BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem yang Sedang Berjalan

Adapun sistem yang sedang berjalan pada Dealer Rajamandala Motor sampai saat ini yaitu sistem informasi pengolahan data yang berjalan pada perusahaan tersebut masih ditulis secara manual. Pengarsipan data *customer*, pengarsipan daftar *spare part*, hingga pada proses transaksi masih dilakukan pencatatan manual, baik itu transaksi pembelian, penjualan *spare part* dan pelayanan servis serta persediaan *spare part* masih dicatat manual dengan proses penghitungan menggunakan kalkulator dan diakhir periodepun masih dengan menggunakan Microsoft Office Excel.

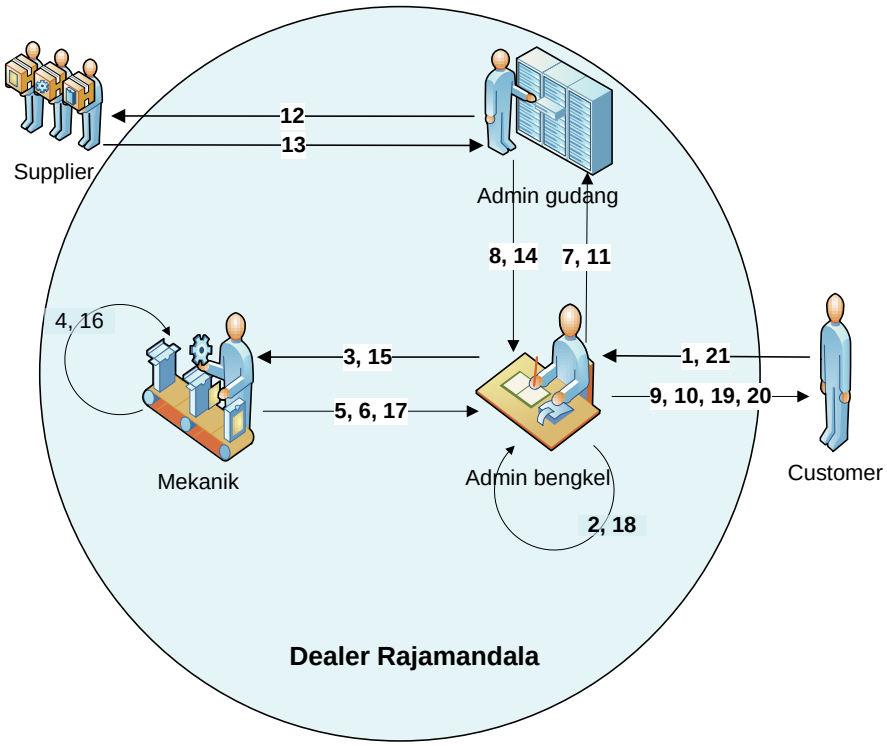
4.1.1 Analisis Prosedur yang Sedang Berjalan

Analisis prosedur yaitu proses analisa terhadap kegiatan-kegiatan yang dilakukan oleh Dealer Rajamandala Motor dalam proses pelayanan pembelian, penjualan *spare part* dan servis motor. Analisis terhadap prosedur yang sedang berjalan seharuanya dilakukan terlebih dahulu sebelum melakukan perancangan terhadap sistem informasi yang akan diusulkan. Harus diketahui pula hal-hal yang menjadi tujuan pemakai sehingga masalah tersebut dapat didefinisikan secara jelas.

4.1.1.1 Proses Bisnis Berjalan

Proses bisnis merupakan suatu kumpulan aktivitas atau pekerjaan yang saling terkait untuk menyelesaikan suatu maslah tertentu yang menghasilkan

produk atau layanan. Proses bisnis servis motor yang sedang berjal an adalah sebagai berikut :



Gambar 4. 1 Proses bisnis servis motor yang sedang berjalan

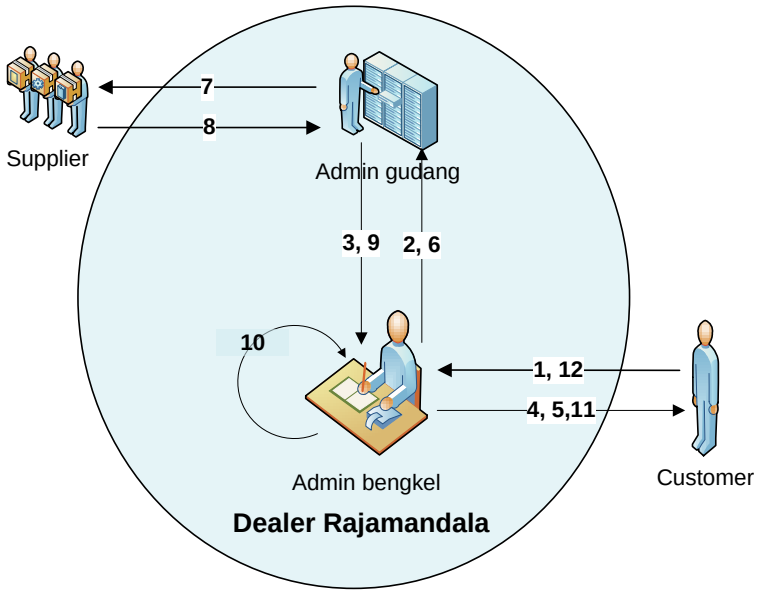
Keterangan :

1. *Customer* datang ke bengkel mendaftar servis kepada admin bengkel dengan menyerahkan KTP atau SIM dan STNK untuk input data.
2. Admin bengkel membuat *work order*.
3. Admin bengkel menyerahkan *work order* ke mekanik.
4. Mekanik melakukan pemeriksaan motor sesuai *work order*
5. Mekanik konfirmasi jenis kerusakan motor dan kerusakan *spare part* ke admin bengkel.

6. Mekanik mengkonfirmasi kebutuhan penggantian *spare part* ke admin bengkel.
7. Admin bengkel menanyakan ketersediaan *spare part* ke admin gudang.
8. Admin gudang mengkonfirmasi ketersediaan kepada admin bengkel
9. Admin bengkel mengkonfirmasi jenis kerusakan dan kebutuhan *spare part* serta ketersediaan ke *customer*, jika *spare part* tersedia dan *customer* setuju dilakukan servis dan penggantian *spare part* lanjut tahap nomor 16, jika *spare part* tidak tersedia lanjut tahap nomor 10
10. Admin bengkel menanyakan kebersediaan menunggu kepada *customer*, jika bersedia lanjut tahap nomor 11, jika tidak *customer* membatalkan servis.
11. Admin bengkel memerintahkan admin gudang untuk melakukan pemesanan dari *supplier*.
12. Admin gudang melakukan pemesanan *spare part* dari *supplier*.
13. *Supplier* menyerahkan *spare part* dan nota pembelian kepada admin gudang (pengiriman *spare part* dilakukan 3 kali dalam seminggu).
14. Admin gudang menyerahkan nota pembelian dan *spare part* kepada admin bengkel.

15. Admin bengkel meminta mekanik melakukan servis dan penggantian *spare part.*
16. Mekanik melakukan servis dan penggantian *spare part.*
17. Mekanik mengkonfirmasi pekerjaan selesai ke admin bengkel.
18. Admin bengkel membuat nota penjualan dan servis.
19. Admin bengkel mengkonfirmasi ke *customer* pekerjaan telah selesai.
20. Nota penjualan dan servis diserahkan ke *customer*.
21. *Customer* melakukan pembayaran.

Proses bisnis penjualan *spare part* yang sedang berjalan adalah sebagai berikut :

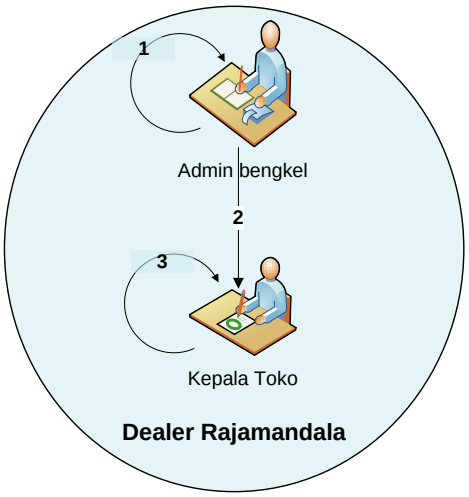


Gambar 4. 2 Proses bisnis penjualan *spare part* yang sedang berjalan

Keterangan :

1. *Customer* datang ke bengkel untuk membeli *spare part*.
2. Admin bengkel menanyakan ketersediaan ke bagian admin gudang
3. Admin gudang mengkonfirmasi ketersediaan kepada admin bengkel
4. Admin bengkel mengkonfirmasi *spare part* ke *customer*, jika tersedia lanjut ke tahap nomor 10 jika tidak ke tahap nomor 5
5. Admin bengkel menanyakan kebersediaan menunggu kepada *customer* untuk dibelikan dari *supplier* atau tidak, jika bersedia lanjut tahap nomor 6, jika tidak maka *customer* membatalkan pembelian.
6. Admin bengkel memerintahkan admin gudang untuk melakukan pemesanan *spare part* dari *supplier*.
7. Admin gudang melakukan pemesanan *spare part* dari *supplier*.
8. *Supplier* menyerahkan *spare part* dan nota pembelian kepada admin gudang.
9. Admin gudang menyerahkan nota pembelian dan *spare part* kepada admin bengkel.
10. Admin bengkel membuat nota penjualan.
11. Admin bengkel menyerahkan *spare part* dan nota penjualan ke *customer*
12. *Customer* melakukan pembayaran.

Proses bisnis *pelaporan* yang sedang berjalan adalah sebagai berikut :



Gambar 4. 3 Proses bisnis pelaporan yang sedang berjalan

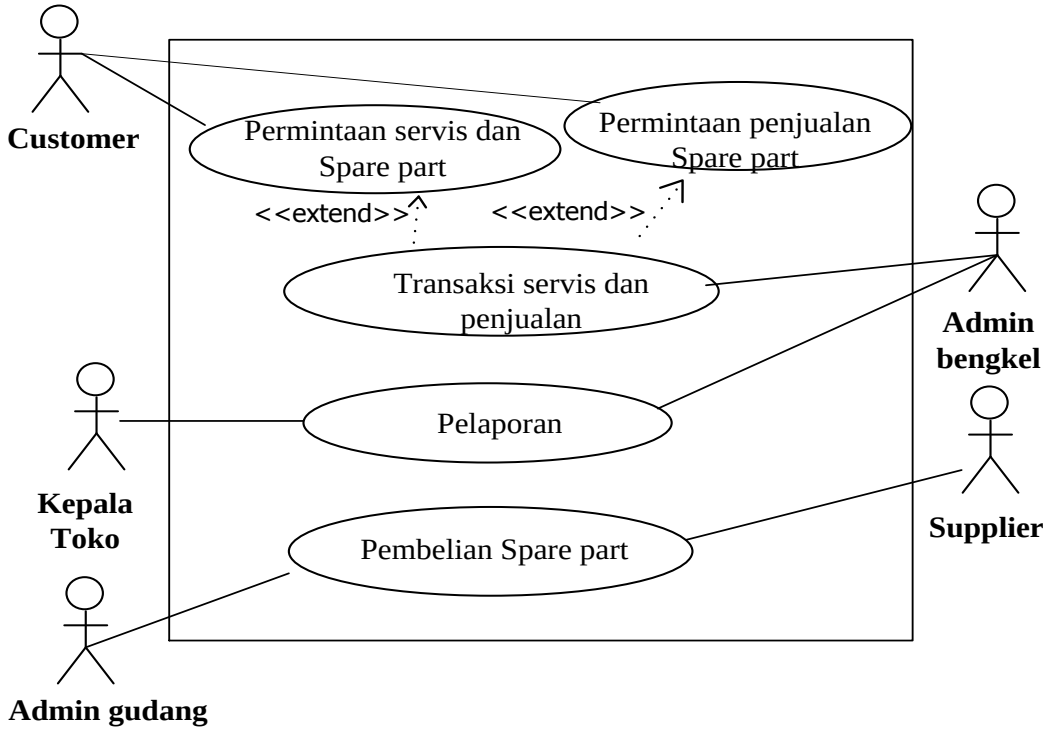
Keterangan :

1. Admin bengkel membuat laporan pembelian, penjualan dan laporan servis.
2. Admin bengkel menyerahkan laporan kepada kepala toko.
3. Kepala toko melakukan pemeriksaan laporan.

4.1.1.2 *Use Case Diagram*

Diagram *use case* dari proses bisnis memperlihatkan hubungan-hubungan antara aktor-aktor bisnis, *use case* proses 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. 4 *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 u*se case* sistem berjalan

No	***Use Case***	**Deskripsi**
UC-01	Permintaan servis dan *spare part*	Fungsionalitas untuk melakukan permintaan servis dan *spare part*
UC-02	Permintaan penjualan *spare part*	Fungsionalitas untuk melakukan permintaan penjualan *spare part*
UC-03	Transaksi servis dan	Fungsionalitas melakukan pencatatan dan

	penjualan	penghitungan transaksi servis dan penjualan
UC-04	Pelaporan	Fungsionalitas untuk melakukan pelaporan
UC-05	Pembelian *spare part*	Fungsionalitas untuk pembelian *spare part* ke *supplier*

2. Identifikasi aktor

Aktor yang berperan dalam menjalankan sistem adalah sebagai berikut :

Tabel 4.2 Identifikasi aktor sistem berjalan

No	**Aktor**	**Deskripsi**
P-01	*Customer*	Aktor di luar sistem yang membutuhkan layanan servis dan *spare part*
P-02	Admin bengkel	Aktor yang berperan mengatur proses administrasi dan transaksi di bengkel
P-03	Admin gudang	Aktor yang berfungsi mengatur persediaan *spare part*.
P-04	Kepala toko	Aktor yang berfungsi memeriksa semua laporan dan bertanggung jawab penuh dengan seluruh kegiatan proses bisnis di Dealer
P-05	Supplier	Aktor yang berada di luar sistem bengkel yang menjual *spare part*.

4.1.1.3 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 :

4.1.1.3.1 Skenario use case permintaan servis dan *spare part*

Tabel 4.3 Skenario *use case* permintaan servis dan *spare part*

Identifikasi	
Judul	Permintaan servis dan *spare part*
No. Use Case	UC-01
Deskripsi Use Case	*Customer* meminta servis dan ingin membeli *spare part*
Aktor	*Customer*
Trigger	*Customer* datang untuk melakukan servis atau membeli *spare part*
Kondisi awal	Admin bengkel siap menerima pendaftaran
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. *Customer* datang ke Dealer melakukan pendaftaran dengan menyerahkan KTP atau SIM dan STNK untuk servis atau membeli spare part	

	2. Admin bengkel mencatat pendaftaran *customer* dan menerima permintaan servis atau pembelian spare part

4.1.1.3.2 Skenario use case permintaan penjualan *spare part*

Tabel 4.4 Skenario *use case* permintaan penjualan *spare part*

Identifikasi	
Judul	Permintaan penjualan *spare part*
No. Use Case	UC-02
Deskripsi Use Case	*Customer* meminta dan ingin membeli *spare part* yang disediakan oleh Dealer
Aktor	*Customer*
Trigger	*Customer* datang untuk melakukan pembelian *spare part*
Kondisi awal	Admin bengkel siap menerima pendaftaran
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. *Customer* datang ke Dealer melakukan pendaftaran pembelian spare part	
	2. Admin bengkel mencatat dan menerima permintaan pembelian spare part

4.1.1.3.3 Skenario *use case* transaksi servis dan penjualan

Tabel 4.5 Skenario *use case* transaksi servis dan penjualan

Identifikasi	
Judul	Transaksi servis dan penjualan
No. Use Case	UC-03
Deskripsi Use Case	Transaksi pembayaran servis motor dan pembelian *spare part* atau penjualan *spare part*
Aktor	Admin bengkel
Trigger	Proses servis dan pembelian *spare part* selesai atau proses penjualan *spare part* selesai
Kondisi awal	Menunggu proses servis dan pembelian *spare part* atau proses penjualan *spare part* selesai
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. Admin bengkel membuat nota pembayaran servis dan penjualan *spare part*	
	2. Menyimpan nota servis dan penjualan *spare part*

3. *Customer* melakukan pembayaran biaya servis dan pembelian *spare part* dan menerima nota pembelian dan servis	

4.1.1.3.4 Skenario use case pelaporan

Tabel 4.6 Skenario *use case* pelaporan

Identifikasi	
Judul	Pelaporan
No. Use Case	UC-04
Deskripsi Use Case	Membuat laporan penjualan, laporan pembelian dan laporan servis
Aktor	Admin bengkel
Trigger	Kepala toko membutuhkan laporan dari admin bengkel
Kondisi awal	Admin bengkel siap untuk membuat laporan
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. Admin bengkel membuat laporan penjualan, laporan pembelian dan laporan servis	

2. Admin bengkel menyerahkan laporan kepada kepala toko	
	3. Kepala toko memeriksa laporan
	4. Kepala toko menyimpan arsip laporan

4.1.1.3.5 Skenario *use case* pembelian *spare part*

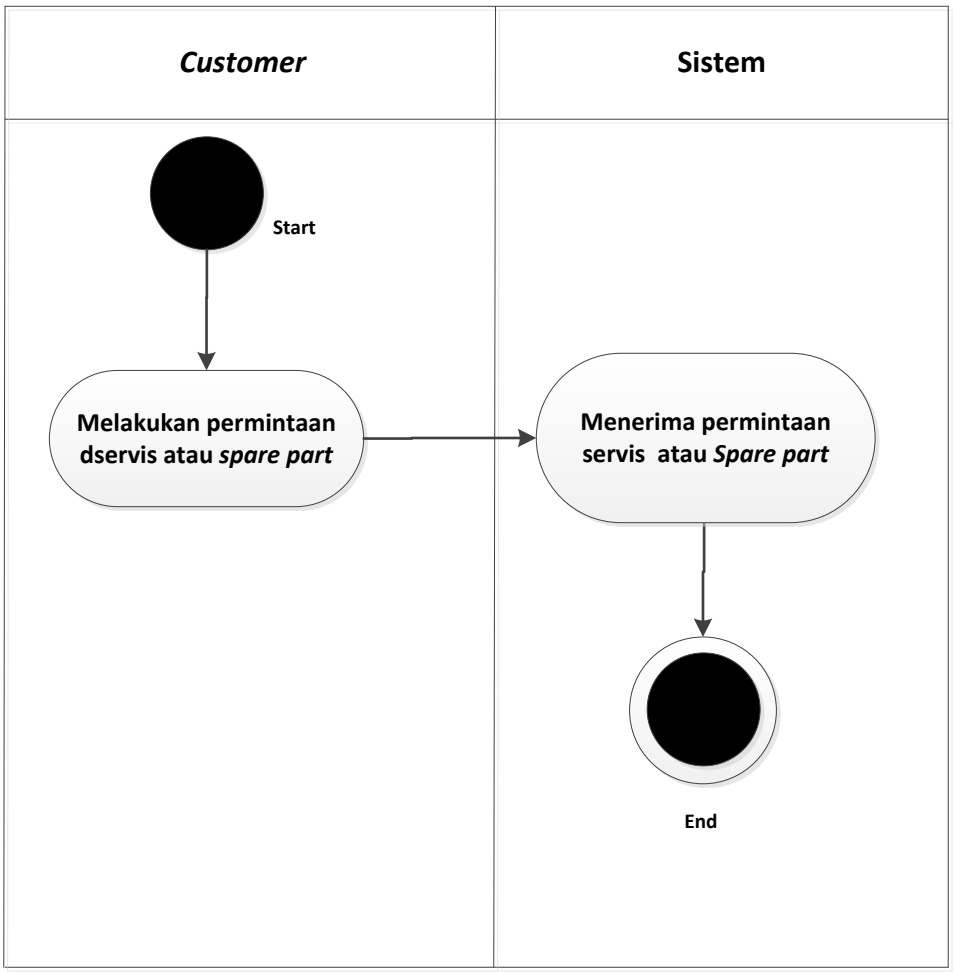
Tabel 4.7 Skenario *use case* pembelian *spare part*

Identifikasi	
Judul	Pembelian *spare part*
No. Use Case	UC-05
Deskripsi Use Case	Melakukan pembelian untuk kebutuhan gudang
Aktor	Admin gudang
Trigger	Admin bengkel meminta barang
Kondisi awal	Admin gudang siap melakukan pembelian
Skenario Utama	
Aksi- Aktor	**Reaksi Sistem**
1. Admin gudang melakukan pembelian *Spare part* dari *supplier* dan pembayaran dengan cara tunai	
	2. Menyimpan *Spare part* di gudang

3. Activity Diagram

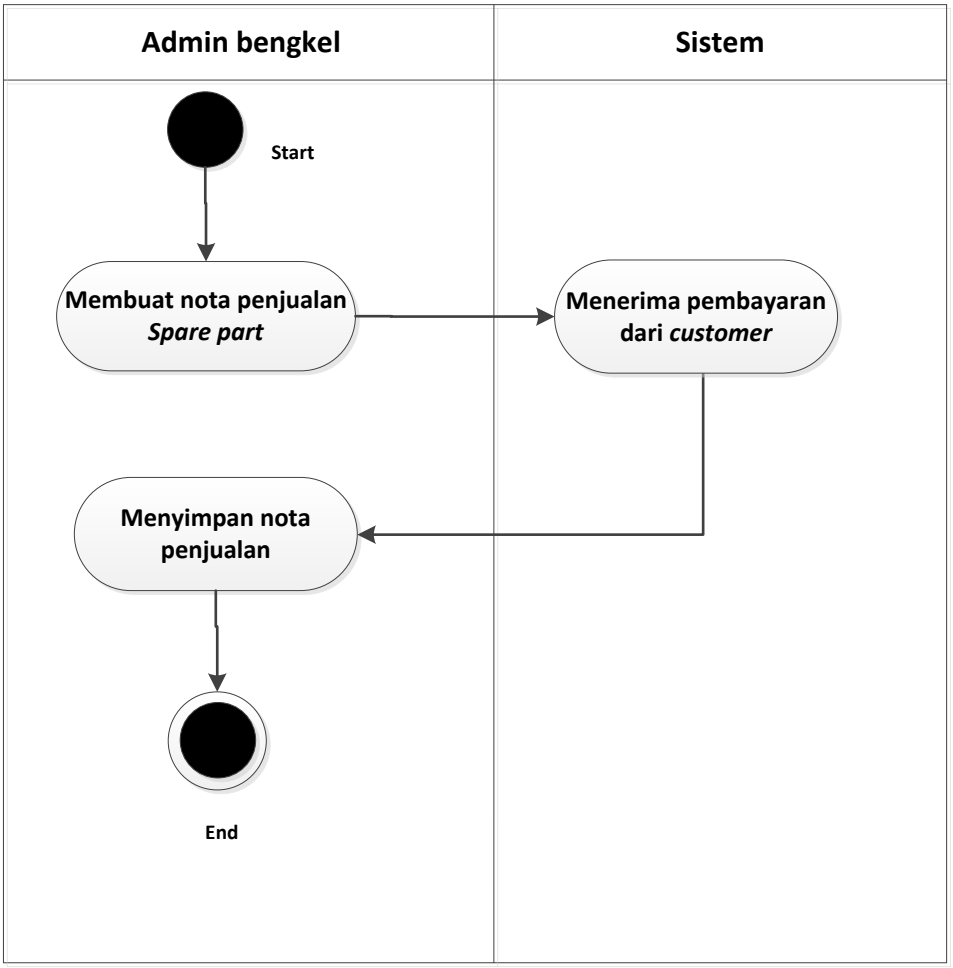
Activity diagram adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Berikut *activity diagram* proses servis pada sistem yang berjalan

4.1.1.3.6 Activity diagram permintaan servis dan *spare part*



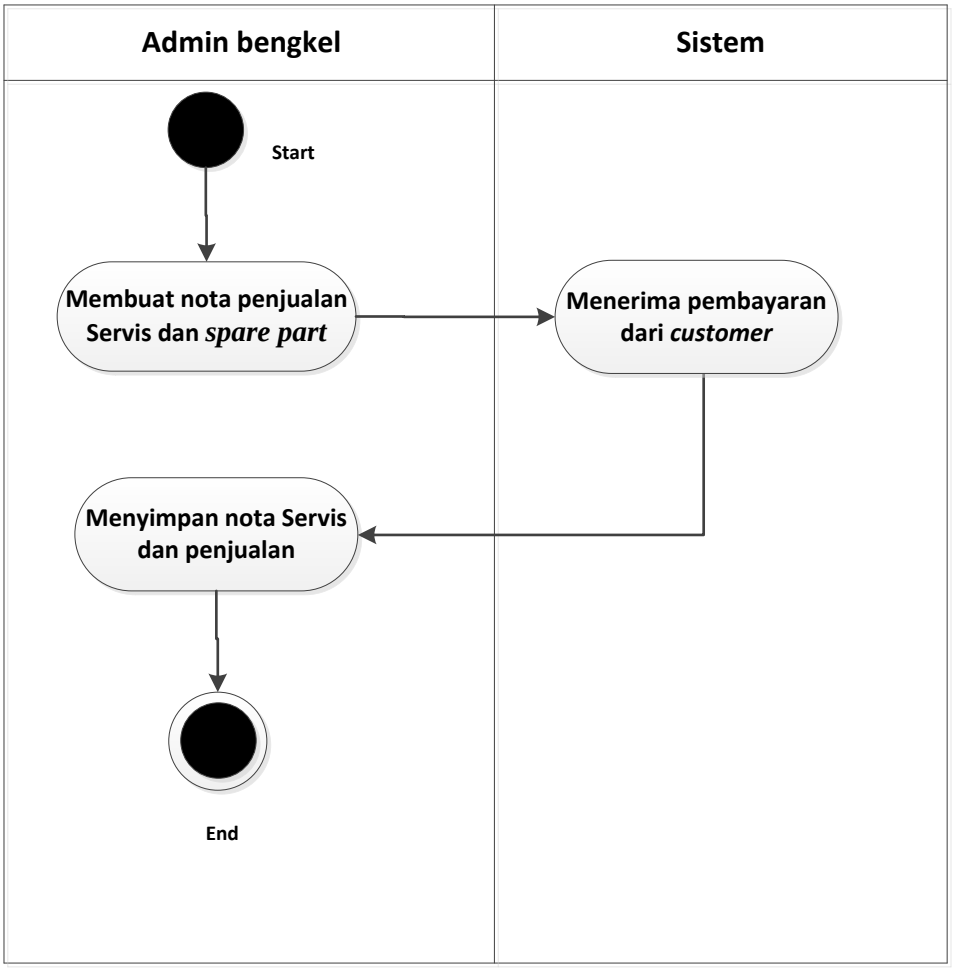
Gambar 4. 5 *Activity diagram* permintaan servis dan *spare part*

4.1.1.3.7 Activity diagram permintaan penjualan *spare part*



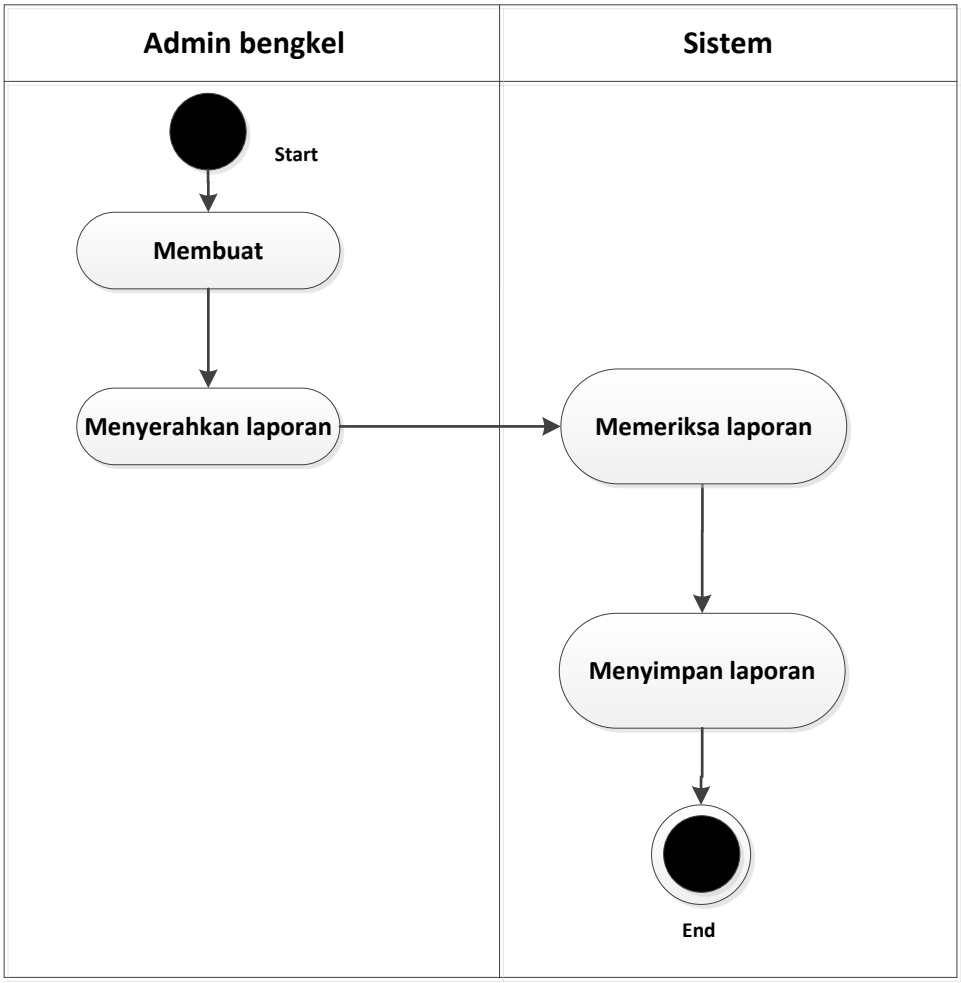
Gambar 4. 6 *Activity diagram* permintaan penjualan *spare part*

4.1.1.3.8 *Activity* diagram transaksi servis dan penjualan



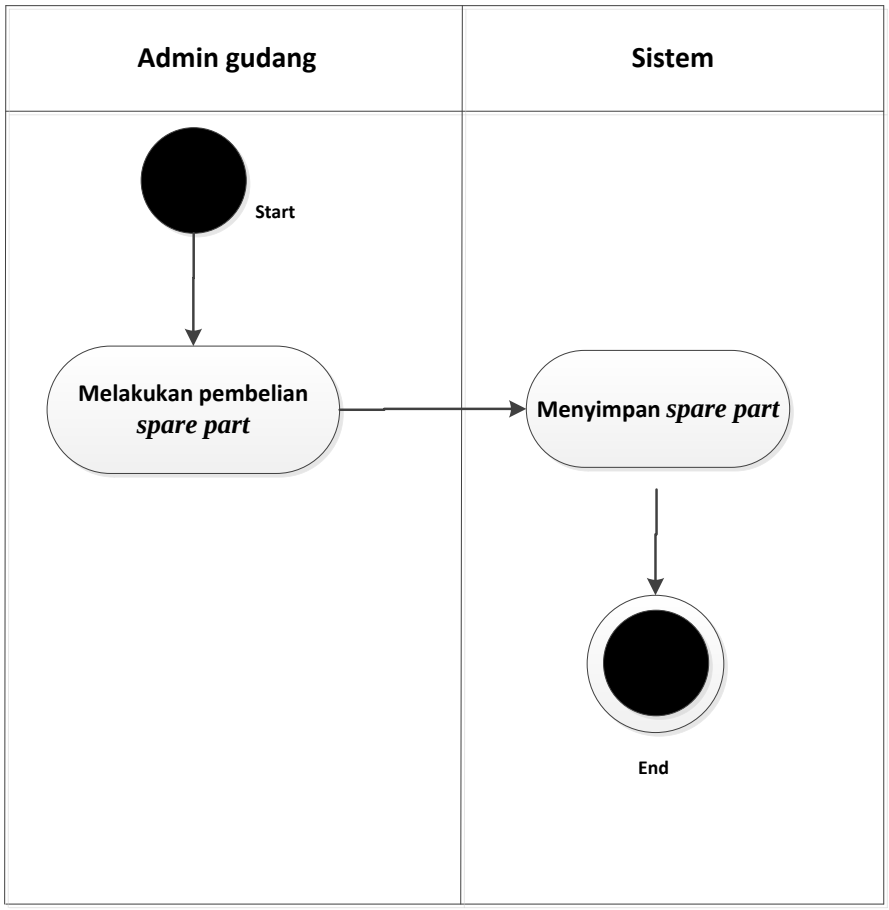
Gambar 4. 7 *Activity* diagram transaksi servis dan penjualan

4.1.1.3.9 *Activity* diagram pelaporan



Gambar 4. 8 *Activity* diagram pelaporan

4.1.1.3.10 *Activity* Diagram pembelian *spare part*



Gambar 4. 9 *Activity* diagram pembelian *spare part*

4.2 Evaluasi Proses Bisnis yang Sedang Berjalan

Setelah dilakukan penelitian pada proses bisnis berjalan di Dealer Rajamandala Motor, terdapat beberapa kelemahan yang ditemukan pada proses yang sedang berjalan. Kelemahan-kelemahan tersebut adalah sebagai berikut:

Tabel 4.8 Evaluasi proses bisnis yang berjalan

No	Permasalahan	Penyelesaian
1.	Proses pengelolaan dan penghitungan transaksi pelayanan servis, penjualan,	Dengan merancang sistem informasi untuk memberikan

	pembelian *spare part* masih dicatat secara manual dan proses penghitungannya menggunakan kalkulator, sehingga sering terjadi kesalahan data transaksi dan membutuhkan waktu yang cukup lama.	kemudahan dalam melakukan proses penghitungan transaksi dengan cepat dan tepat.
2.	Kesulitan dalam pengambilan keputusan untuk menentukan garansi servis yang diberikan oleh pihak bengkel karena tidak terdapatnya *database customer* dengan riwayat servis kendaraan sehingga terjadi kesalah pahaman antara *customer* dan pihak bengkel.	Dengan merancang sistem informasi untuk mengelola *database* secara baik sehingga membantu dalam kegiatan proses bisnis dan pengambilan keputusan.
3.	Kurangnya informasi pengawasan persediaan *spare part* 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*.	Dengan merancang sistem informasi untuk persediaan *spare part* yang dilengkapi dengan *database* yang dapat meningkatkan efisiensi waktu, dan meningkatkan produktifitas Dealer.

4.	Kepala Toko merasa kesulitan dalam memeriksa arsip laporan yang dikarenakan tidak sesuainya data transaksi pembelian *spare part*, penjualan dan pelayanan servis erta sering terjadi kehilangan data dan dokumen karena penyimpanannya masih dalam bentuk arsip.	Dengan merancang sistem informasi untuk mempermudah dalam membuat data laporan seluruh kegiatan transaksi dan menyimpannya dalam *database* sehingga mudah untuk mencari dan memeriksanya.

4.3 Perancangan Sistem

Perancangan sistem yaitu memperbaiki berbagai fungsi didalam sistem yang sedang berjalan agar menjadi lebih efisien ,mengubah sasaran sistem yang sedang berjalan, merancang atau mengganti *output* yang sedang digunakan, untuk mencapai tujuan yang sama dengan seperangkat *input* yang lain. Pada bagian ini penyusun akan memberikan usulan yang merupakan sistem informasi secara komputerisasi yang diharapkan akan membantu dan mempermudah pekerjaan.

4.3.1 Tujuan 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, persediaan *spare part* dan servis motor yang semuanya terintegrasi dengan *database* dan aplikasi tersebut dapat mencapai sasaran kebutuhan dan tujuan dari perusahaan yaitu kepuasan *customer*.

4.3.2 Gambaran Umum Sistem yang Diusulkan

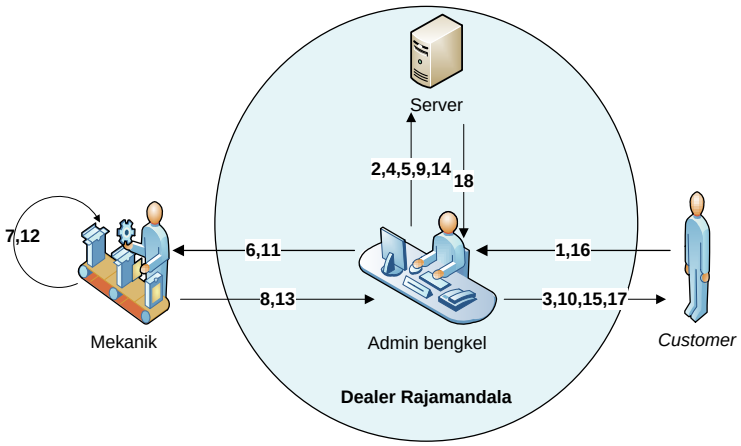
Gambaran umum tentang sistem yang diusulkan pada proses perancangan ini adalah suatu sistem yang merubah seluruh proses bisnis yang manual menjadi terkomputerisasi. Proses bisnis yang dimaksud adalah pengolahan data *customer*, transaksi penjualan pembelian *spare part*, pelayanan servis, hingga proses laporan kepada kepala toko 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.

4.3.3 Perancangan Prosedur yang Diusulkan

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).

4.3.3.1 Proses Bisnis Diusulkan

Proses bisnis servis motor yang diusulkan adalah sebagai berikut:



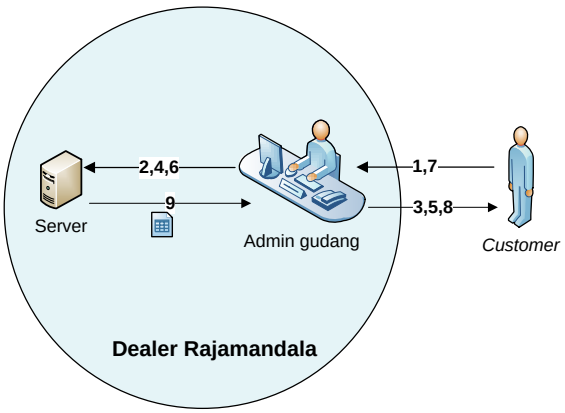
Gambar 4. 10 Proses bisnis servis yang diusulkan

Keterangan :

1. *Customer* datang ke bengkel dan mendaftarkan servis kepada admin bengkel dengan menyerahkan SIM atau KTP dan STNK.
2. Admin bengkel melakukan *login* ke sistem.
3. Admin bengkel menanyakan kebutuhan dan keluhan *customer*.
4. Admin bengkel melakukan pemeriksaan di *database customer* dan kendaraan. Jika di *database* tidak ditemukan, maka dilakukan input data *customer* baru dan kendaraan pada *database*
5. Admin bengkel membuat *work order* sesuai keluhan dan permintaan *customer*.
6. Admin bengkel memberikan *work order* kepada mekanik..
7. Mekanik melakukan pemeriksaan kendaraan sesuai *work order*.

8. Mekanik melaporkan jenis kerusakan dan kebutuhan *spare part*.
9. Admin bengkel membuat form estimasi (perkiraan biaya servis dan harga *spare part*).
10. Admin bengkel menyerahkan form estimasi ke *customer*.
11. Admin bengkel menyerahkan form estimasi dan *spare part* kepada mekanik.
12. Mekanik melakukan servis sesuai estimasi.
13. Mekanik konfirmasi pekerjaan selesai ke admin bengkel.
14. Admin bengkel membuat faktur penjualan(biaya servis dan harga *spare part*).
15. Admin mengkonfirmasi pekerjaan selesai.
16. *Customer* melakukan pembayaran tunai kepada admin bengkel.
17. Admin bengkel menyerahkan faktur penjualan.
18. Sistem memberikan respon sesuai dengan data yang diinginkan.

Proses bisnis penjualan *spare part* yang diusulkan adalah sebagai berikut :

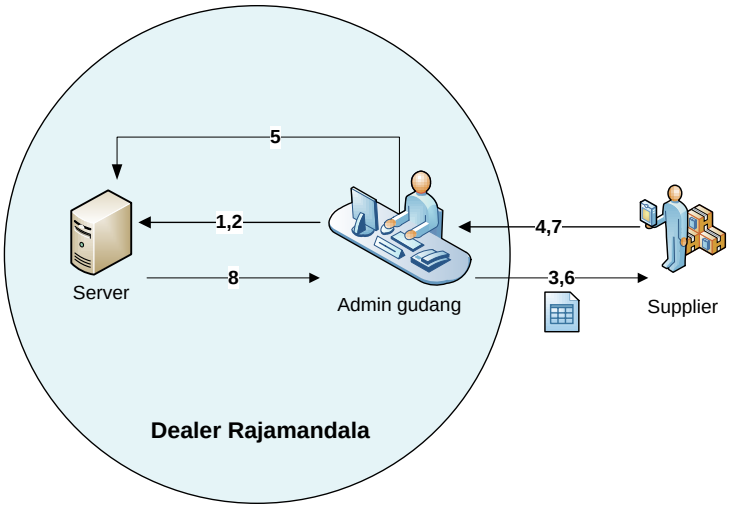


Gambar 4. 11 Proses bisnis penjualan yang diusulkan

Keterangan :

1. *Customer* datang ke bengkel melakukan pembelian kepada admin gudang.
2. Admin gudang melakukan login ke sistem.
3. Admin gudang menanyakan kebutuhan *spare part customer*.
4. Admin gudang melakukan pemeriksaan ketersediaan dan harga *spare part* pada *database.*
5. Admin gudang mengkonfirmasi harga *spare part.*
6. Admin gudang membuat faktur penjualan
7. *Customer* melakukan pembayaran tunai kepada admin gudang.
8. Admin gudang menyerahkan faktur penjualan dan *spare part* kepada *customer*.
9. Sistem memberikan respon sesuai dengan data yang diinginkan.

Proses bisnis pembelian yang diusulkan adalah sebagai berikut :

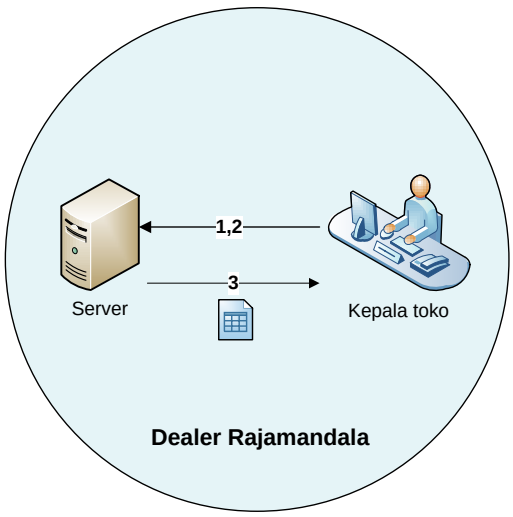


Gambar 4. 12 Proses bisnis pembelian yang diusulkan

Keterangan :

1. Admin gudang melakukan login ke sistem.
2. Admin gudang membuat pemesanan pembelian (PO).
3. Admin gudang mengirimkan PO kepada *supplier*.
4. Supplier mengirimkan *spare part* sesuai PO kepada admin gudang.
5. Admin gudang menerima barang dan faktur pembelian dan membuat penyesuaian penerimaan pemesanan pembelian.
6. Admin gudang melakukan pembayaran secara tunai kepada *supplier.*
7. *Supplier* menyerahkan faktur pembelian kepada admin gudang.
8. Sistem memberikan respon sesuai dengan data yang diinginkan

Proses bisnis *pelaporan* yang diusulkan adalah sebagai berikut :



Gambar 4. 13 Proses bisnis pelaporan yang diusulkan

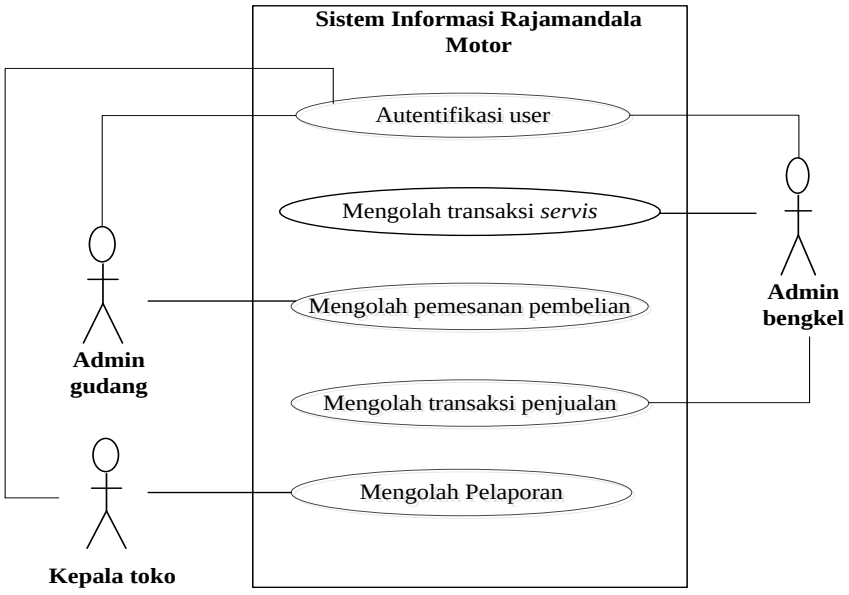
Keterangan :

1. Kepala toko melakukan login ke sistem
2. Kepala toko melakukan pelaporan dengan memasukkan data periode laporan dan jenis laporan yang diinginkan.
3. Sistem secara terkomputerisasi akan menampilakan laporan sesuai dengan kondisi yang diinginkan.

5 *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. 14 *Use case* diagram yang diusulkan

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 4.8 Identifikasi u*se case* sistem yang diusulkan

No	***Use Case***	**Deskripsi**
UC-01	Autentifikasi user	Fungsionalitas untuk menentukan hak akses user pada saat login ke sistem
UC-02	Mengolah transaksi servis	Fungsionalitas untuk mengolah data transaksi servis
UC-03	Mengolah pemesanan pembelian	Fungsionalitas untuk melakukan penambahan data transaksi pemesanan pembelian
UC-04	Mengolah transaksi penjualan	Fungsionalitas untuk mengolah data transaksi penjualan.
UC-05	Mengolah pelaporan	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**
P-01	Admin bengkel	Aktor yang berperan mengatur proses administrasi dan transaksi servis di bengkel
P-02	Admin gudang	Aktor yang berfungsi mengatur proses

		pemesanan pembelian, penjualan dan persediaan *spare part*
P-03	Kepala toko	Aktor yang berfungsi memeriksa semua laporan dan bertanggung jawab penuh dengan seluruh kegiatan proses bisnis di Dealer

6 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 :

4.3.3.1.1 Skenario *use case autentifikasi user*

Tabel 4.10 Skenario *use case* autentifikasi user

Identifikasi	
Judul	Autentifikasi user
No. Use case	UC-01
Deskripsi use case	Aktor login ke sistem untuk menentukan hak aksesnya
Aktor	Admin bengkel, Admin gudang, Kepala toko
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 keabsahan *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.*

4.3.3.1.2 Skenario *use case* mengolah transaksi servis

Tabel 4.11 Skenario *use case* mengolah transaksi servis

Identifikasi	
Judul	Mengolah transaksi servis
No. Use case	UC-02
Deskripsi use case	Aktor menambah data transaksi servis

Aktor	Admin bengkel
Trigger	Aktor memilih menu untuk menambah transaksi servis
Kondisi awal	Aplikasi menampilkan sebuah *form* transaksi servis
Skenario Utama	
Aksi- aktor	**Reaksi sistem**
1. Aktor mengisi data jenis servis yang baru	
2. Aktor men-*submit* data transaksi servis yang baru.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data telah diisi dan sesuai.
	4. Data transaksi servis ditampilkan pada *form* dengan hasil kalkulasi.
	5. Sistem menambahkan data transaksi servis yang baru pada *database*
	6. Sistem memperbaharui data *spare part* dengan data pada transaksi servis jika ada pengeluaran *spare part*
	7. *Form* transaksi servis dicetak untuk kemudian diprint

Kondisi akhir skenario utama	Sistem menyimpan data transaksi servis yang baru pada *database* dan memperbaharui data *spare part*
Skenario Alternatif – 1	
Aksi- aktor	**Reaksi sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* dimana ada *field-field* pada form yang seharusnya diisi tetapi dikosongkan atau ada data yang tidak sesuai.	2. *Form* pengisian data transaksi servis ditampilkan dengan pesan bahwa ada data *field* yang harus diisi namun masih kosong, atau ada data yang tidak sesuai.
	3. Data servis baru tidak disimpan pada *database.*
Kondisi akhir skenario alternatif -1	*Form* penambahan data transaksi servis ditampilkan

4.3.3.1.3 Skenario use case mengolah pemesanan pembelian

Tabel 4.12 Skenario *use case* mengolah pemesanan pembelian

Identifikasi	
Judul	Mengolah pemesanan pembelian
No. Use case	UC-03
Deskripsi use case	Aktor menambah data transaksi pemesanan pembelian.
Aktor	Admin gudang

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 men-*submit* data pemesanan pembelian yang baru.	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data telah diisi dan sesuai.
	4. Data pembelian ditambahkan atau dirubah pada *database*.
	5. Data *spare part* diperbaharui sesuai dengan data pembelian.
Kondisi akhir skenario utama	Data pemesanan pembelian dan *spare part* ditambahkan atau dirubah pada *database*
Skenario Alternatif – 1	

Aksi- aktor	**Reaksi sistem**
1. Aktor men-*submit* data dengan menekan sebuah *button* dimana 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

4.3.3.1.4 Skenario *use case* mengolah transaksi penjualan

Tabel 4.13 Skenario *use case* mengolah transaksi penjualan

Identifikasi	
Judul	Mengolah transaksi penjualan
No. Use case	UC-04
Deskripsi use case	Aktor menambah data transaksi penjualan.
Aktor	Admin gudang
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 men-*submit* 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 *spare part* sesuai dengan data pada transaksi penjualan
	7. Form transaksi penjualan dicetak untuk kemudian di*print*
Kondisi akhir skenario utama	Sistem menyimpan data transaksi yang baru pada *database* dan memperbaharui data *spare part*

Skenario Alternatif – 1	
Aksi- aktor	**Reaksi sistem**

1. Aktor men-*submit* 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 penjualan ditampilkan 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

4.3.3.1.5 Skenario use case mengolah pelaporan

Tabel 4.14 Skenario *use case* mengolah pelaporan

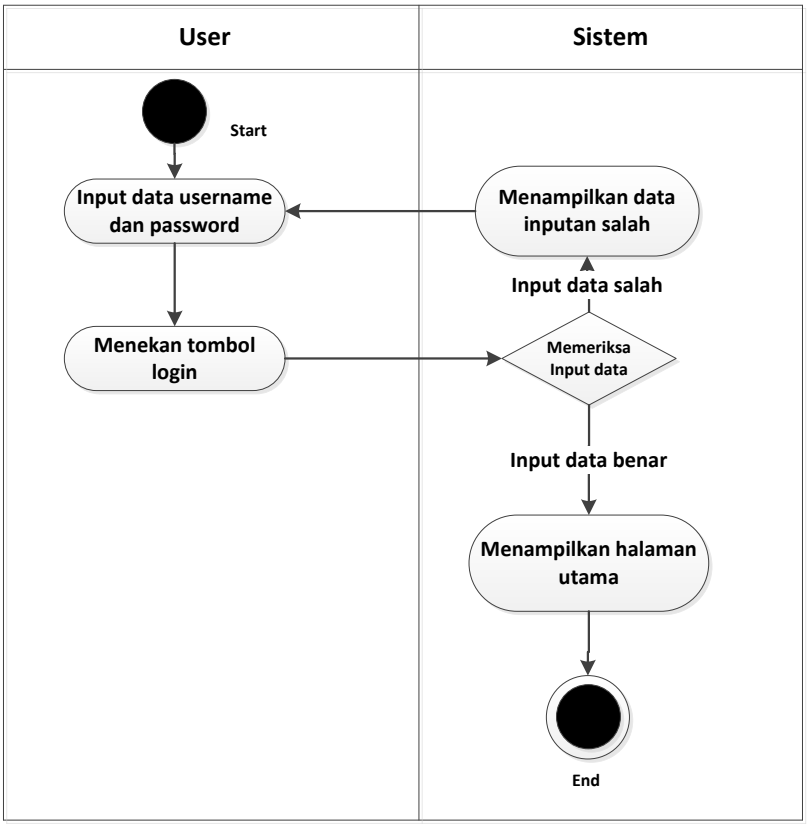
Identifikasi	
Judul	Mengolah laporan
No. Use case	UC-05
Deskripsi use case	Aktor mencetak laporan yang terjadi pada *range* waktu yang diinginkan.
Aktor	Kepala toko.
Trigger	Aktor memilih menu untuk mencetak laporan.
Kondisi awal	Aplikasi menampilkan *form* untuk menentukan periode dan jenis laporan 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 sama sekali.	
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.

7 *Activity Diagram*

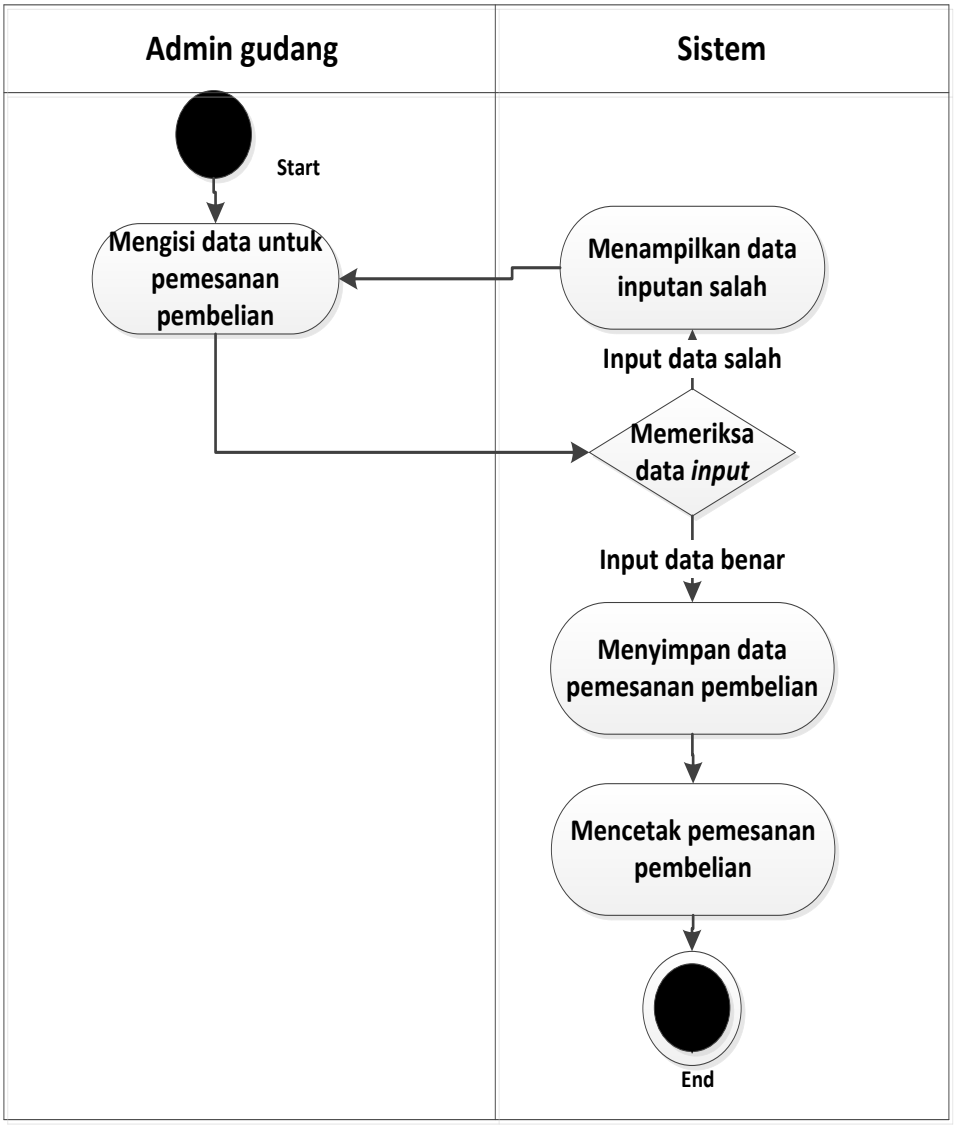
Activity diagram (diagram aktivitas) adalah diagram yang menggambarkan aliran fungsionalitas dari sistem. Berikut *activity diagram* proses servis pada sistem yang diusulkan :

4.3.3.1.6 *Activity* Diagram autentikasi *user*



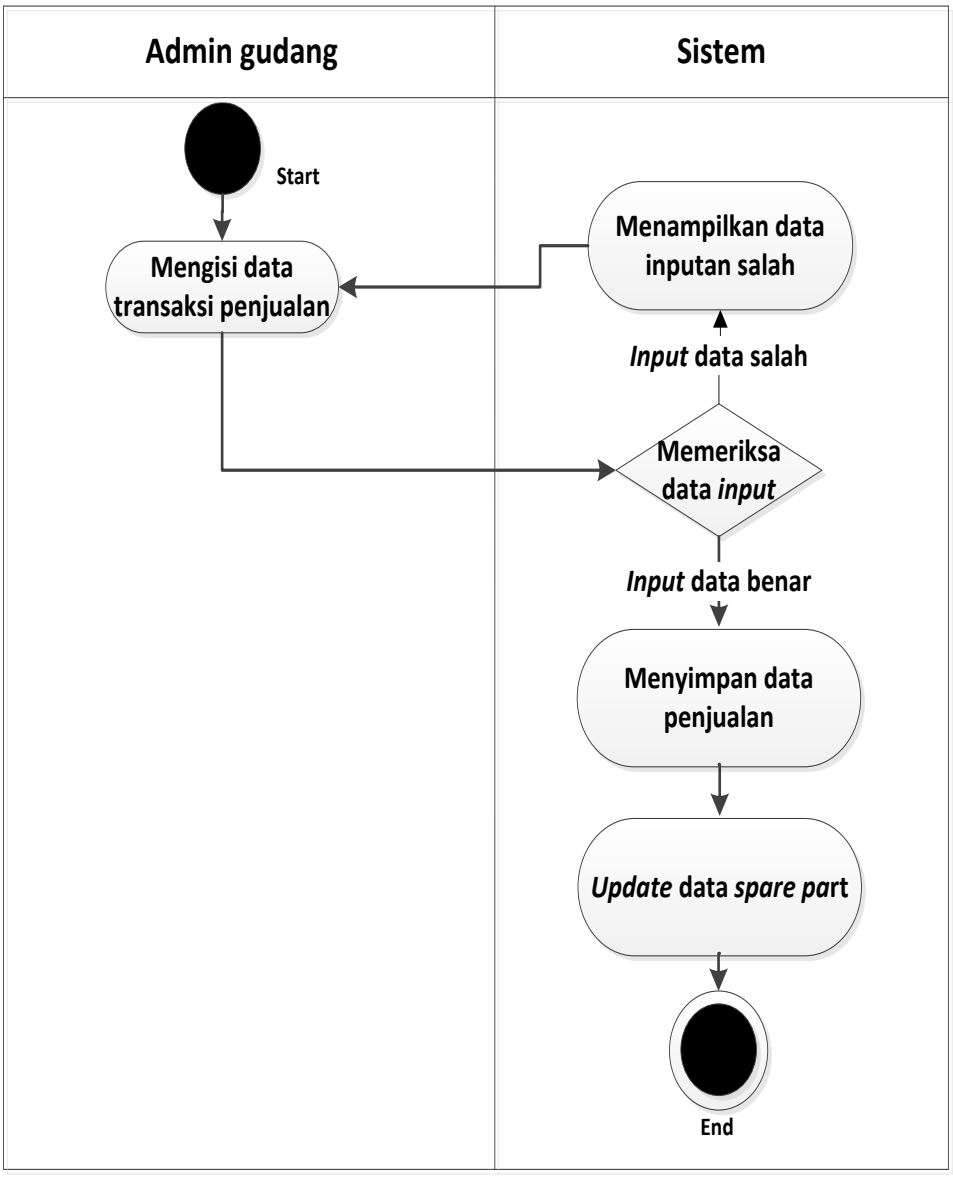
Gambar 4. 15 *Activity diagram* autentifikasi *user*

4.3.3.1.7 Activity Diagram mengolah pemesanan pembelian



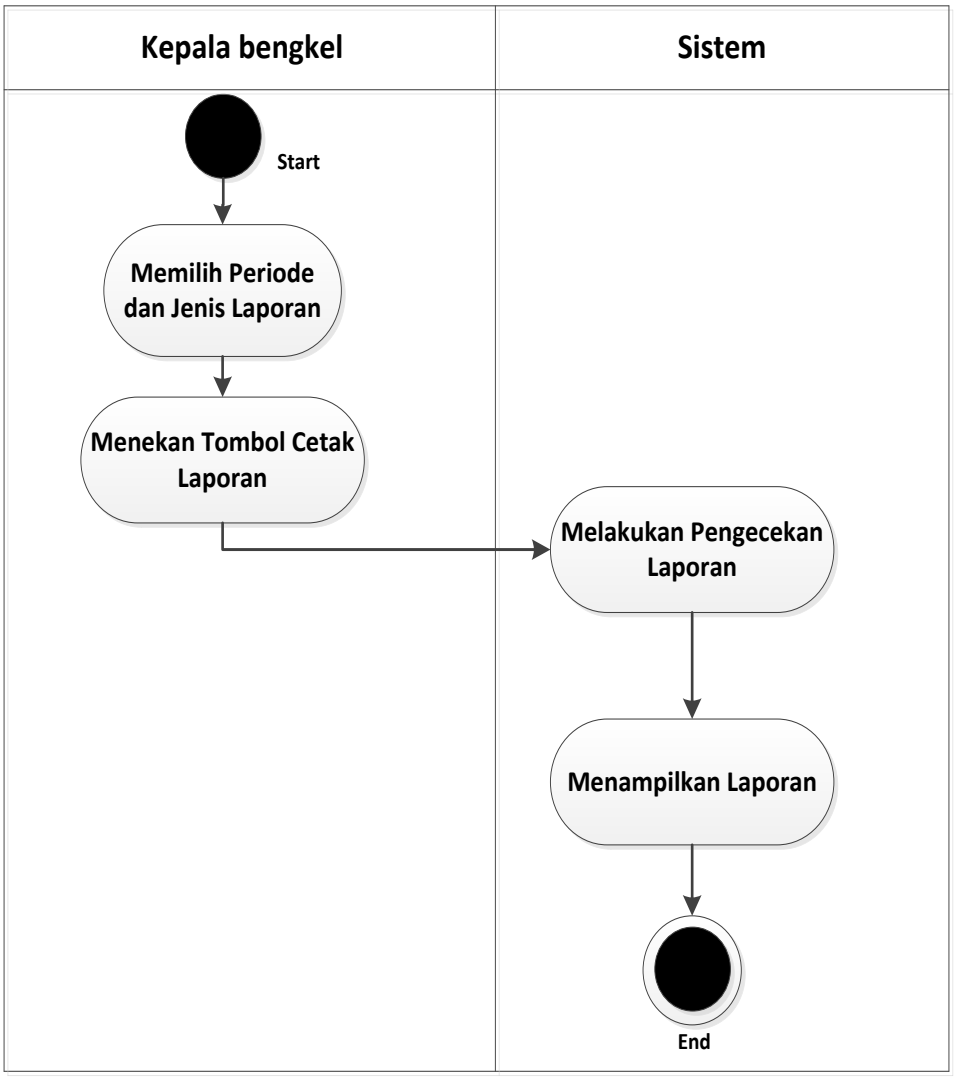
Gambar 4. 16 *Activity* diagram mengolah pemesanan pembelian

4.3.3.1.8 Activity Diagram mengolah transaksi penjualan



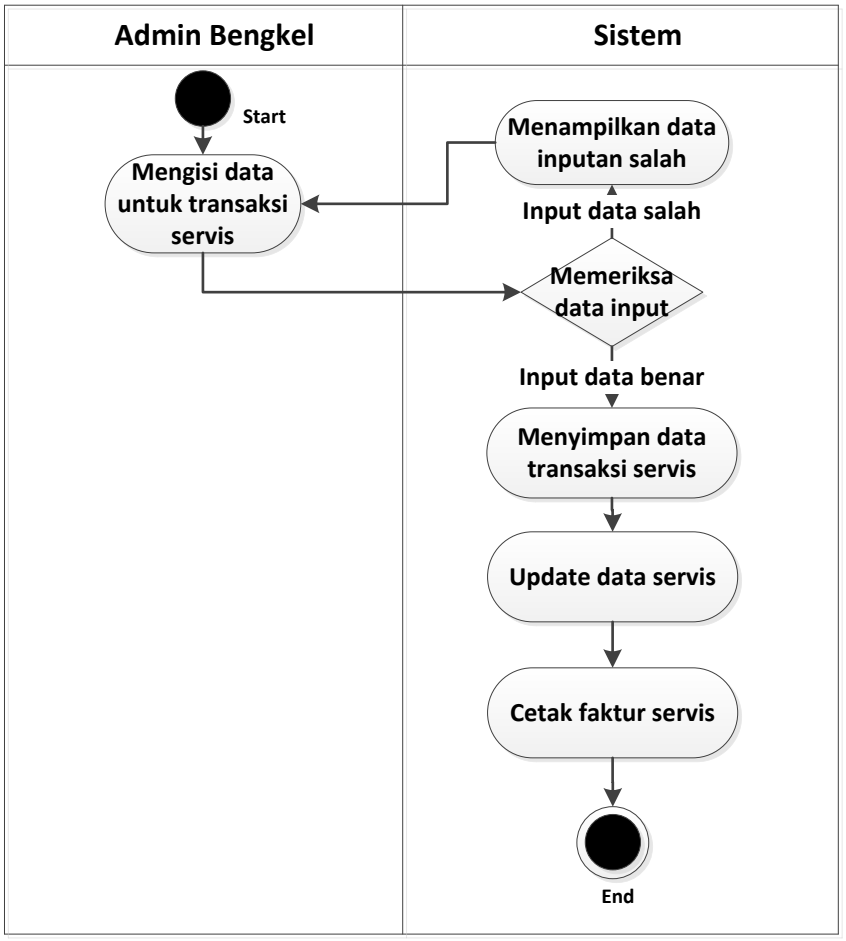
Gambar 4. 17 *Activity* diagram mengolah transaksi penjualan

4.3.3.1.9 Activity Diagram mengolah pelaporan



Gambar 4. 18 *Activity* diagram mengolah pelaporan

4.3.3.1.9 Activity Diagram mengolah transaksi servis



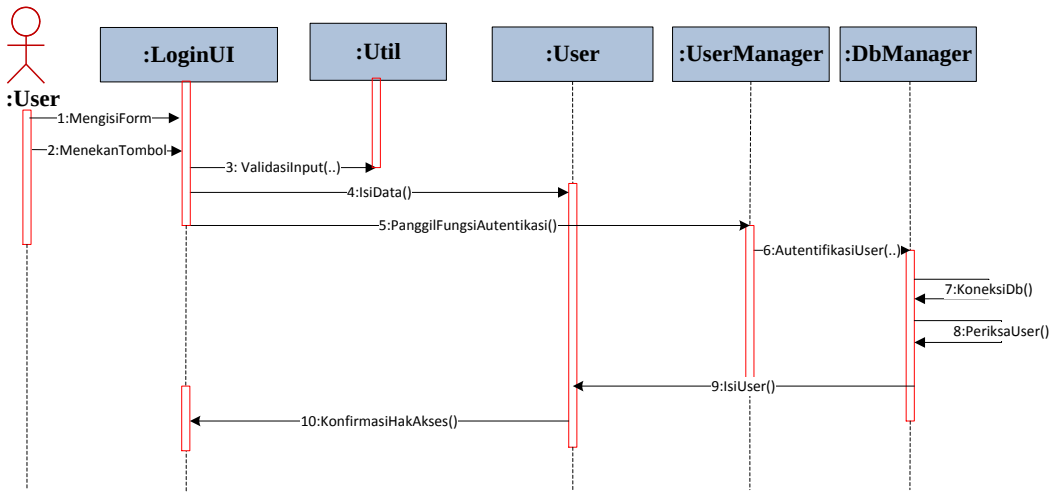
Gambar 4. 19 *Activity* diagram mengolah transaksi servis

8 *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 objek, 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 :

4.3.3.1.10 *Sequence* Diagram autentifikasi *user*



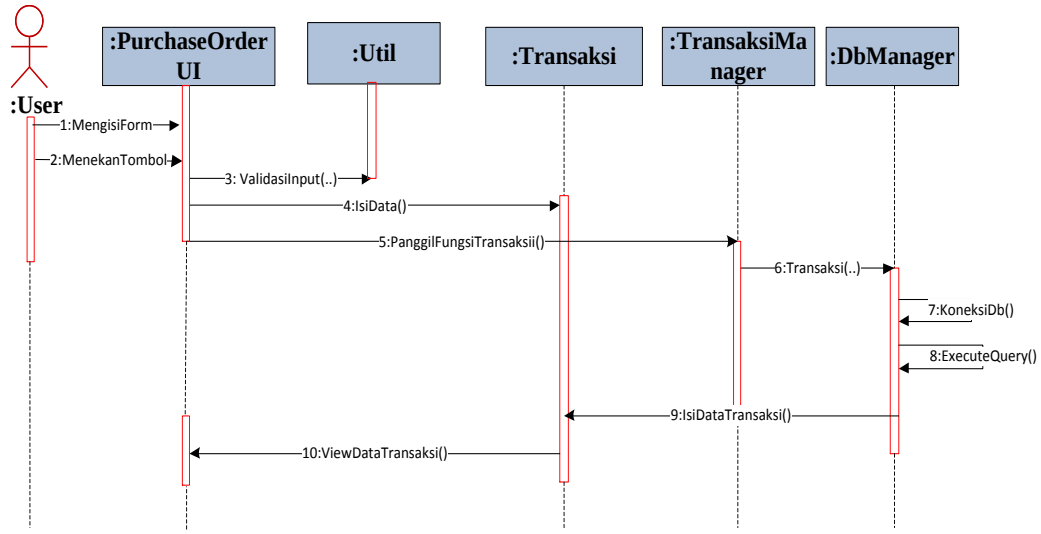
Gambar 4. 20 *Sequence* diagram Autentikasi *User*

Keterangan:

Pada Gambar 4.20 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. LoginUI
2. Util
3. User
4. UserManager
5. DbManager

4.3.3.1.11 *Sequence* Diagram mengolah pemesanan pembelian



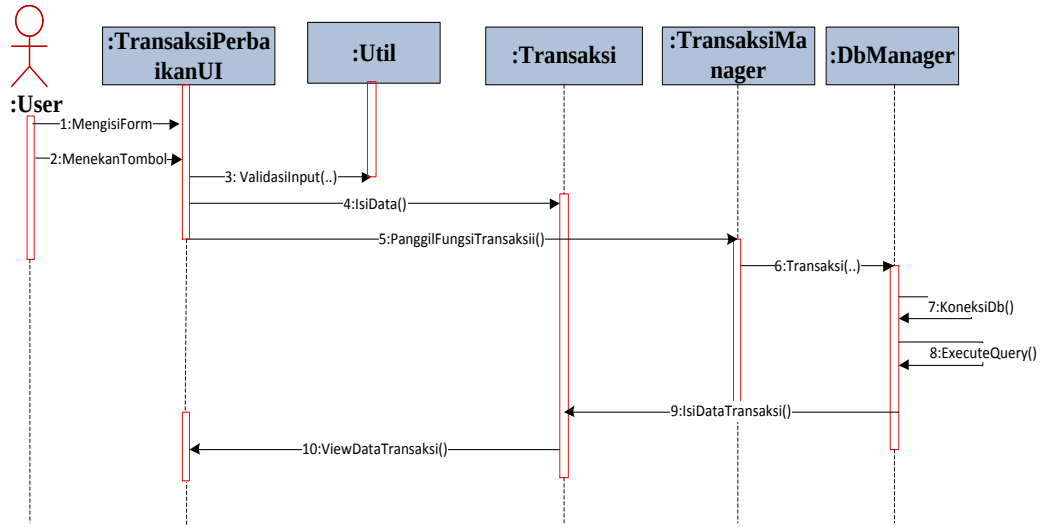
Gambar 4. 21 *Sequence* diagram Mengolah Pemesanan pembelian

Keterangan:

Pada Gambar 4.21 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. PurchaseOrderUI
2. Util
3. Transaksi
4. TransaksiManager
5. DbManager

4.3.3.1.12 *Sequence* Diagram mengolah transaksi servis



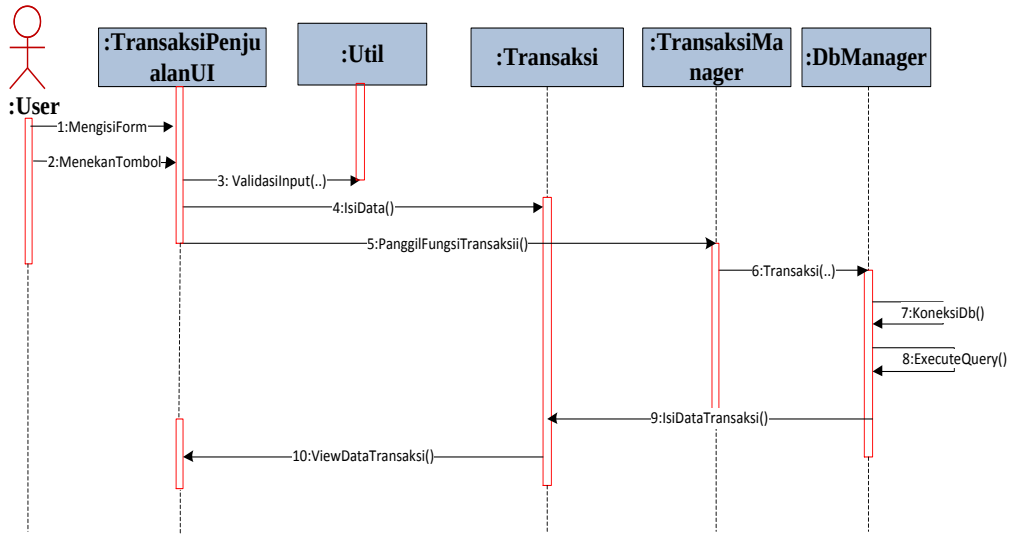
Gambar 4. 22 *Sequence* diagram Mengolah Transaksi Servis

Keterangan:

Pada Gambar 4.22 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. TransaksiServisUI
2. Util
3. Transaksi
4. TransaksiManager
5. DbManager

4.3.3.1.13 Sequence Diagram mengolah penjualan



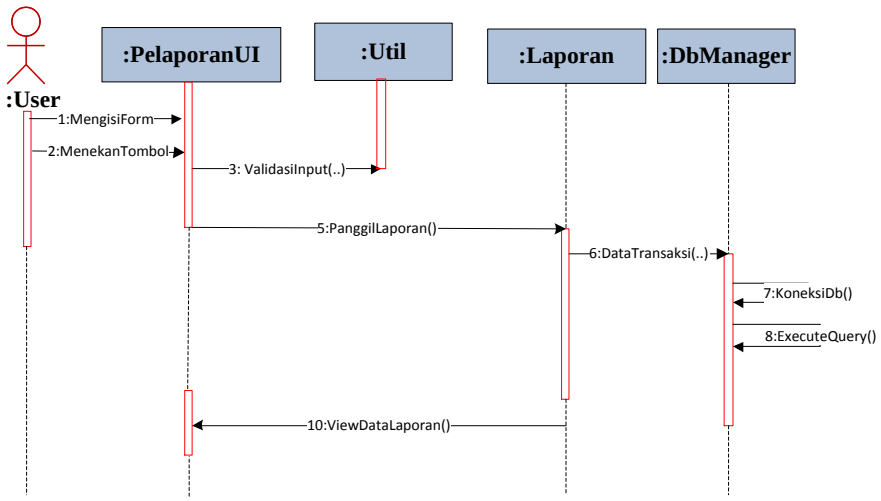
Gambar 4. 23 *Sequence* diagram Mengolah Transaksi Penjualan

Keterangan:

Pada Gambar 4.23 dapat dilihat bahwa ada 5 kelas yang saling berinteraksi, yakni:

1. TransaksiPenjualanUI
2. Util
3. Transaksi
4. TransaksiManager
5. DbManager

4.3.3.1.14 Sequence Diagram mengolah pelaporan



Gambar 4. 24 *Sequence* diagram Mengolah Pelaporan

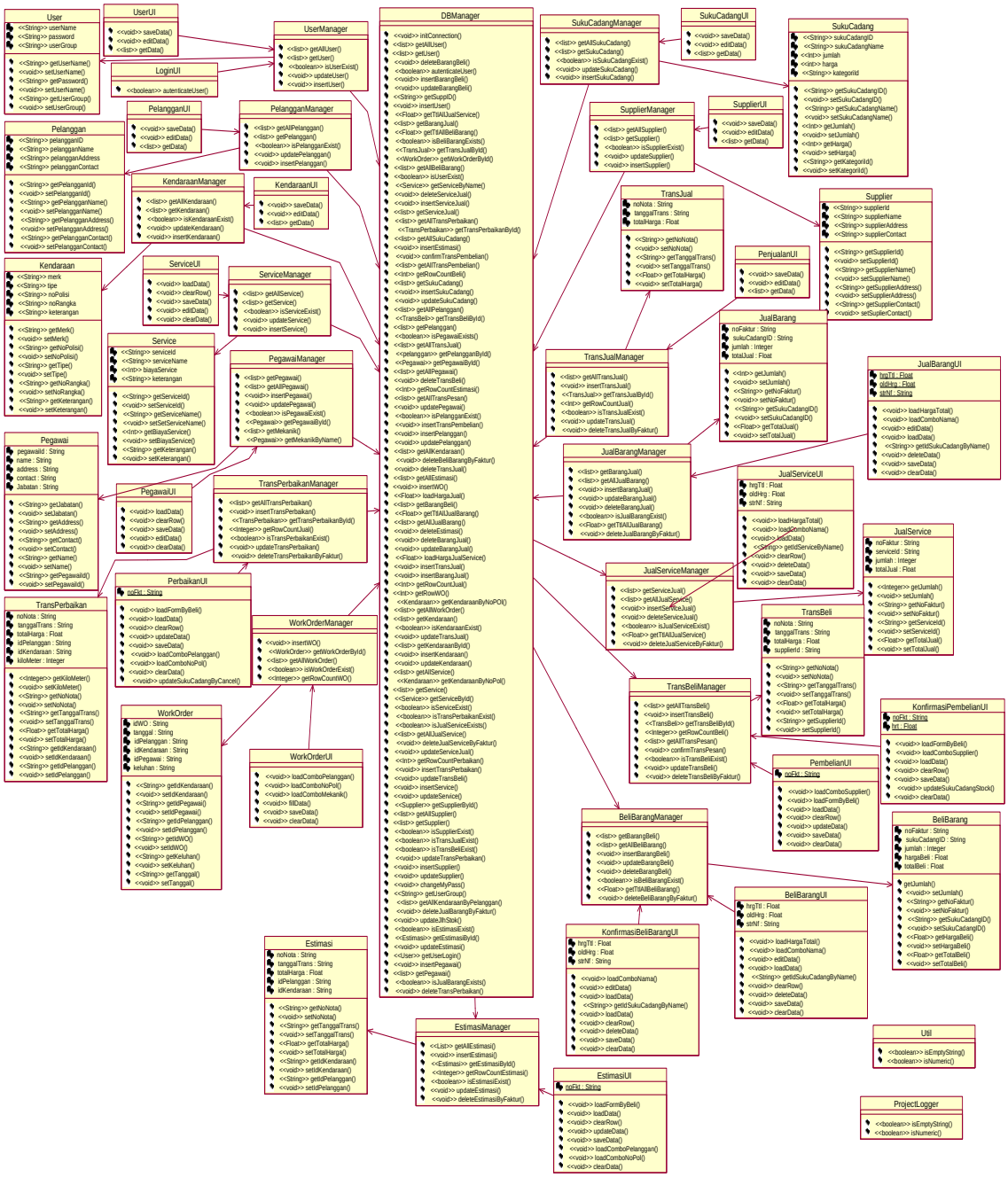
Keterangan:

Pada Gambar 4.24 dapat dilihat bahwa ada 4 kelas yang saling berinteraksi, yakni:

1. TransaksiPenjualanUI
2. Util
3. Laporan
4. DbManager

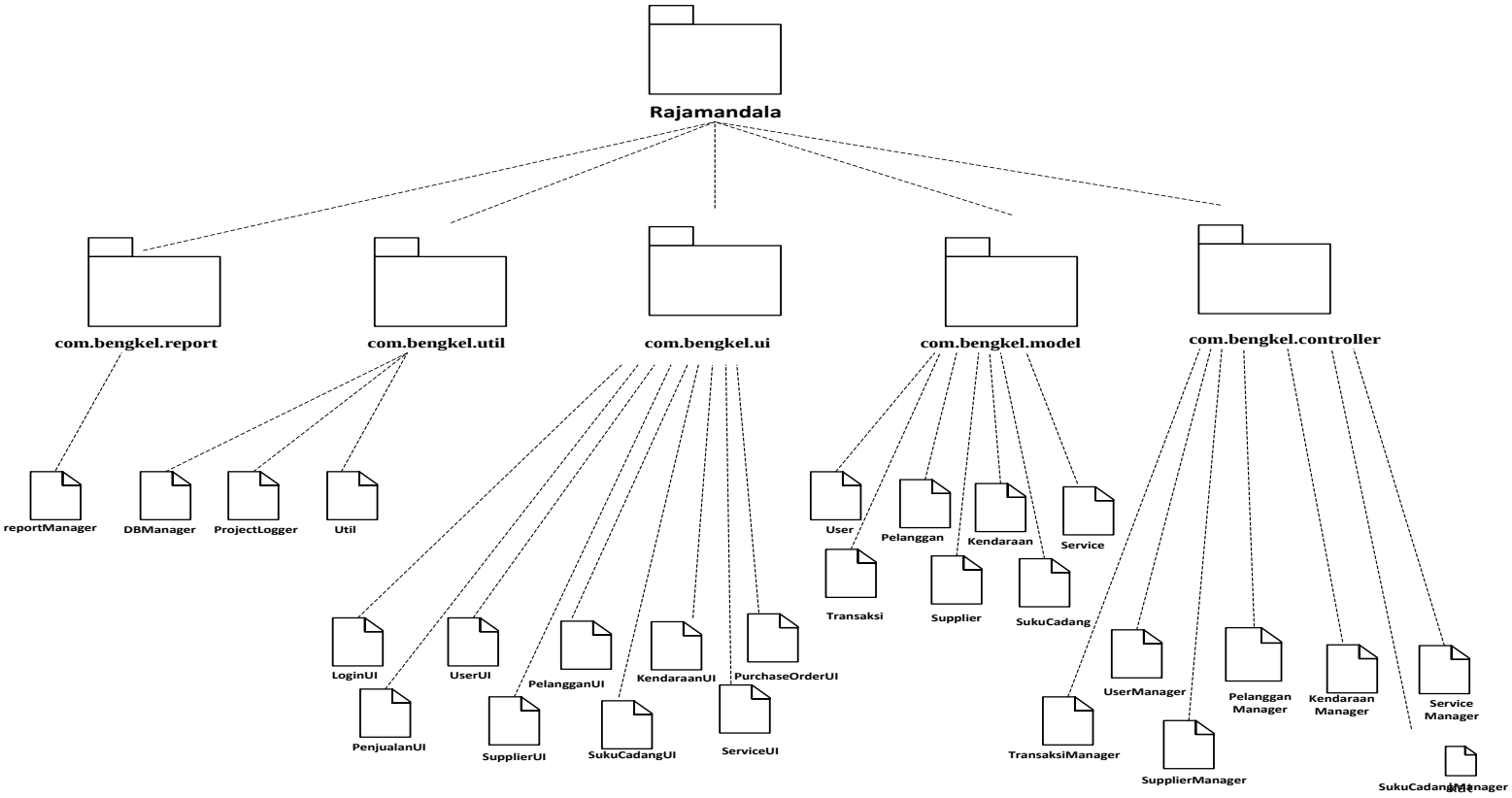
4.3.3.1.15 *Class* Diagram

Menggambarkan struktur statis *class* didalam 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 Raja Mandala Motor:



Gambar 4. 25 *Class* Diagram

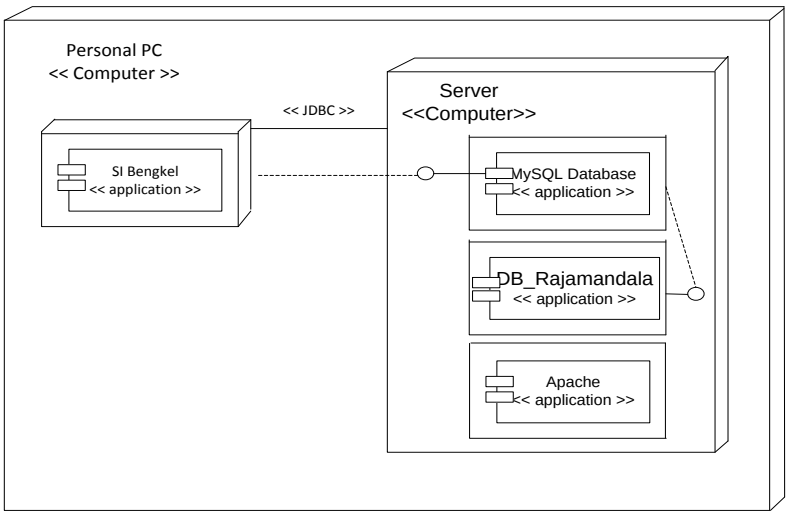
4.3.3.1.16 *Component* Diagram



Gambar 4. 26 Component Diagram

4.3.3.1.17 *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. 27 *Deployment* diagram sistem yang diusulkan

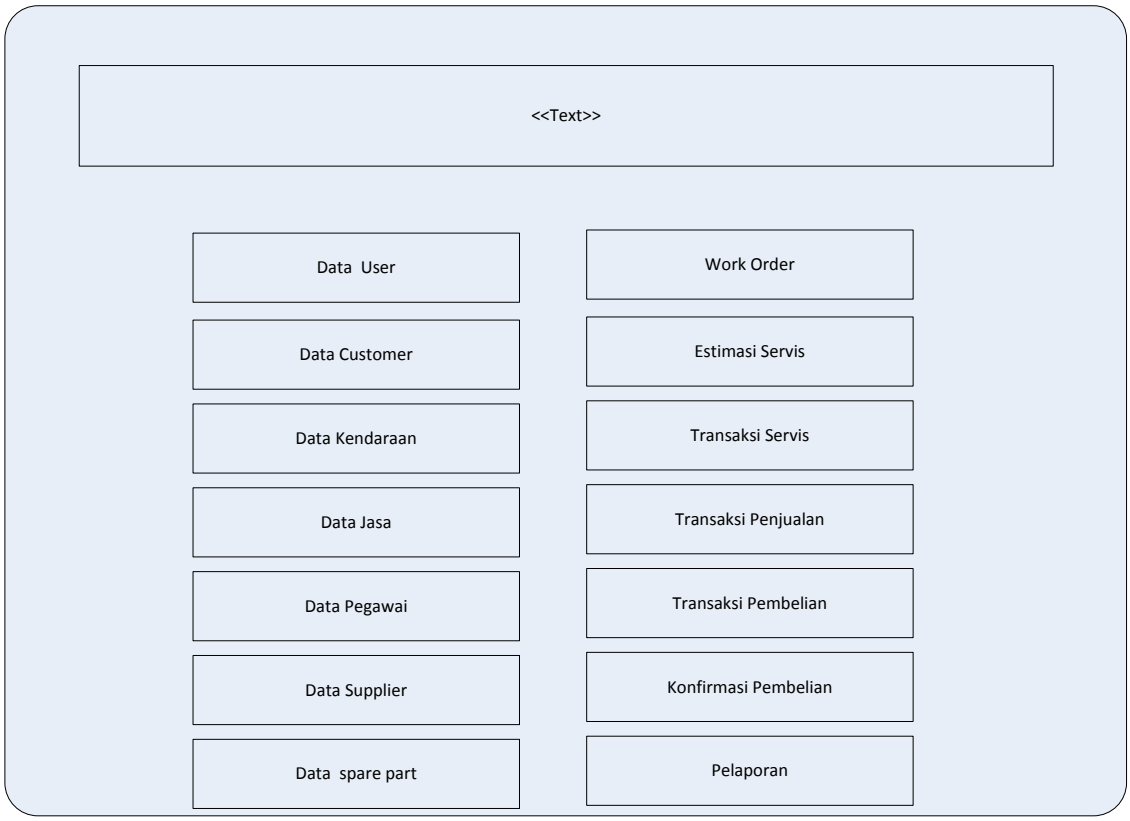
4.3.4 Perancangan Antar Muka

Perancangan antar muka merupakan perancangan yang dibuat sebelum program aplikasi dibuat, perancangan antar muka pada sistem informasi Dealer Rajamandala Motor adalah sebagai berikut:

4.3.4.1 Perancangan Struktur Menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian perangkat lunak.

Berikut rancangan menu perangkat lunak ini:



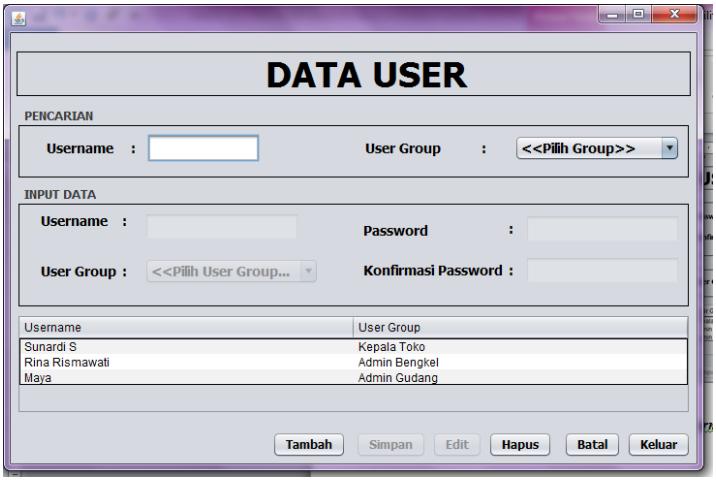
Gambar 4. 28 Perancangan *menu* sistem Dealer Rajamandala Motor

4.3.4.2 Perancangan Input

Dalam perancangan *input* ini, data yang dimasukkan akan mempengaruhi hasil yang ditampilkan. Adapun perancangan-perancangan *input* yang ada dalam perancangan ini adalah:

1. Rancangan *Master Data User*

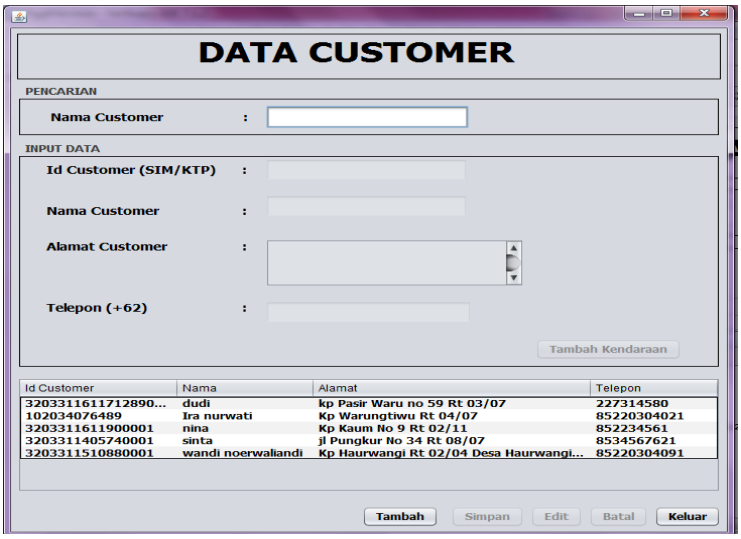
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data user.



Gambar 4. 29 Rancangan Master Data User

2. Rancangan *Master* Data *Customer*

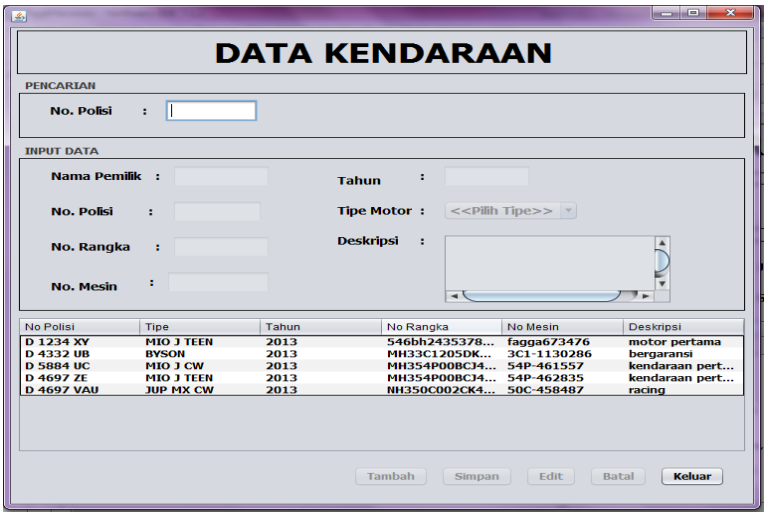
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data *customer*.



Gambar 4. 30 Rancangan *Master* Data *Customer*

3. Rancangan *Master* Data Kendaraan

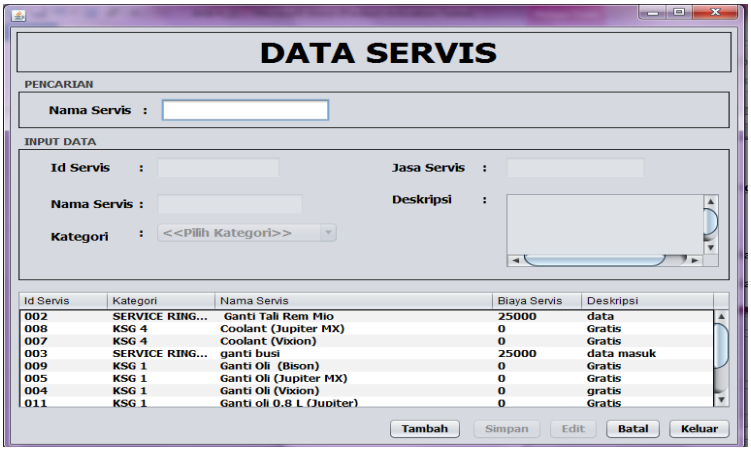
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data kendaraan.



Gambar 4. 31 Rancangan *Master* data kendaraan

4. Rancangan *Master* Data Servis

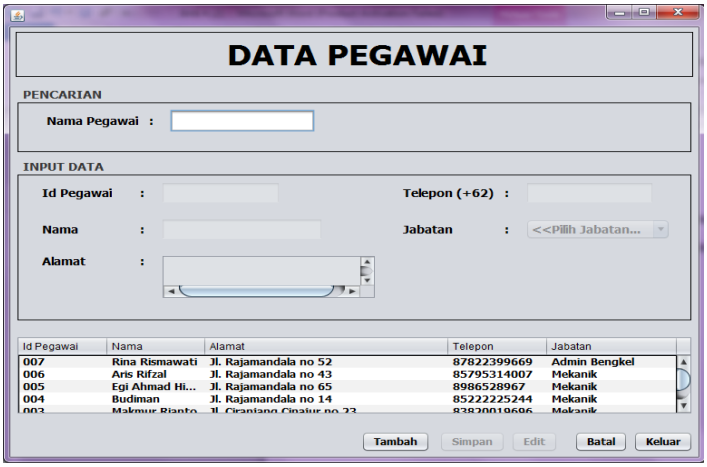
Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data Servis.



Gambar 4. 32 Rancangan *Master* data servis

5. Rancangan *Master* Data Pegawai

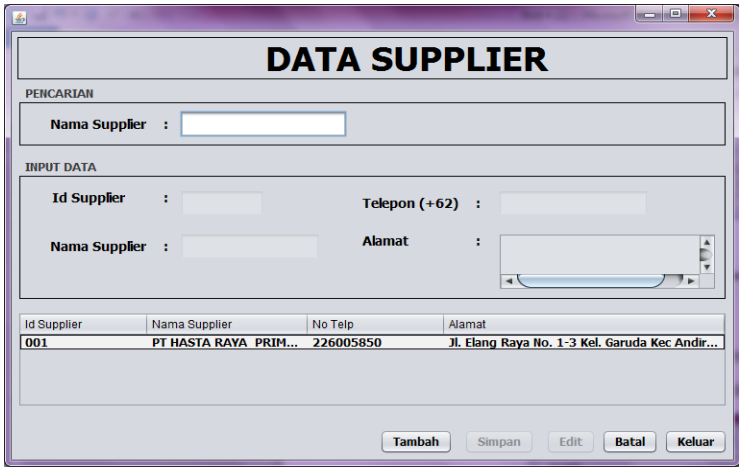
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data pegawai.



Gambar 4. 33 Rancangan Master data Pegawai

6. Rancangan *Master* Data *Supplier*

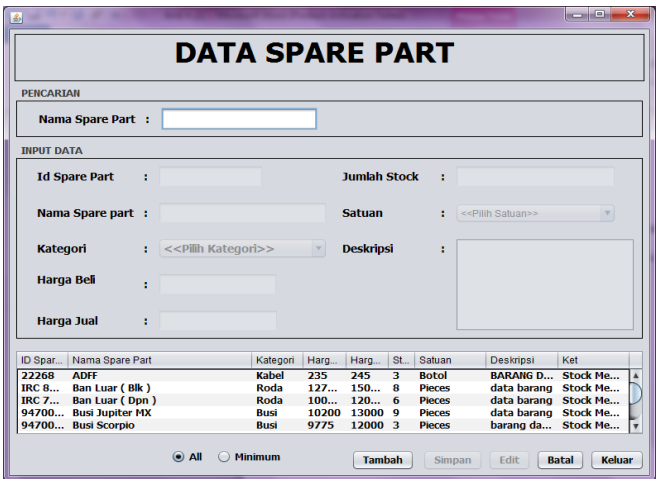
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data *supplier*.



Gambar 4. 34 Rancangan Master data *supplier*

7. Rancangan *Master* Data *Spare Part*

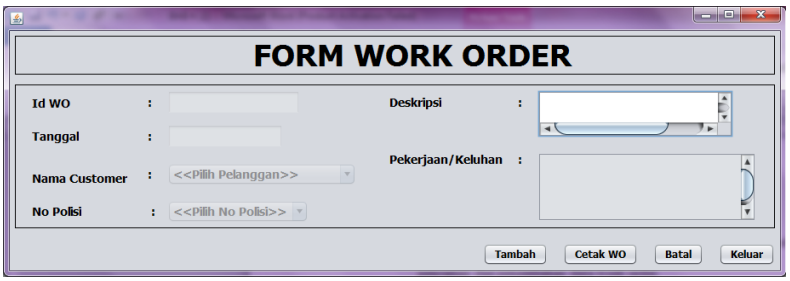
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian, pengubahan, dan penambahan data *Spare Part*



Gambar 4. 35 Rancangan *Master* data servis

8. Rancangan *Work Order*

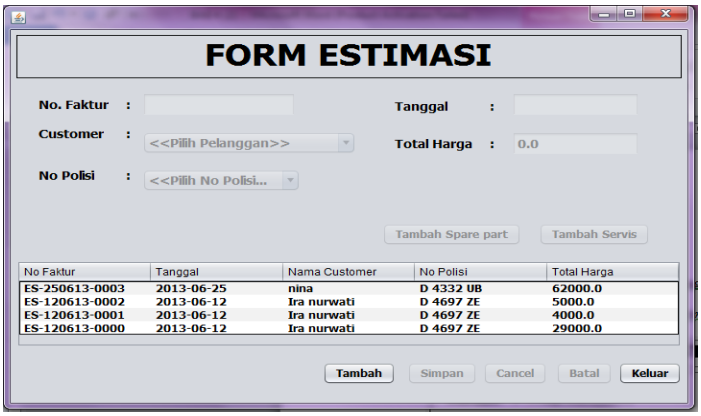
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan penambahan data work order.



Gambar 4. 36 Rancangan *Work Order*

9. Rancangan *Form Estimasi*

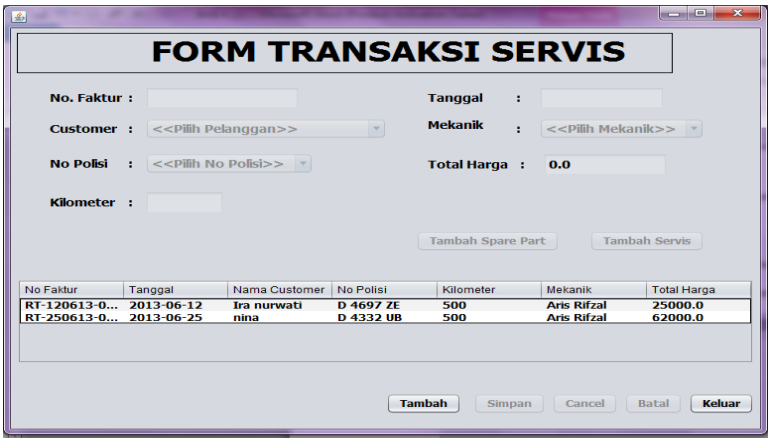
   Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan penambahan data *estimasi*.



Gambar 4. 37 Rancangan *Form Estimasi*

10. Rancangan *Form* Transaksi servis

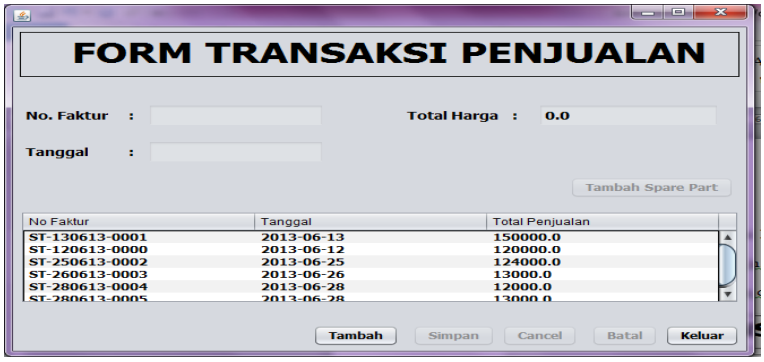
    Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan penambahan data transaksi servis.



Gambar 4. 38 Rancangan *Form* Transaksi servis

11. Rancangan *Form* Transaksi Penjualan

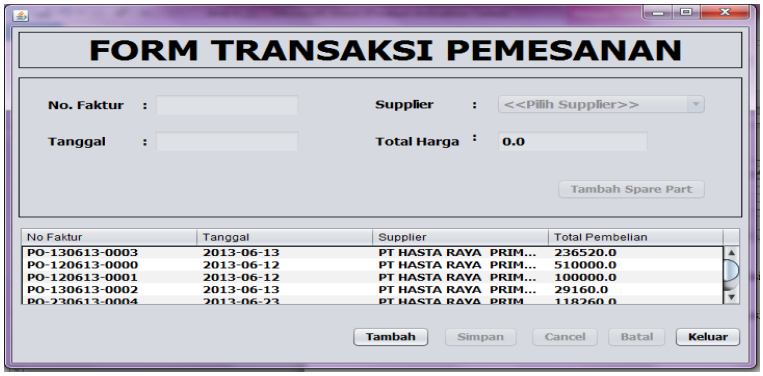
    Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan penambahan data transaksi penjualan.



Gambar 4. 39 Rancangan *Form* Transaksi Penjualan

12. Rancangan Form Transaksi Pemesanan

    Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan penambahan data transaksi pemesanan.



Gambar 4. 40 Rancangan Form Transaksi Pemesanan

13. Rancangan Form Konfirmasi PemesananTampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pencarian dan penambahan data konfirmasi pemesanan.



Gambar 4. 41 Rancangan Form konfirmasi pemesanan

14. Rancangan Form Pelaporan

Tampilan yang dirancang ini berfungsi sebagai tampilan untuk melakukan pembuatan laporan.

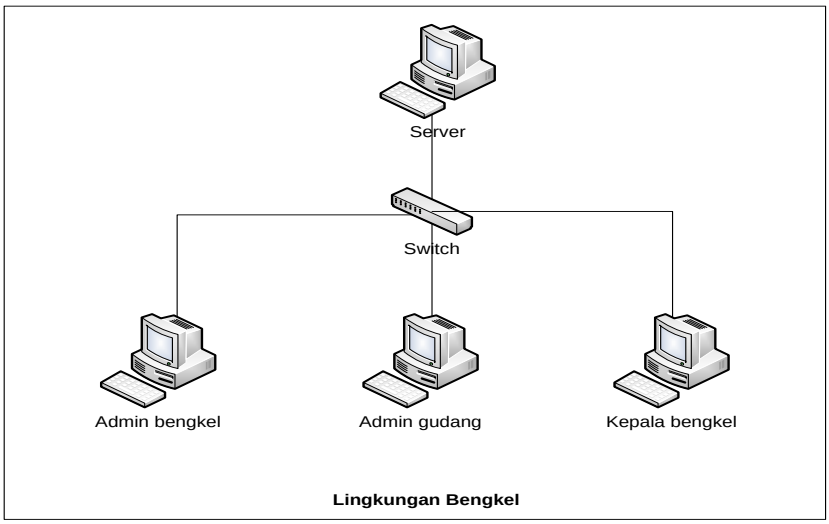


Gambar 4. 42 Rancangan Form pelaporan

4.3.4 Perancangan Arsitektur Jaringan

Arsitektur jaringan yang digunakan adalah sistem *client/server* mempunyai dua komponen utama yaitu komputer *server* dan komputer *client*. *Server* merupakan komputer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari komputer client dan bertindak sebagai

server database yang menyimpan data. *Client* yaitu komputer yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya.



Gambar 4. 43 Arsitektur jaringan yang diusulkan

Gambar 4.43 menunjukkan arsitektur jaringan. Disini penulis menggunakan topologi star dengan alasan jika salah satu komputer *client* rusak, komputer yang lain masih dapat diakses dan kontrol menagement lebih mudah karena semuanya terpusat ke server.