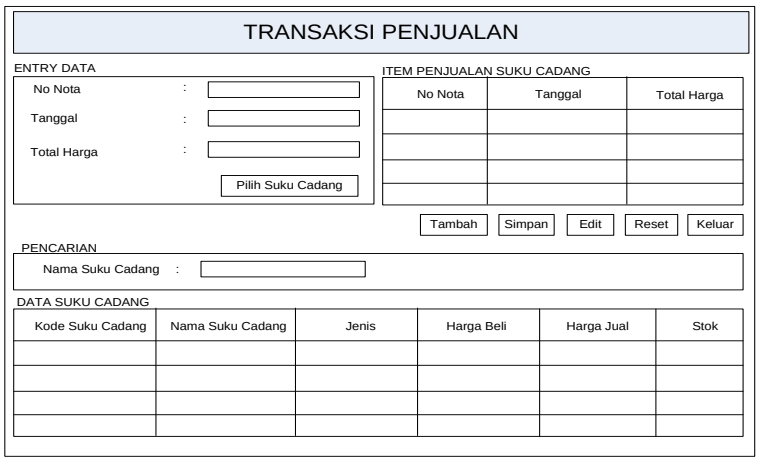
b. *Combobox* Nama jasa digunakan untuk memilih jenis pekerjaan jasa.

c. *Textbox* Harga akan otomatis terisi ketika dilakukan pemilihan pada *combobox* Nama Jasa / jenis pekerjaan.

d. *Textbox* Total akan terisi secara otomatis hasil penjumlahan harga atau biaya jasa apabila seluruh pemilihan jenis pekerjaan telah dipilih dan disimpan dalam tabel.
e. *Button* Tambah digunakan untuk menambah data penjualan jasa
f. *Button* Simpan digunakan untuk menyimpan data penjualan jasa setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.
g. *Button* Reset digunakan untuk mereset input data yang belum disimpan.
h. *Button* Keluar digunakan untuk kembali ke halaman utama.

16. Rancangan *form* transaksi penjualan

Halaman ini merupakan *form* transaksi penjualan. *Form* ini dapat diakses oleh *part man* dan kepala bengkel. *Form* ini dipergunakan ketika terdapat pelanggan yang hanya melakukan pembelian suku cadang di bengkel tanpa melakukan perbaikan.



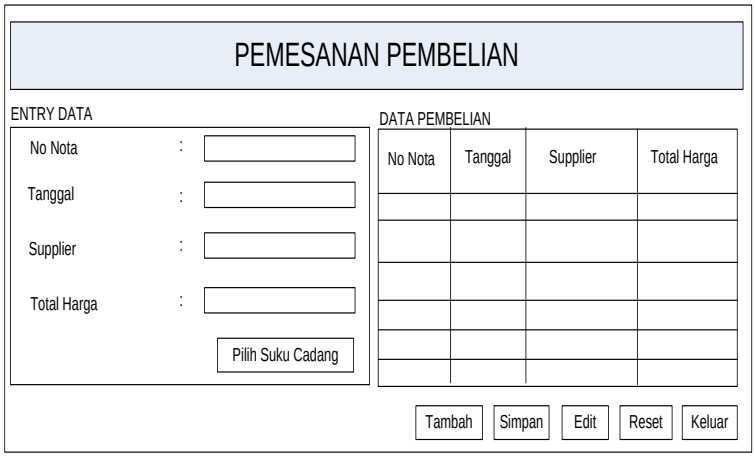
Gambar 4. 40 Desain tampilan *form* transaksi penjualan

Keterangan :

a. *Textbox* No.Nota dan Tanggal otomatis digenerate ketika tombol tambah ditekan.
b. *Textbox* Total Harga akan otomatis terisi total harga jumlah penjualan apabila telah melakukan pemilihan suku cadang.
c. *Button* Tambah Suku Cadang digunakan untuk melakukan pemilihan jenis suku cadang apa yang akan digunakan.
d. *Button* Cancel digunakan untuk membatalkan transaksi.
e. *Button* Reset digunakan untuk mereset input data yang belum disimpan.
f. *Button* Keluar digunakan untuk kembali ke halaman utama.

17. Rancangan *form* pemesanan pembelian

Halaman ini merupakan *form* transaksi pemesanan suku cadang kepada *supplier*. *Form* ini dipergunakan untuk melakukan pemesanan pembelian kepada *supplier*.



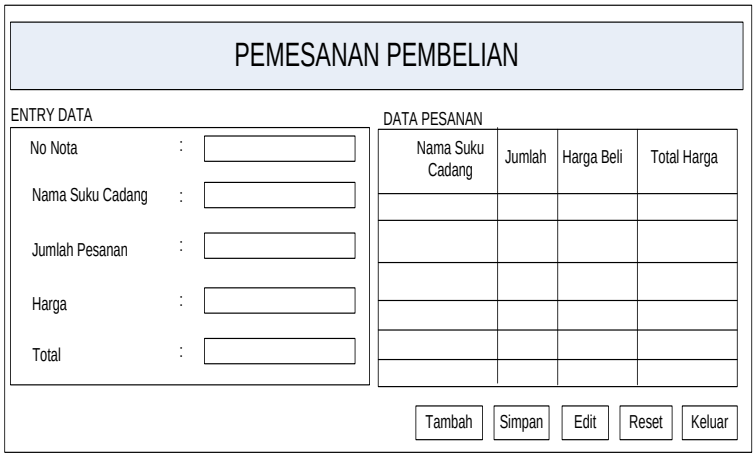
Gambar 4. 41 Desain tampilan *form* transaksi pemesanan

Keterangan :

a. *Textbox* No. Nota dan Tanggal akan otomatis digenerate ketika *button* Tambah transaksi dipilih.
b. *Combobox Supplier* digunakan untuk memilih *supplier*
c. *Textbox* Total Harga akan otomatis terisi ketika dilakukan pemilihan suku cadang yang akan dipesan.
d. *Button* Tambah Suku Cadang digunakan untuk melakukan pemilihan jenis suku cadang apa yang akan dipesan
e. *Button* Tambah digunakan untuk menambah data transaksi pemesanan suku cadang
f. *Button* Simpan digunakan untuk menyimpan data penjualan suku cadang setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.
g. *Button* Cancel digunakan untuk membatalkan transaksi.
h. *Button* Reset digunakan untuk mereset input data yang belum disimpan.
i. *Button* Keluar digunakan untuk kembali ke halaman utama.

18. Rancangan *form* pemesanan pembelian

    Halaman ini merupakan *form* pemesanan pembelian. Pada *form* berikut *part man* melakukan pemilihan jenis suku cadang apa saja yang akan dipesan dan berapa jumlah kebutuhannya.



Gambar 4. 42 Desain tampilan *form* pemesanan pembelian

Keterangan :

a. *Textbox* No.Nota akan otomatis digenerate ketika *button* tambah transaksi dipilih.
b. *Combobox* Nama Suku Cadang digunakan untuk memilih jenis suku cadang.
c. *Textbox* Harga akan otomatis terisi ketika dilakukan pemilihan pada *combobox* Nama Suku Cadang
d. *Textbox* Total akan terisi secara otomatis hasil penjumlahan harga suku cadang apabila seluruh pemilihan jenis suku cadang telah dipilih dan disimpan dalam tabel.
e. *Textbox* Jumlah digunakan untuk melakukan pengisian jumlah suku cadang yang digunakan pada saat perbaikan kendaraan.
f. *Button* Tambah digunakan untuk menambah data pemesanan pembelian suku cadang

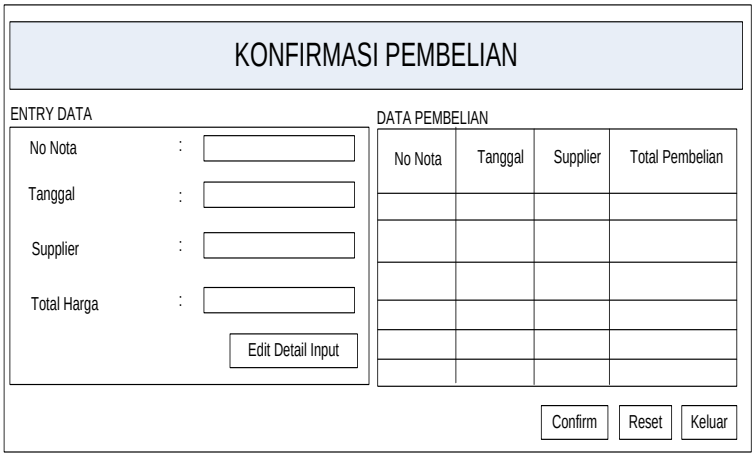
g. *Button* Simpan digunakan untuk menyimpan data data pemesanan pembelian suku cadang setelah dilakukan *input* data dengan benar dan telah divalidasi oleh sistem.

h. *Button* Reset digunakan untuk mereset input data yang belum disimpan.

i. *Button* Keluar digunakan untuk kembali ke halaman utama.

19. Rancangan *form* konfirmasi pemesanan

Halaman ini merupakan *form* konfirmasi pemesanan. *Form* ini dipergunakan ketika barang pesanan datang dari *supplier*. Apabila barang yang datang sesuai dengan permintaan pesanan maka dilakukan konfirmasi.



Gambar 4. 43 Desain tampilan *form* konfirmasi pemesanan

Keterangan :

a. *Textbox* No.Nota dan Tanggal akan otomatis digenerate ketika daftar pemesanan pembelian yang ada di dalam tabel dipilih (double klik). Daftar pemesanan di tabel menunjukkan bahwa pesanan barang/suku cadang belum datang atau diterima oleh *part man*

b. *Combobox Supplier* akan otomatis digenerate ketika daftar pemesanan pembelian yang ada di dalam tabel dipilih (double klik).

c. *Textbox* Total Harga akan otomatis digenerate ketika daftar pemesanan pembelian yang ada di dalam tabel dipilih (double klik).

d. *Button* Tambah Suku Cadang digunakan untuk penyesuaian atau terjadi perbedaan antara barang yang dipesan dan yang datang. Perbedaan yang dimaksud yaitu jumalah barang sampai dan harga pembelian.

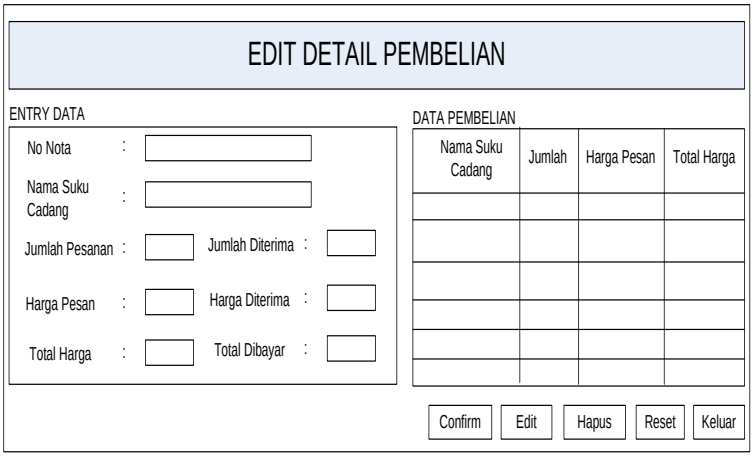
e. *Button* Konfirmasi digunakan apabila pesanan suku cadang yang datang dan diterima oleh *part man* sesuai jumlah dan harganya pada saat pemesanan pembelian dilakukan.

f. *Button* Reset digunakan untuk mereset input data yang belum disimpan.

g. *Button* Keluar digunakan untuk kembali ke halaman utama.

20. Rancangan *form* konfirmasi pembelian suku cadang

Halaman ini merupakan *form* konfirmasi pembelian. *Form* berikut merupakan bagian dari *form* konfirmasi pemesanan. *Form* ini dipergunakan ketika barang yang datang jumlah atau harganya berbeda. Dan kemudian dilakukan penyesuaian mengenai jumlah dan harga.



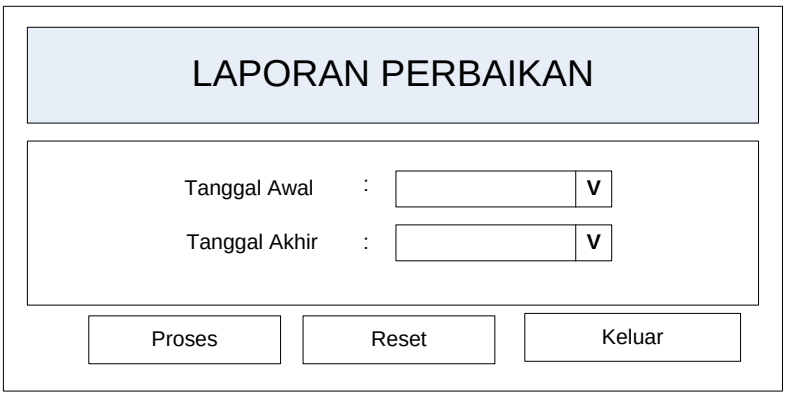
Gambar 4. 44 Desain tampilan *form* konfirmasi pembelian suku cadang

Keterangan :

a. *Textbox* No.Nota, Tanggal, Jumlah, Harga Pesan dan Total Harga Pesan akan otomatis digenerate ketika daftar pemesanan pembelian yang ada di dalam tabel dipilih (double klik).
b. *Combobox Supplier* akan otomatis digenerate ketika daftar pemesanan pembelian yang ada di dalam tabel dipilih (double klik).
c. *Textbox* Jumlah Diterima digunakan untuk mengisi jumlah barang yang diterima *part man* berbeda dengan *form* pemesanan pembelian.
d. *Textbox* Harga Diterima digunakan untuk mengisi harga barang yang diterima *part man* berbeda dengan *form* pemesanan pembelian.
e. *Textbox* Total Bayar terisi otomatis berdasarkan hasil perkalian *Textbox* Jumlah dengan *Textbox* Harga Sampai.
f. *Button* Konfirmasi digunakan untuk melakukan konfimasi bahwa semua data yang diisi telah sesuai dan untuk kemudian dilakukan penyimpanan di database.
g. *Button* Reset digunakan untuk mereset input data yang belum disimpan.

21. Rancangan *form* laporan perbaikan

    Halaman ini merupakan *form* laporan perbaikan. *Form* ini dipergunakan oleh kepala bengkel untuk melakukan pemeriksaan laporan perbaikan berdasarkan periode waktu tertentu.

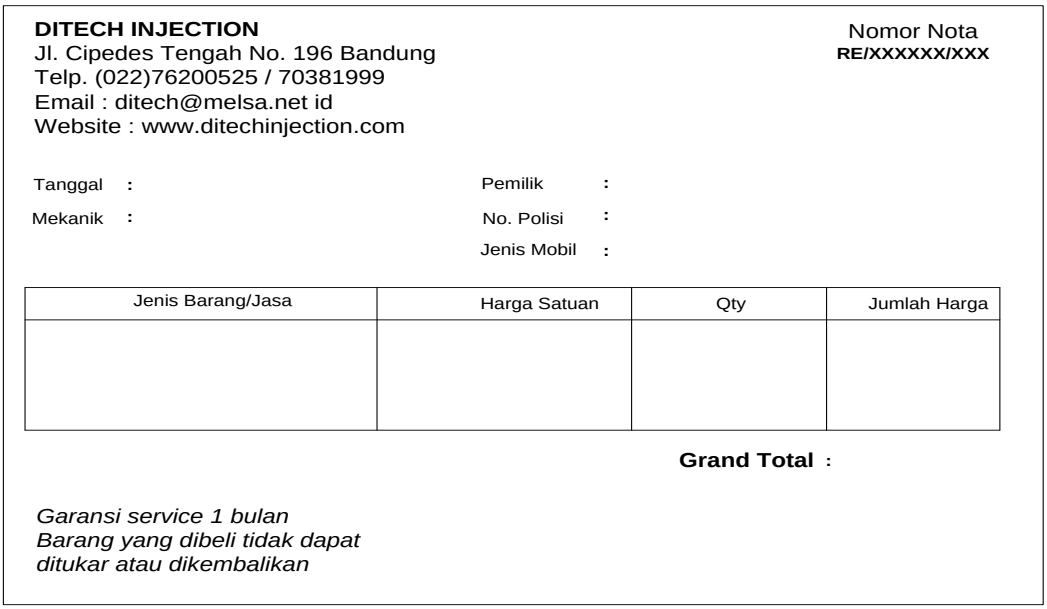


Gambar 4. 45 Desain tampilan *form* laporan perbaikan

4.2.4.2 Perancangan Output

1. Rancangan nota perbaikan

   Halaman ini merupakan rancangan nota perbaikan



Gambar 4. 46 Desain tampilan nota perbaikan

2. Rancangan nota penjualan

Halaman ini merupakan rancangan nota penjualan



DITECH INJECTION
Jl. Cipedes Tengah No . 196 Bandung
Telp. (022)76200525 / 70381999
Email : ditech@melsa.net id
Website : www.ditechinjection.com

Nomor Nota
SE/XXXXXX/XXX

Tanggal : <date>

Nama Suku Cadang	Harga Satuan	Qty	Jumlah Harga

Barang yang dibeli tidak dapat ditukar atau dikembalikan

Grand Total :

Gambar 4. 47 Desain tampilan nota penjualan

3. Rancangan *form* pemesanan pembelian

Halaman ini merupakan rancangan *form* pemesanan pembelian



DITECH INJECTION
Jl. Cipedes Tengah No. 196 Bandung
Telp. (022)76200525 / 70381999
Email : ditech@melsa.net id
Website : www.ditechinjection.com

Pemesanan Pembelian
PO/XXXXXX/XXX

Kepada Yth :
:

Dengan surat ini kami melakukan pemesanan barang sesuai dengan daftar berikut, mohon utnuk dikirm segera

Jenis Barang/Jasa	Harga Satuan	Qty	Jumlah Harga

Hormat kami,

Ditech Injection

Gambar 4. 48 Desain tampilan *form* pemesanan pembelian

4. Rancangan laporan penjualan

   Halaman ini merupakan rancangan laporan penjualan


DITECH INJECTION

LAPORAN PENJUALAN

No Nota	Tanggal	Jenis Barang	Qty	Harga Barang	Jumlah
		GRAND TOTAL :	7		880000

Gambar 4. 49 Desain tampilan laporan penjualan

5. Rancangan laporan perbaikan

   Halaman ini merupakan rancangan laporan perbaikan


DITECH INJECTION

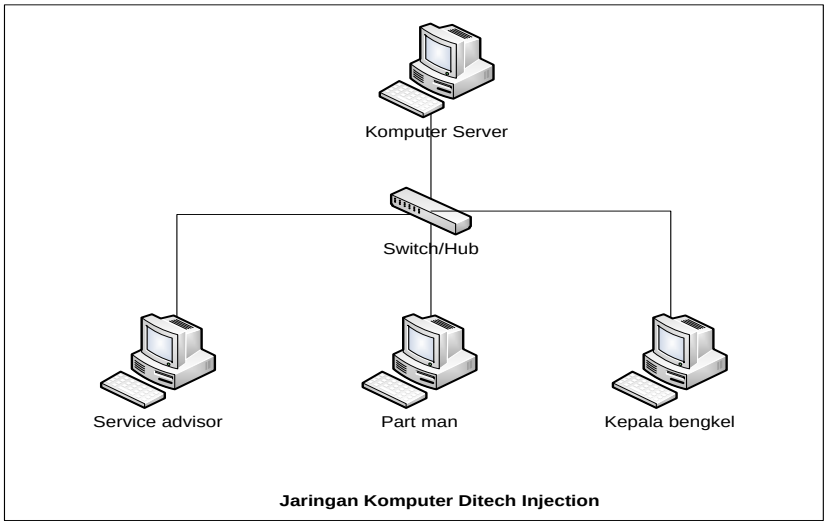
LAPORAN PERBAIKAN

No Nota	Tanggal	Pemilik	No Polisi	Mekanik	Jenis Jasa	Biaya
					GRAND TOTAL :	925000

Gambar 4. 50 Desain tampilan laporan perbaikan

4.2.5 Perancangan Arsitektur Jaringan

Arsitektur jaringan yang digunakan adalah sistem klien server mempunyai dua komponen utama yaitu komputer server dan komputer klien. Server merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer client dan bertindak sebagai server *database* yang menyimpan data. Klien yaitu komputer yang melakukan pengiriman permintaan-permintaan data pada server kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya.



Gambar 4. 51 Arsitektur jaringan yang diusulkan

Gambar 4.51 menunjukkan arsitektur jaringan. Disini penulis menggunakan topologi star dengan alasan jika salah satu komputer klien rusak, komputer yang lain masih dapat diakses dan kontrol menagement lebih mudah karena semuanya terpusat ke server.