BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Setelah Menganalisa dan mengevaluasi sistem yang sedang berjalan, maka sebagai tindak lanjut untuk penyelesaian masalah tersebut dapat di buat suatu Sistem Informasi Penjualan *sparepart* dan *Service* Motor di Bengkel Djaya Motor dengan menggunakan sistem yang lebih baik dengan menggunakan perangkat lunak, sehingga dapat membantu untuk mendapatkan informasi yang lebih cepat dan akurat.

4.1.1 Tujuan perancangan sistem

Proses perancangan diperlukan untuk menghasilkan suatu rancangan sistem yang baik, karena dengan adanya rancangan yang tepat maka akan menghasilkan sebuah sistem yang akan memudahkan proses pengembangan sistem dimasa yang akan datang. Dalam perancangan sistem penulis mempunyai beberapa tujuan utama yaitu sebagai berikut :

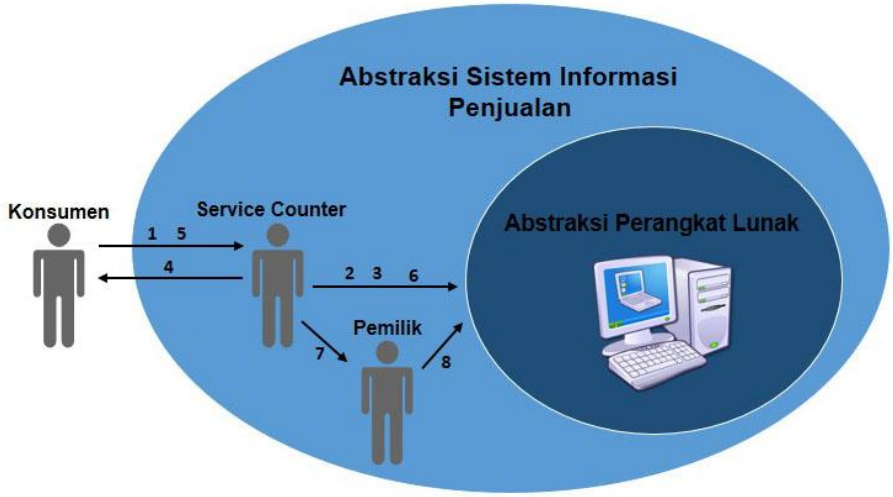
1. Membuat sistem informasi penjualan *sparepart* dan *service* motor dengan harapan dapat mempermudah operasional bengkel.
2. Mempermudah kerja bagian yang bersangkutan dalam pengolahan data.
3. Memberikan informasi yang cepat dan akurat kepada yang membutuhkan.

4.1.2 Gambaran umum sistem yang diusulkan

Dalam tahapan ini gambaran sistem yang diusulkan bertujuan untuk menghasilkan sistem informasi penjualan *Sparepart* dan *service* motor yang terkomputerisasi di bengkel Djaya Motor. Perancangan sistem yang diusulkan adalah merubah sistem yang masih manual atau masih dikerjakan dengan cara menulis ulang dari nota penjualan ke dalam buku besar laporan penjualan menjadi sistem yang terkomputerisasi. Menjadikan pengelolaan data penjualan *sparepart* lebih cepat, tepat dan akurat dengan adanya sistem yang terkomputerisasi.

4.1.3 Perancangan Prosedur yang Diusulkan

Berikut ini merupakan prosedur penjualan *sparepart* yang diusulkan pada bengkel Djaya Motor :

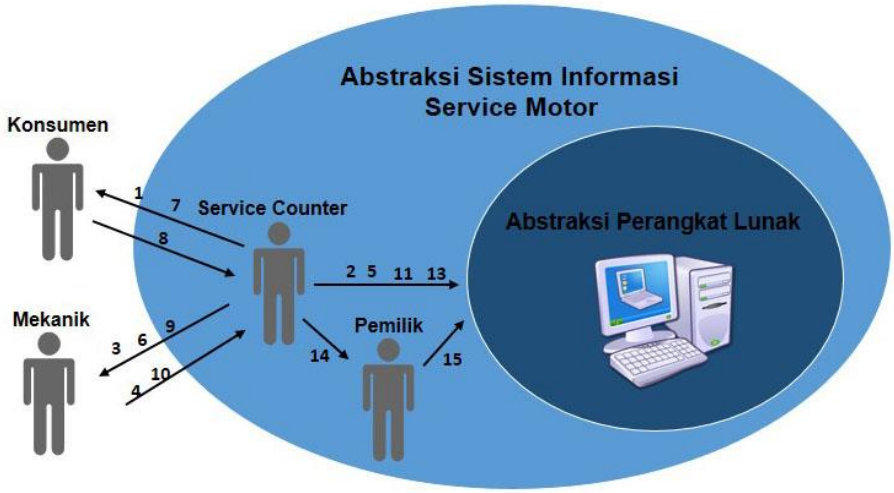


Gambar 4.1 Abstraksi Sistem Informasi Penjualan yang diusulkan

1. Konsumen menyampaikan *sparepart* yang dibutuhkan kepada bagian penjualan.
2. Bagian penjualan akan mengecek data *sparepart* yang dibutuhkan konsumen di *database*.

3. Jika *sparepart* ada, maka bagian penjualan akan menginput data penjualan dan mencetak faktur penjualan.
4. Bagian penjualan memberikan *sparepart* beserta faktur penjualan ke konsumen.
5. Konsumen akan membayar sesuai dengan jumlah yang tetera di faktur penjualan.
6. Bagian penjualan akan membuat laporan penjualan *sparepart* secara berkala.
7. Bagian penjualan memberikan laporan penjualan sparepart kepada pemilik bengkel.
8. Pemilik mengakses laporan penjualan *sparepart*.

Berikut ini merupakan prosedur *service* motor yang diusulkan pada bengkel Djaya Motor :



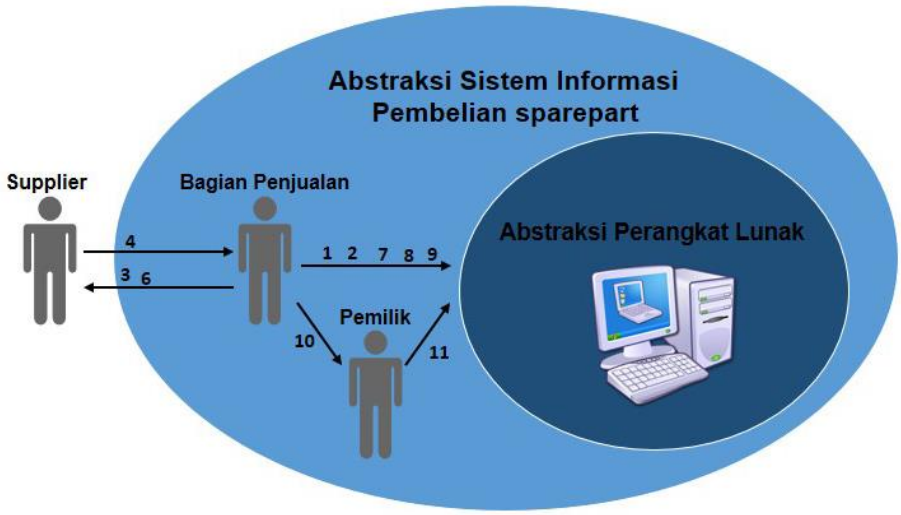
Gambar 4.2 Abstraksi Sistem Informasi *Service* Motor yang diusulkan

1. Bagian penjualan akan menanyakan data motor, data diri konsumen dan keluhan motor.

2. Bagian penjualan memasukan data tersebut ke *database* dan mencetak surat perintah kerja (SPK).
3. Bagian penjualan menyampaikan SPK ke mekanik untuk dilakukan pengecekan dan perbaikan.
4. Mekanik melakukan pengecekan dan perbaikan motor sesuai dengan keluhan yang tertera di SPK, jika memerlukan pergantian *sparepart* mekanik menuliskan data *sparepart* pada SPK dan memberikan ke bagian penjualan.
5. Bagian penjualan akan melakukan pencarian data *sparepart* di *database*.
6. Jika kosong SPK akan dikembalikan ke mekanik.
7. Jika ada, bagian penjualan akan memberi tahu ke konsumen untuk menyetujui pergantian *sparepart*.
8. Konsumen menyetujui pergantian *sparepart* kepada bagian penjualan.
9. Bagian penjualan memberikan *sparepart* yang dibutuhkan ke mekanik.
10. Mekanik melakukan pergantian *sparepart*, jika telah selesai mekanik akan memvalidasi SPK tanda bahwa telah selesai dan menyerahkannya ke bagian penjualan.
11. Bagian penjualan akan membuatkan faktur pembayaran dan diserahkan ke konsumen.
12. Konumen akan melakukan pembayaran sesuai dengan jumlah yang tertera di faktur.
13. Bagian penjualan akan membuat laporan *service* motor secara berkala.
14. Laporan diserahkan ke pemilik.

15. Pemilik mengakses laporan *service* motor.

Berikut ini merupakan prosedur pembelian *sparepart* yang diusulkan pada bengkel Djaya Motor :



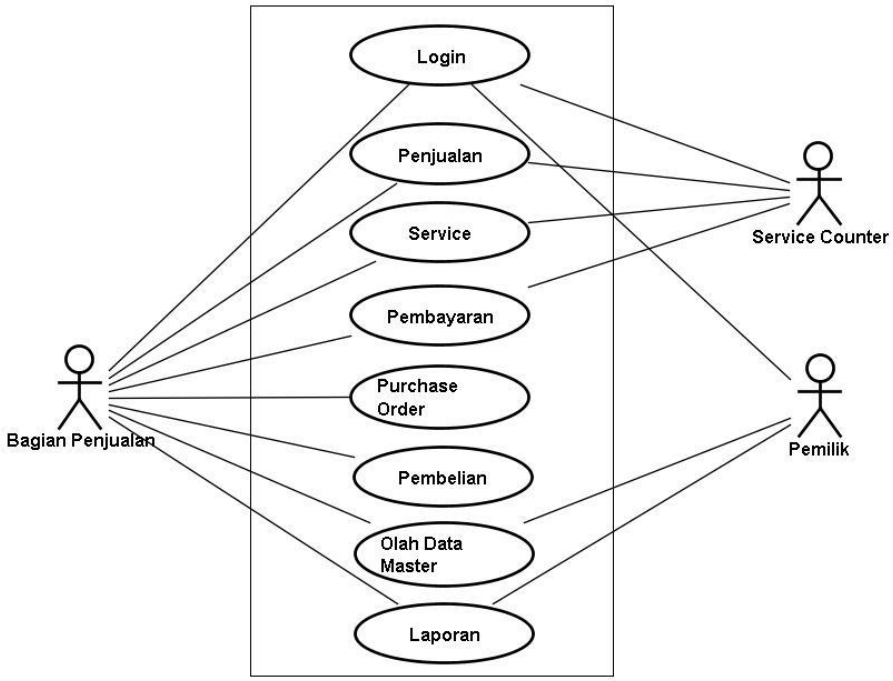
Gambar 4.3 Abstraksi Sistem Informasi Pembelian *Sparepart* yang diusulkan

1. Bagian penjualan akan mengecek ketersediaan *sparepart*, jika *sparepart* kosong atau kurang dari 6 bagian penjualan akan memasukan data *sparepart* ke PO,
2. Jika sudah memasukan data *sparepart* yang di butuhkan maka bagian penjualan akan mencetak PO,
3. Lalu PO di berikan kepada supplier untuk memenuhi *sparepart* yang diminta,
4. Setelah *supplier* menerima PO, supplier akan memenuhi *order* dan mengirimkan *sparepart* tersebut ke bagian penjualan beserta faktur pembelian,
5. Bagian penjualan akan mengecek data PO dengan barang yang dikirim *supplier*,

6. Jika tidak sesuai *sparepart* beserta faktur akan dikembalikan lagi ke *supplier*, dan meminta memenuhi sesuai dengan po,
7. Jika sesuai, bagian penjualan akan masuk ke pembelian melakukan *input* transaksi pembelian,
8. Bagian penjualan akan menginput harga beli sparepart dan akan langsung muncul harga jual yaitu 15% dari harga beli tersebut,
9. Bagian penjualan akan mencetak laporan pembelian *sparepart*,
10. Bagian penjualan memberikan laporan pembelian *sparepart* kepada pemilik
11. Pemilik mengakses laporan pembelian *sparepart* dan stok *sparepart*.

4.1.3.1 *Use case diagram*

Berikut ini adalah diagram *Use Case* yang menggambarkan proses utama dari sistem yang diusulkan :



Gambar 4.4 *Use Case Diagram* yang diusulkan

4.1.3.2 Skenario *use case* yang diusulkan

Skenario *Use Case* bertujuan untuk mendeskripsikan *Diagram Use Case*. Berikut *Skenario Use Case* yang sedang diusulkan di bengkel Djaya Motor.

Nama *use case* : *Login*

Aktor : Bagian Penjualan, *Service* Counter, Pemilik

Tujuan : Keamanan untuk masuk ke dalam sistem

Tabel 4.1 Skenario *Use Case Login*

Bagian Penjualan, Pemilik	**Sistem**
1. *Input username* dan *password*	
	2. Baca inputan *username* dan *password*.
	3. Memeriksa inputan dalam *database*.
	4. Menampilkan pesan *login* berhasil.
	5. Menampilkan Menu Utama.
Kondisi akhir	
Sistem siap digunakan.	

Nama *use case* : Penjualan

Aktor : Bagian Penjualan, *Service Counter*

Tujuan : Memudahkan dalam transaksi penjualan

Tabel 4.2 Skenario *Use Case* Penjualan

Bagian Penjualan, *Service Counter*	**Sistem**
1. Memilih menu Penjualan.	
	2. Menampilkan menu Penjualan.
3. Pilih tombol Tambah.	
	4. Menu Penjualan aktif.

Bagian Penjualan, *Service Counter*	**Sistem**
5. Pilih tombol Cari Data *sparepart*.	
	6. Menampilkan menu dialog pilih *sparepart*.
7. Pilih data *sparepart* dan pilih tombol Pilih.	
	8. Menampilkan data *sparepart* pada *form* penjualan.
9. Input jumlah *sparepart* dan pilih tombol Tambah.	
	10. Menghitung sub total, total harga, total item, total diskon, total bayar dan menampilkan data penjualan pada tabel penjualan.
11. Input jumlah bayar dan tekan enter.	
	12. Menghitung jumlah kembalian.
13. Pilih tombol OK.	
	14. Menampilkan pesan konfirmasi "Apakah data yang akan disimpan sudah benar?".
15. Pilih tombol "Yes".	
	16. Menyimpan data penjualan ke database, menampilkan pesan "Input Penjualan Sukses", mencetak faktur penjualan dan form penjualan tidak aktif.
Kondisi akhir	
Transaksi penjualan tersimpan di database, dan dapat digunakan di laporan penjualan, dan sistem telah mencetak faktur penjualan.	

Nama *use case* : *Service*

Aktor : Bagian Penjualan, *Service Counter*

Tujuan : Memudahkan dalam transaksi *service*

Tabel 4.3 Skenario *Use Case Service*

Bagian Penjualan, *Service Counter*	**Sistem**
1. Memilih menu *Service*.	
	2. Menampilkan menu *Service*.
3. Pilih tombol Tambah.	
	4. Menu *Service* aktif.
5. Pilih tombol cari data konsumen.	
	6. Menampilkan menu dialog pilih data konsumen.
7. Pilih data konsumen dan pilih tombol Pilih.	
	8. Menampilkan data konsumen di *form service*.
9. Input data *service* dan pilih tombol tambah *service*.	
	10. Menghitung total *service*, menghitung total *service* dan *sparepart* dan menampilkan data *service* pada tabel *service*.
11. Centang *checkbox sparepart*.	
	12. *Form sparepart* aktif.
13. Pilih tombol cari *sparepart*.	
	14. Menampilkan menu dialog pilih *sparepart*.
15. Pilih data *sparepart* dan pilih tombol Pilih.	

Bagian Penjualan, *Service Counter*	**Sistem**
	16. Menampilkan data *sparepart* pada *form service*.
17. Input jumlah *sparepart* dan pilih tombol tambah *sparepart*.	
	18. Menghitung jumlah *sparepart*, mengitung total harga *sparepart*, menghitung total *service* dan *sparepart* dan menampilkan data penjualan *sparepart* pada tabel penjualan.
19. Pilih tombol Simpan.	
	20. Menampilkan pesan konfirmasi "Apakah data yang akan disimpan sudah benar?".
21. Pilih tombol OK.	
	22. Menyimpan data *service* ke *database*.
	23. Menampilkan pesan "*Input Service* Sukses".
	24. Mencetak Surat Perintah Kerja (SPK).
Kondisi akhir	
Transaksi *service* tersimpan di database, dan dapat digunakan di laporan *service*, mekanik menerima surat perintah kerja.	

Nama *use case* : Pembayaran

Aktor : Bagian Penjualan, *Service Counter*

Tujuan : Memudahkan dalam transaksi pembayaran

Tabel 4.4 Skenario *Use Case* Pembayaran

Bagian Penjualan, *Service Counter*	**Sistem**
1. Memilih menu Pembayaran.	
	2. Menampilkan menu Pembayaran.
3. Pilih tombol Tambah.	
	4. Menu Pembayaran aktif.
5. Pilih tombol Cari.	
	6. Menampilkan menu dialog cari data *service*.
7. Pilih data *service* dan pilih tombol Pilih.	
	8. Menampilkan data *service* di form pembayaran.
9. Input jumlah bayar dan *Enter*.	
	10. Menghitung jumlah kembalian tombol Bayar aktif.
11. Pilih tombol bayar.	
	12. Menampilkan pesan konfirmasi “Apakah data yang akan disimpan sudah benar?”.
13. Pilih tombol OK.	
	14. Menyimpan inputan ke database.
	15. Menampilkan pesan “Input Pembayaran Sukses”.
	16. Mencetak faktur pembayaran.
Kondisi akhir	

Transaksi pembayaran tersimpan di database, dan dapat digunakan di laporan *service*, dan sistem telah mencetak bukti transaksi.

Nama *use case* : *Purchase Order*

Aktor : Bagian Penjualan

Tujuan : Memudahkan dalam transaksi pemesanan

Tabel 4.5 Skenario *Use Case Purchase Order*

Bagian Penjualan	Sistem
1. Memilih menu *Purchase Order*.	
	2. Menampilkan menu *Purchase Order*.
3. Pilih tombol Tambah.	
	4. Menu *purchase order* aktif.
5. Input tanggal kirim.	
6. Pilih tombol Cari *Supplier*.	
	7. Menampilkan menu dialog pilih data *supplier*.
8. Pilih data *supplier* dan pilih tombol Pilih.	
	9. Menampilkan data *supplier* pada *form purchase order*.
10. Pilih tombol Cari *Sparepart*.	
	11. Menampilkan menu dialog pilih stok *sparepart* hampir habis.
12. Pilih data *sparepart* dan pilih tombol Pilih.	
	13. Menampikan data *sparepart* pada *form purchase order*.

Bagian Penjualan	Sistem
14. Input jumlah *sparepart* dan pilih tombol tambah *sparepart*.	
	15. Menghitung jumlah sparepart dan menampilkan data *purchase order* pada tabel *purchase order*.
16. Pilih tombol simpan.	
	17. Menampilkan pesan konfirmasi "Apakah data yang akan disimpan sudah benar?".
18. Pilih tombol OK.	
	19. Menyimpan data *Purchase Order* ke *database*.
	20. Menampilkan pesan "Input *Purchase Order* Sukses".
	21. Mencetak surat *purchase order*.
Kondisi akhir	
Transaksi *purchase order* tersimpan di database, dan dapat digunakan di laporan *pemesanan*, dan sistem telah mencetak *purchase order*.	

Nama *use case* : Pembelian

Aktor : Bagian Penjualan

Tujuan : Memudahkan dalam transaksi pembelian

Tabel 4.6 Skenario *Use Case* Pembelian

Bagian Penjualan	Sistem
1. Memilih menu Pembelian.	
	2. Menampilkan menu Pembelian.
3. Pilih tombol tambah.	
	4. Form pembelian aktif.

Bagian Penjualan	Sistem
5. Pilih tombol Cari data *Purchase Order*.	
	6. Menampilkan dialog menu *Purchase Order*.
7. Pilih data *Purchase Order* dan pilih tombol Pilih.	
	8. Menampilkan data *Purchase Order* di *form* Pembelian dan tabel pembelian.
9. Pilih data *sparepart* pada tabel pembelian dan input harga beli lalu tekan *Enter*.	
	10. Menghitung harga jual 15% dari harga beli.
11. Pilih tombol Simpan Edit Harga.	
	12. Menjumlah total beli dan total pembelian.
13. Pilih tombol simpan.	
	14. Menampilkan pesan konfirmasi "Apakah data yang akan disimpan sudah benar?".
15. Pilih tombol OK.	
	16. Menyimpan data pembelian di *database*.
	17. Menampilkan pesan "Input Pembelian Sukses".
Kondisi akhir	
Transaksi pembelian tersimpan di database, dan dapat digunakan di laporan pembelian *service*, meng *update* data harga *sparepart* dan data *purchase order*.	

Nama *use case* : Olah Data Master

Aktor : Bagian Penjualan, Pemilik

Tujuan : Untuk menambah, merubah dan menghaspu data

Tabel 4.7 Skenario *Use Case* Olah Data Master

Bagian Penjualan, Pemilik	Sistem
1. Memilih menu *Data Sparepart*.	
	2. Menampilkan menu Data Sparepart.
3. Tambah, *Edit*, Hapus *data sparepart*.	
	4. Meng *update* dan menyimpan *data sparepart* ke *database*.
Kondisi akhir	
Data terupdate dan tersimpan di *database*.	

Nama *use case* : Laporan

Aktor : Bagian Penjualan, Pemilik

Tujuan : Memudahkan dalam transaksi pembelian

Tabel 4.8 Skenario *Use Case* Laporan

Bagian Penjualan, Pemilik	Sistem
1. Memilih menu Laporan.	
	2. Menampilkan menu Laporan.
3. Pilih *radio button* periode laporan.	
	4. Textfield aktif.
5. Input tanggal dan pilih *button* Tampil.	

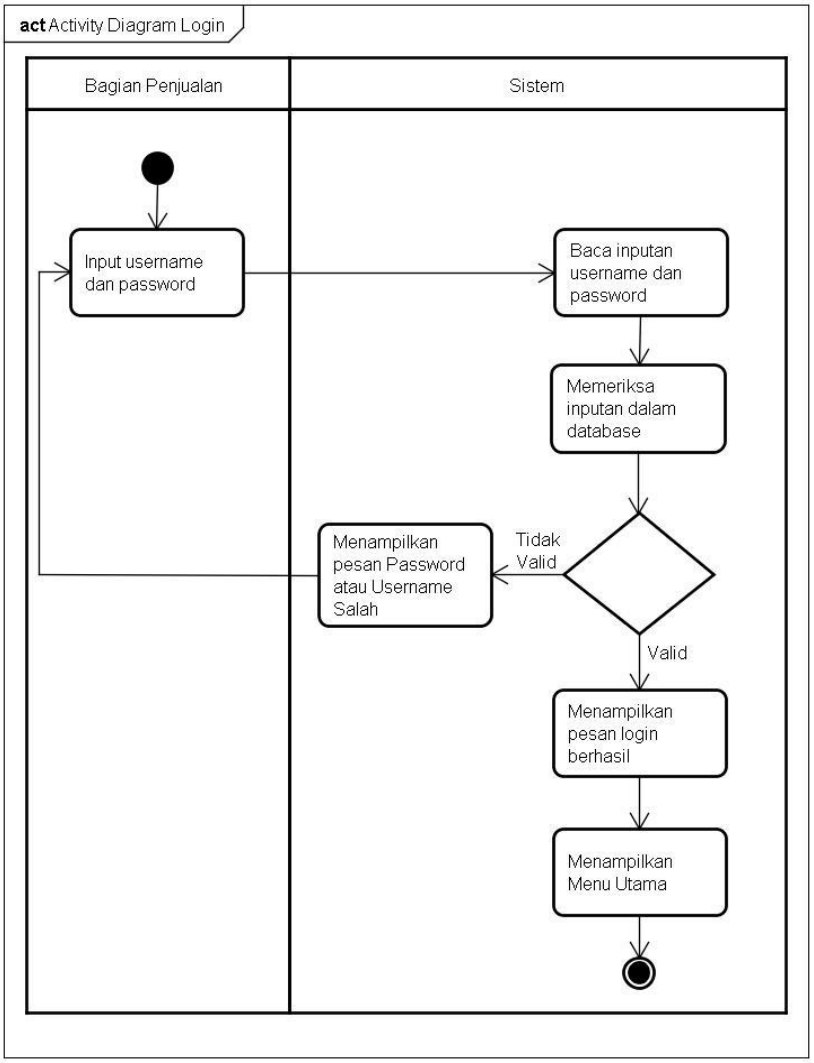
Bagian Penjualan, Pemilik	Sistem
	6. Menampilkan laporan.
Kondisi akhir	
Transaksi pembelian tersimpan di database, dan dapat digunakan di laporan pembelian *service*, meng *update* data harga *sparepart* dan data *purchase order*.	

4.1.3.3 *Activity diagram*

activity Diagram memodelkan alur kerja (*workflow*) sebuah proses bisnis dan urutan aktifitas dalam suatu proses.

a) *Activity Diagram Login*

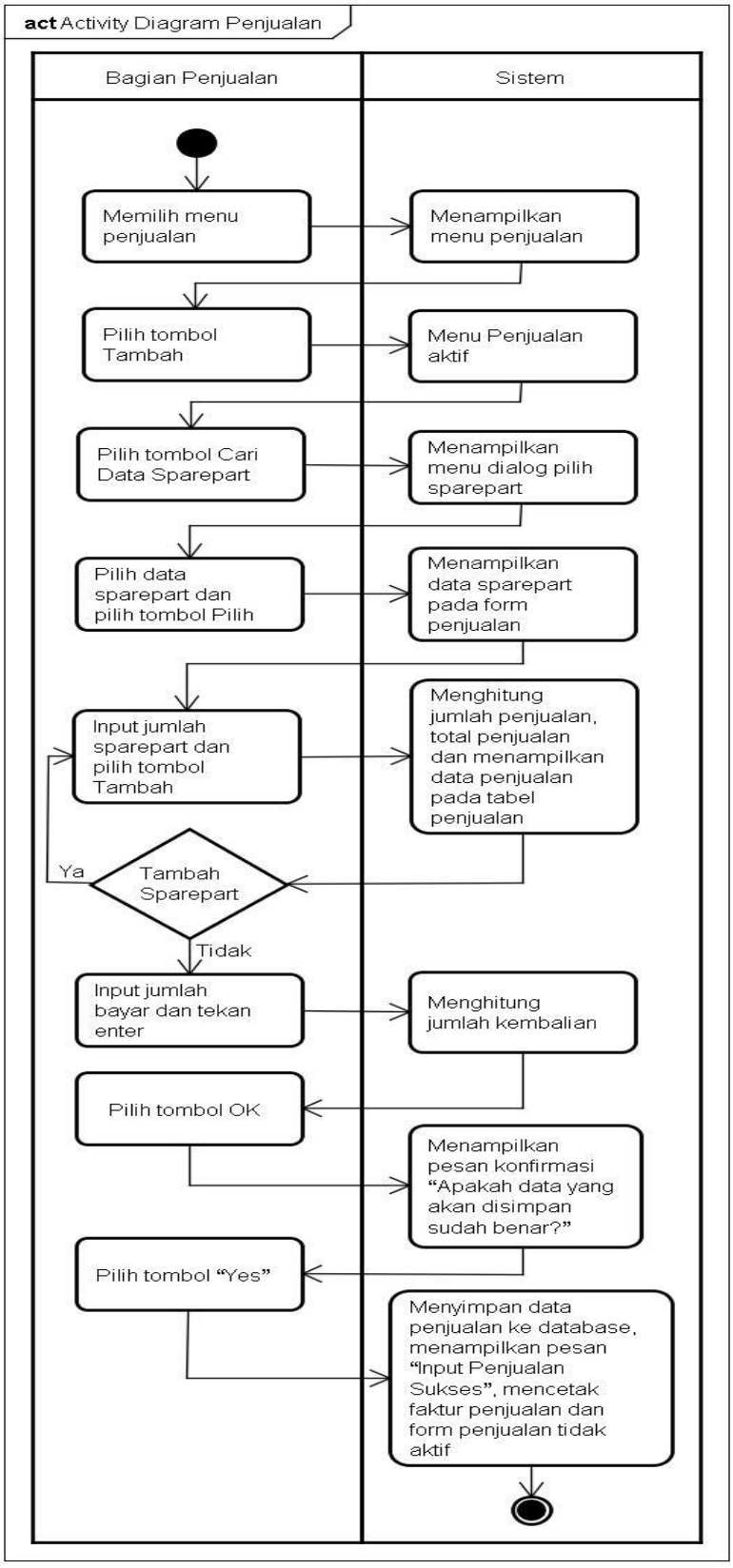
Acitivity Diagram Login memperlihatkan segala aktifitas yang terjadi pada saat *login* kedalam sistem. Aktor yang berperan dalam proses ini adalah bagian penjualan, pemilik. Adapun *Activity Diagram Login* adalah sebagai berikut :



Gambar 4.5 *Activity Diagram Login* yang diusulkan

b) *Activity Diagram* Penjualan

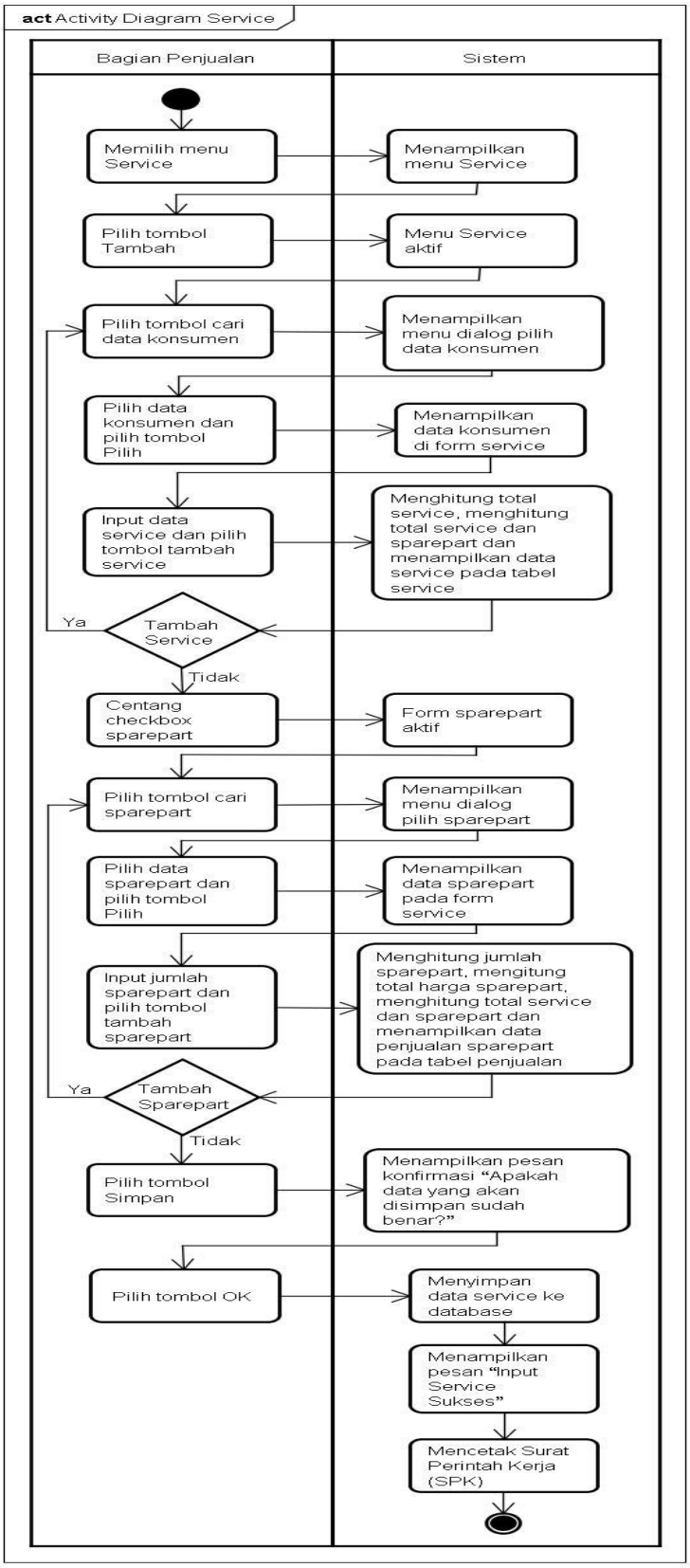
Acitivity Diagram Penjualan memperlihatkan segala aktifitas yang terjadi pada saat transaksi penjualan di dalam sistem. Aktor yang berperan dalam proses ini adalah bagian penjualan. Adapun *Activity Diagram* Penjualan adalah sebagai berikut :



Gambar 4.6 *Activity Diagram* Penjualan yang diusulkan

c) *Activity Diagram Service*

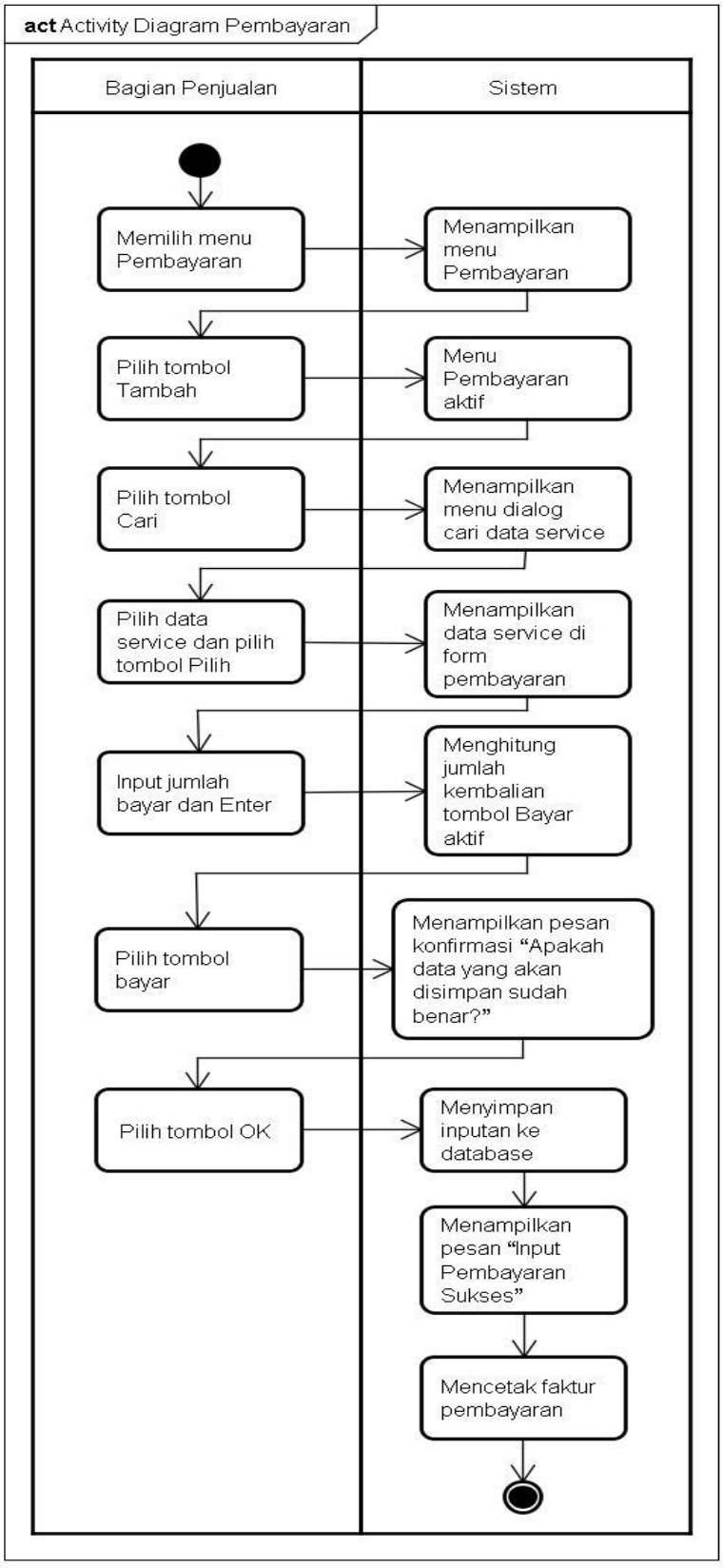
Acitivity Diagram Service memperlihatkan segala aktifitas yang terjadi pada saat transaksi *service* di dalam sistem. Aktor yang berperan dalam proses ini adalah bagian penjualan. Adapun *Activity Diagram Service* adalah sebagai berikut :



Gambar 4.7 *Activity Diagram Service* yang diusulkan

d) *Activity Diagram* Pembayaran

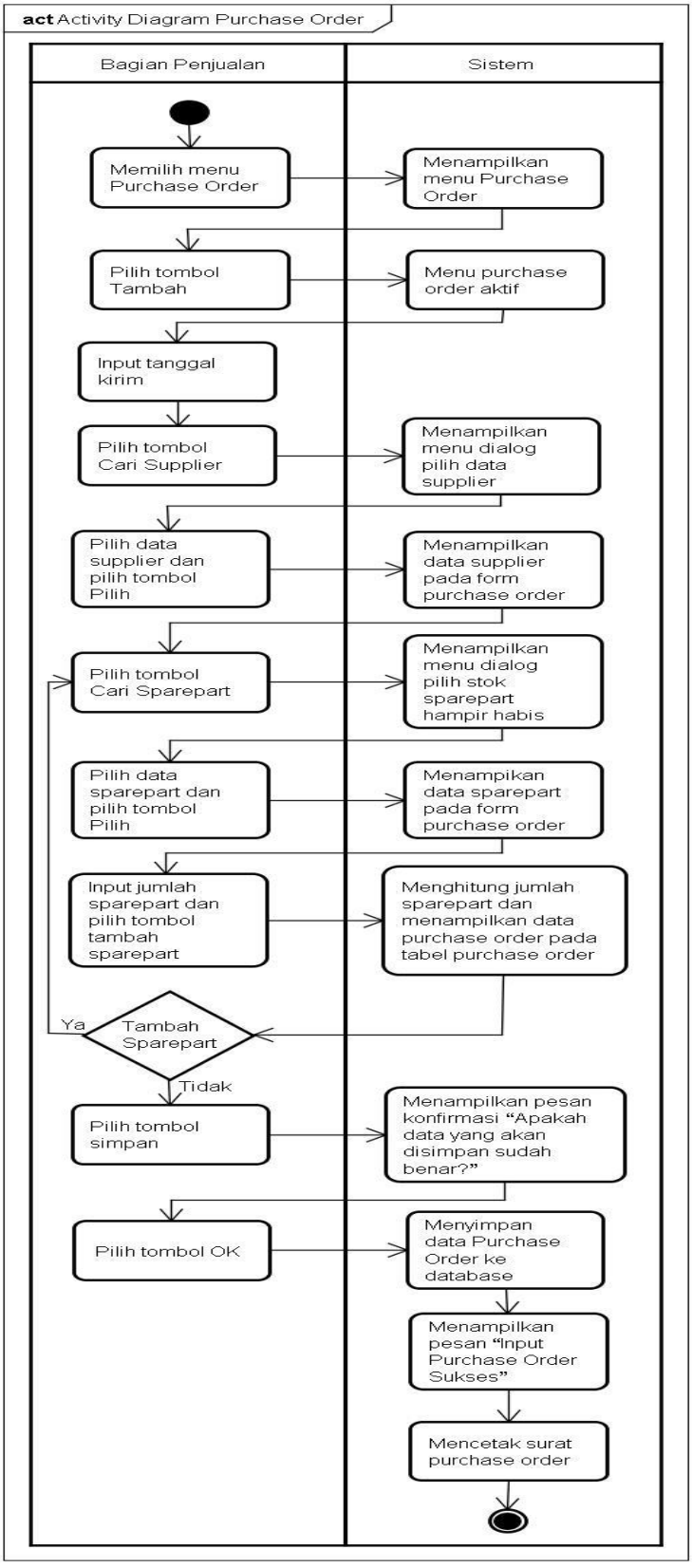
Acitivity Diagram Pembayaran memperlihatkan segala aktifitas yang terjadi pada saat transaksi pembayaran di dalam sistem. Aktor yang berperan dalam proses ini adalah bagian penjualan. Adapun *Activity Diagram* Pembayaran adalah sebagai berikut :



Gambar 4.8 *Activity Diagram* Pembayaran yang diusulkan

e) *Activity Diagram Purchase Order*

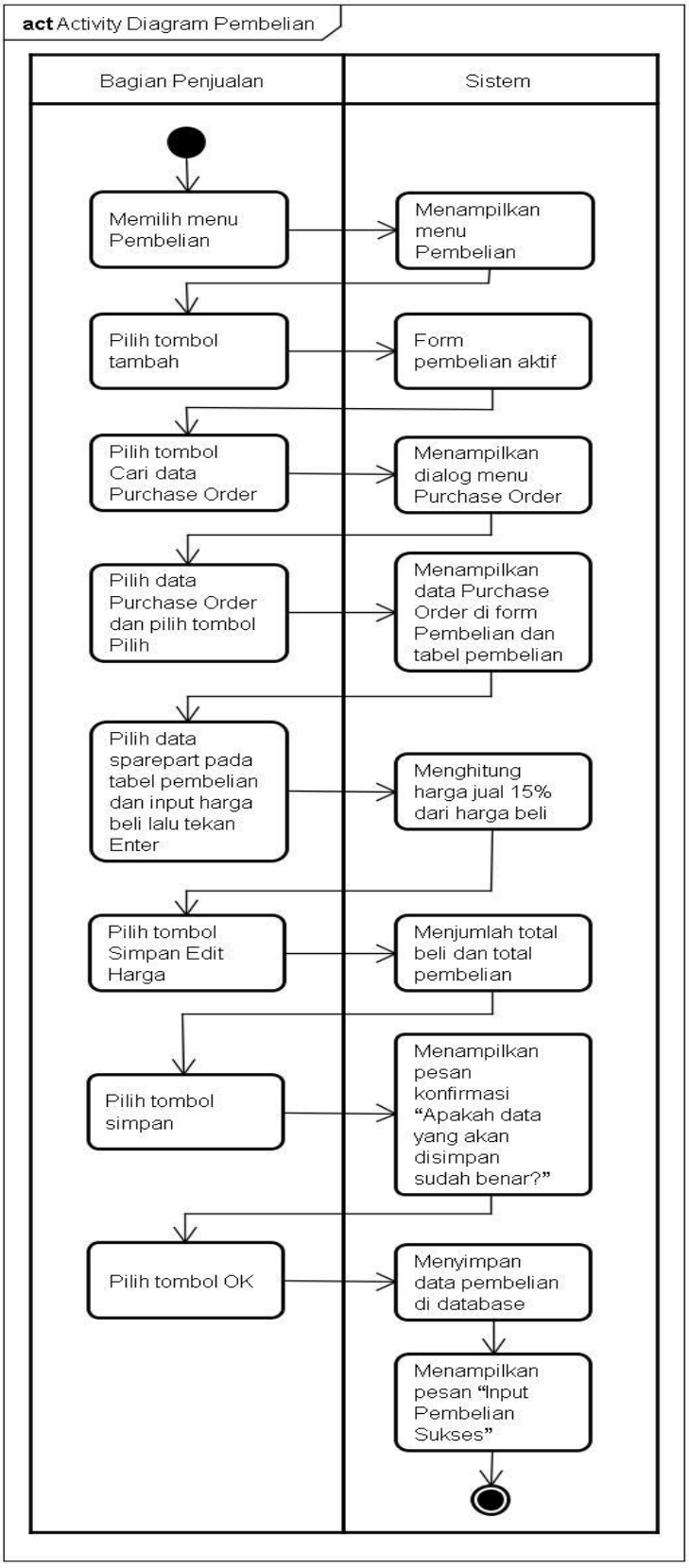
Acitivity Diagram Purchase Order memperlihatkan segala aktifitas yang terjadi pada saat transaksi *purchase order* di dalam sistem. Aktor yang berperan dalam proses ini adalah bagian penjualan. Adapun *Activity Diagram Purchase Order* adalah sebagai berikut :



Gambar 4.9 *Activity Diagram Purchase Order* yang diusulkan

f) *Activity Diagram* Pembelian

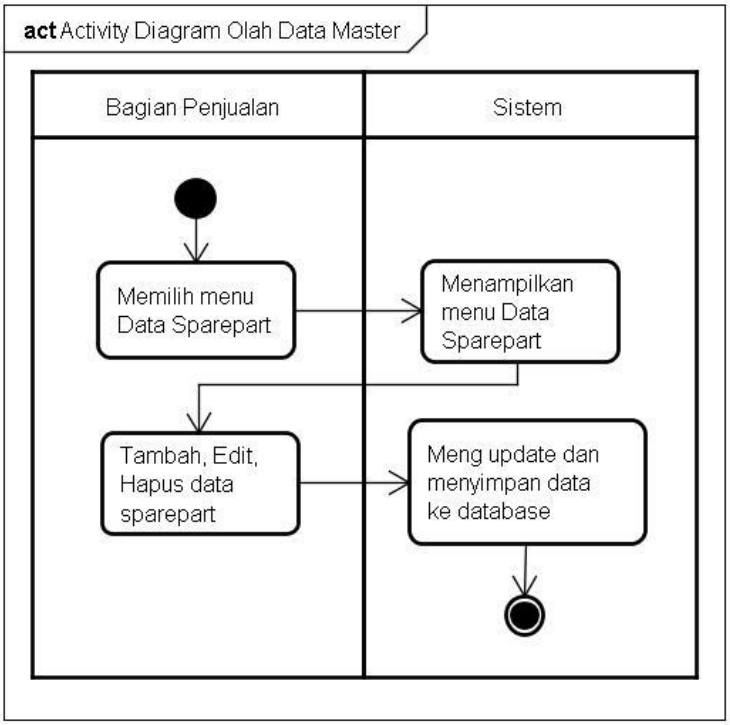
Acitivity Diagram Pembelian memperlihatkan segala aktifitas yang terjadi pada saat transaksi Pembelian di dalam sistem. Aktor yang berperan dalam proses ini adalah bagian penjualan. Adapun *Activity Diagram* Pembelian adalah sebagai berikut :



Gambar 4.10 *Activity Diagram* Pembelian yang diusulkan

g) *Activity Diagram* Olah Data Master

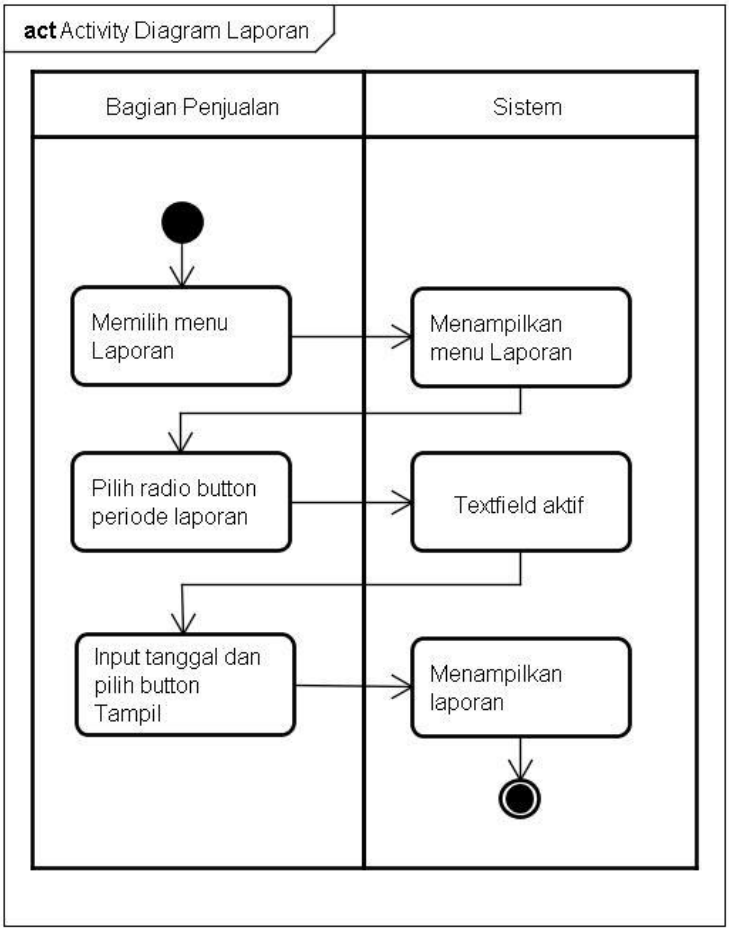
Acitivity Diagram Olah Data Master memperlihatkan segala aktifitas yang terjadi pada saat olah data master. Aktor yang berperan dalam proses ini adalah bagian penjualan dan pemilik. Adapun *Activity Diagram* olah Data Master adalah sebagai berikut :



Gambar 4.11 *Activity Diagram* Olah Data Master yang diusulkan

h) Activity Diagram Laporan.

Acitivity Diagram Laporan memperlihatkan segala aktifitas yang terjadi pada saat membuatan laporan. Aktor yang berperan dalam proses ini adalah bagian penjualan dan pemilik. Adapun *Activity Diagram* Laporan adalah sebagai berikut :



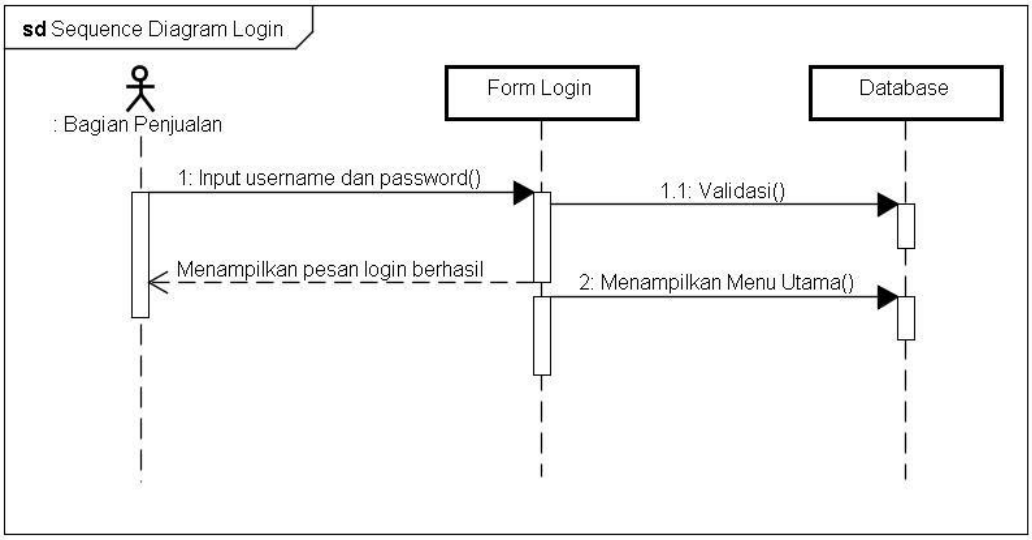
Gambar 4.12 *Activity Diagram* Laporan yang diusulkan

4.1.3.4 *Sequene diagram*

Diagram Sekuensial atau *Sequence Diagram* digunakan untuk menunjukkan aliran fungsionalitas dalam *Use Case. Sequence Diagram* digunakan untuk memberikan gambaran detail dari setiap *Use Case Diagram* yang telah dibuat sebelumnya. Setiap objek yang terlibat dalam sebuah *Use Case* digambarkan dengan garis putus-putus vertikal, kemudian *message* yang dikirim oleh objek digambarkan dengan garis horizontal secara kronologis dari atas ke bawah.

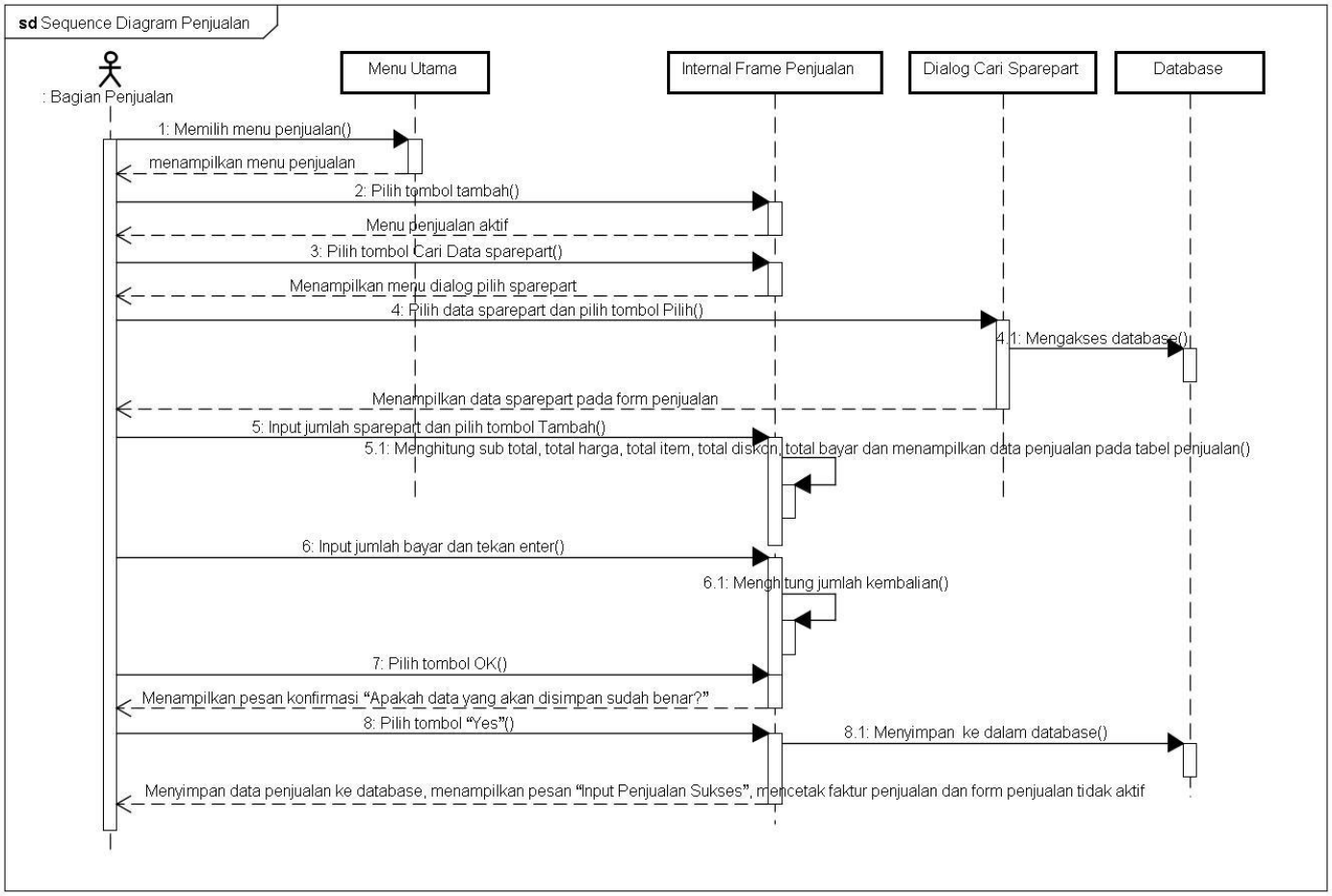
Berikut adalah Diagram Sekuensial yang menggambarkan fungsionalitas Sistem Informasi Penjualan *Sparepart* dan *Service* Motor pada bengkel Djaya Motor.

1. *Sequence Diagram Login*



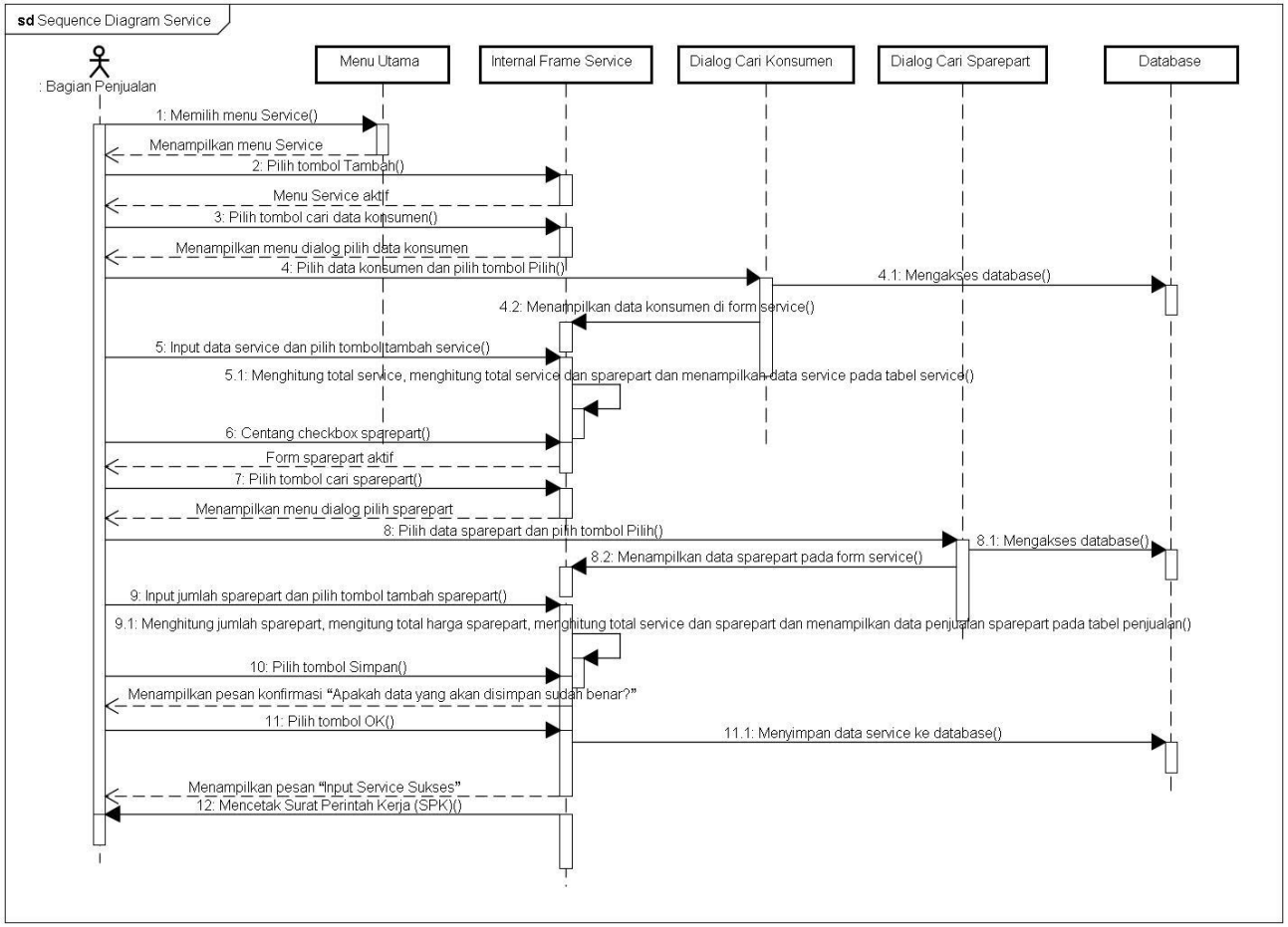
Gambar 4.13 *Sequence Diagram Login* yang diusulkan

2. *Sequence Diagram* Penjualan



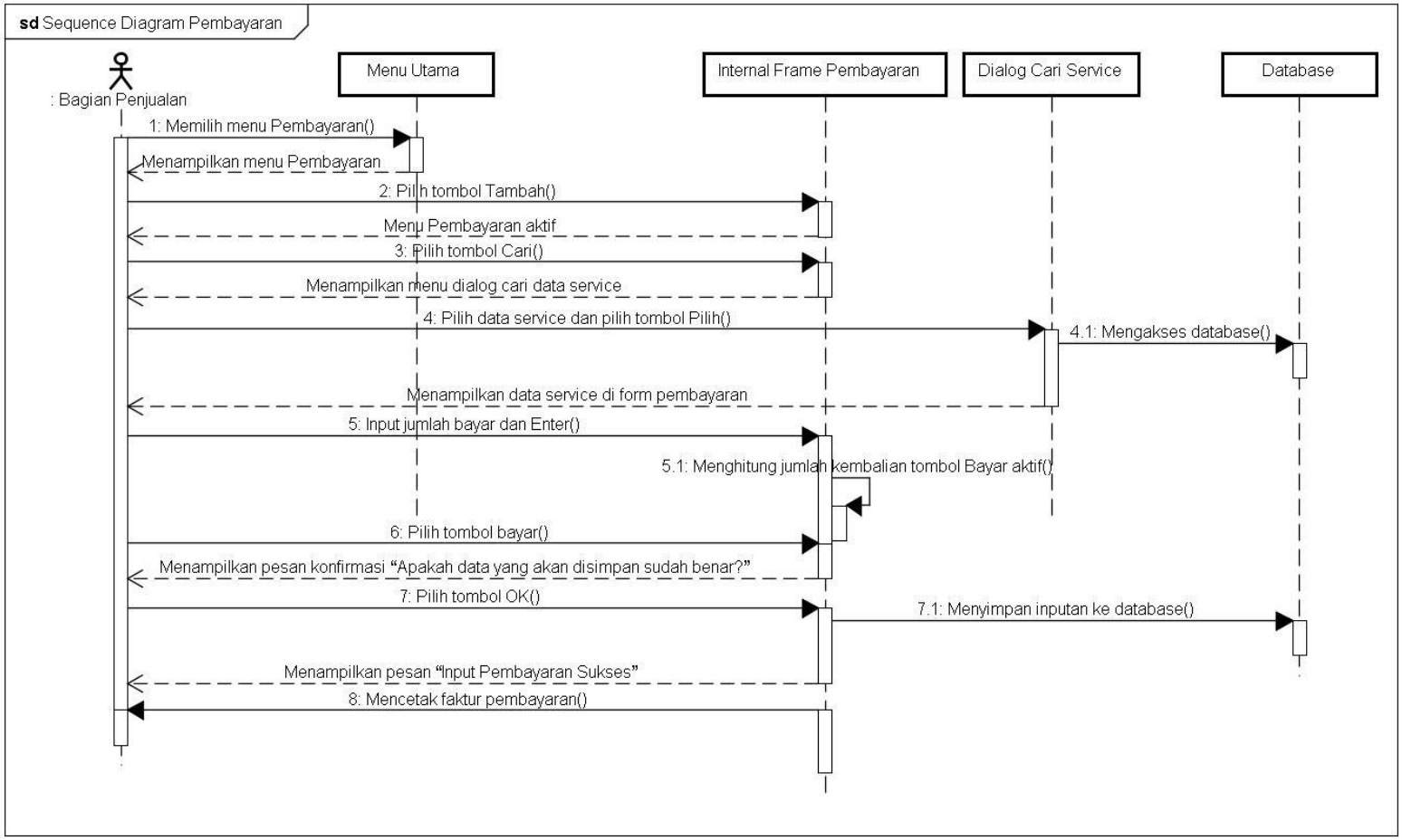
Gambar 4.14 *Sequence Diagram* Penjualan yang diusulkan

3. *Sequence Diagram Service*



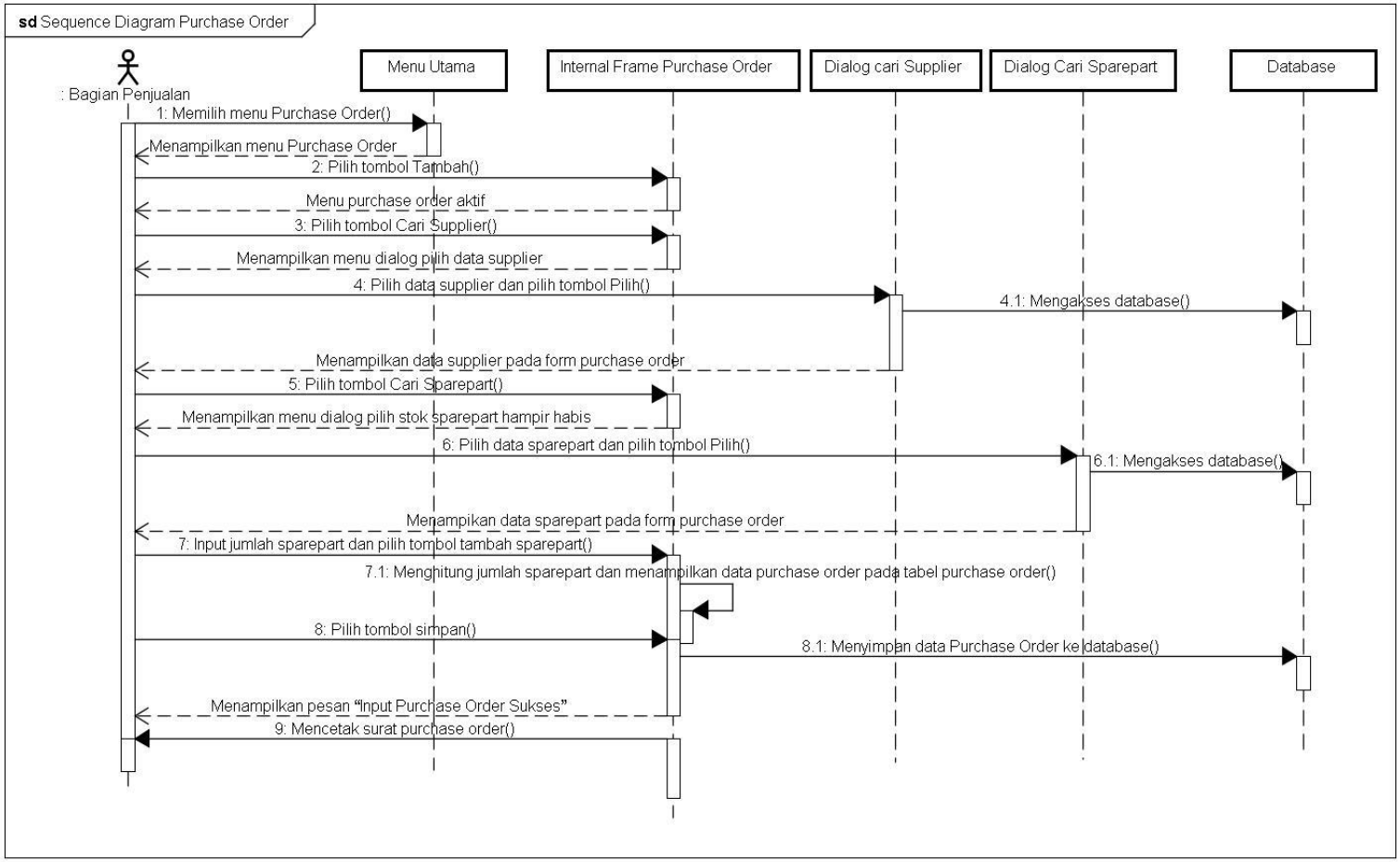
Gambar 4.15 *Sequence Diagram Service* yang diusulkan

4. *Sequence Diagram* Pembayaran



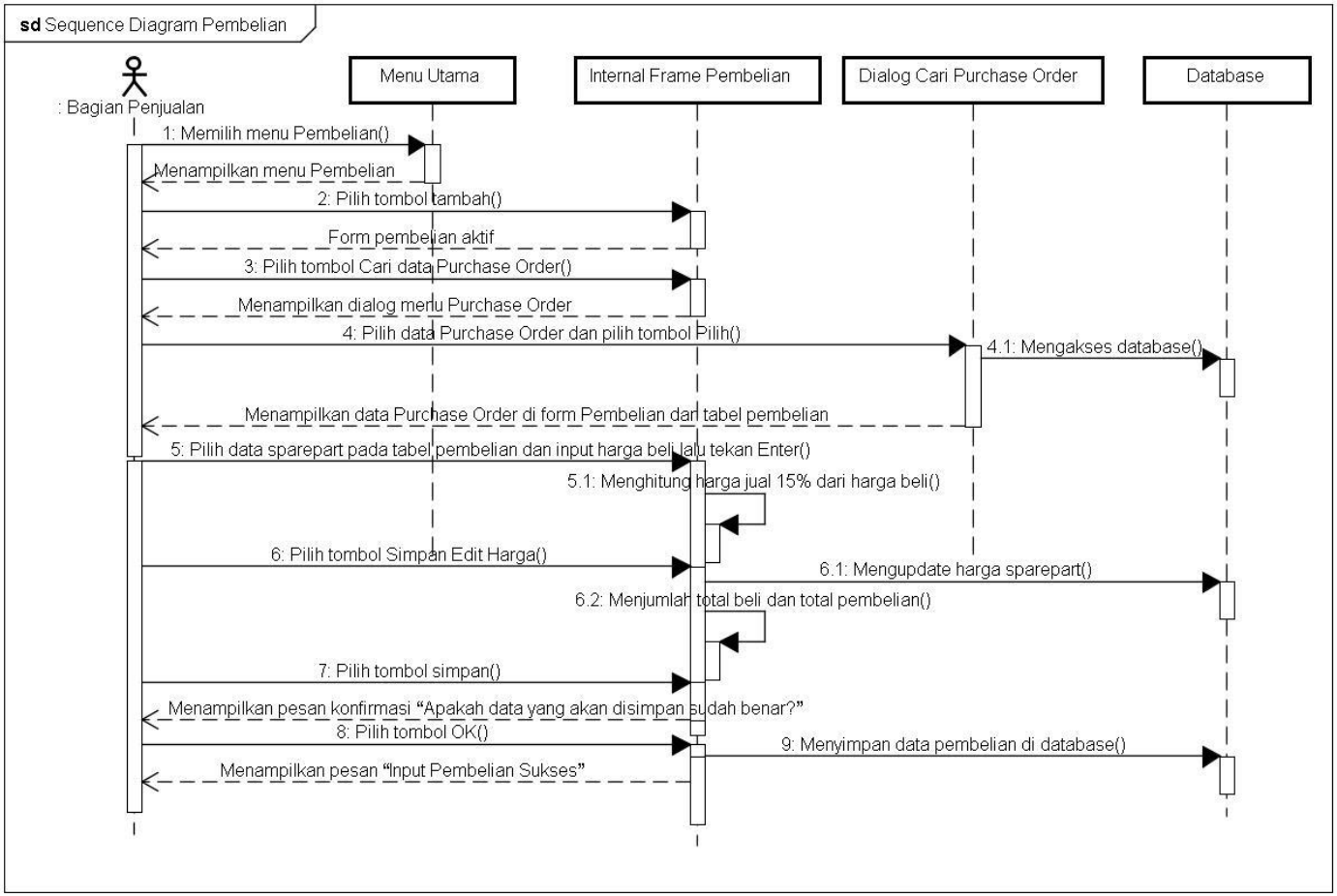
Gambar 4.16 *Sequence Diagram* Pembayaran yang diusulkan

5. *Sequence Diagram Purchase Order*



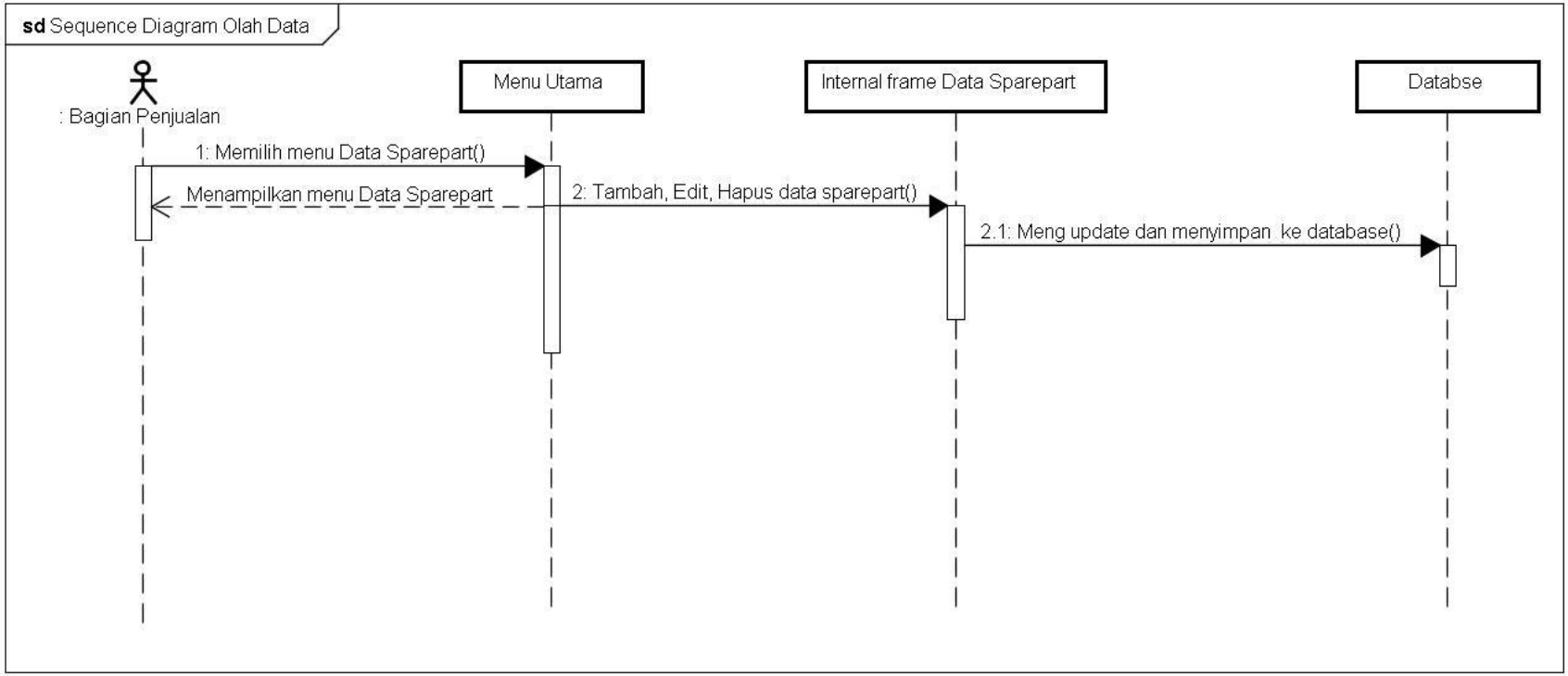
Gambar 4.17 *Sequence Diagram Purchase Order* yang diusulkan

6. *Sequence Diagram* Pembelian



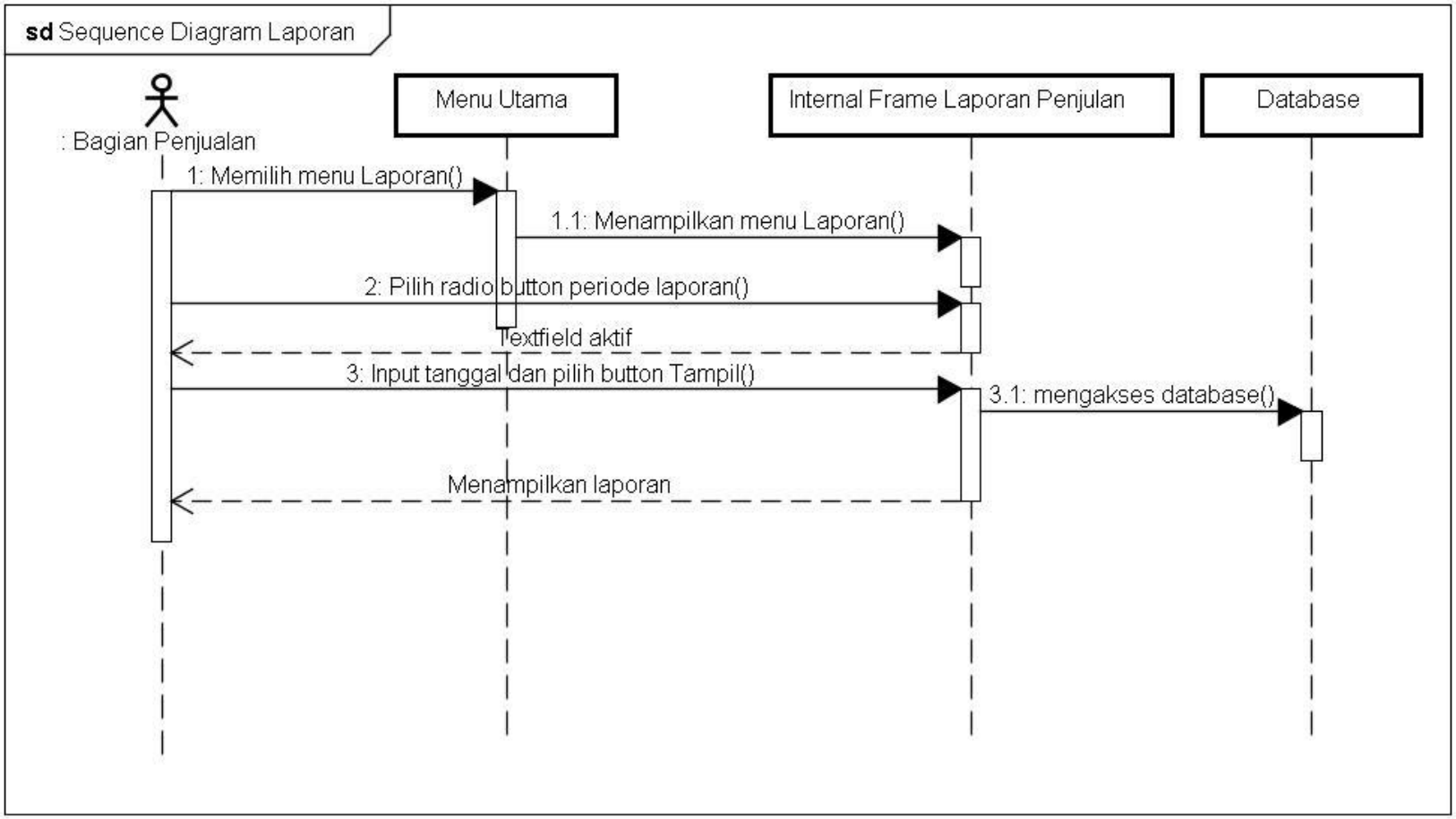
Gambar 4.18 *Sequence Diagram* Pembelian yang diusulkan

7. *Sequence Diagram* Olah Data Master



Gambar 4.19 *Sequence Diagram* Olah Data Master yang diusulkan

8. *Sequence Diagram* Laporan

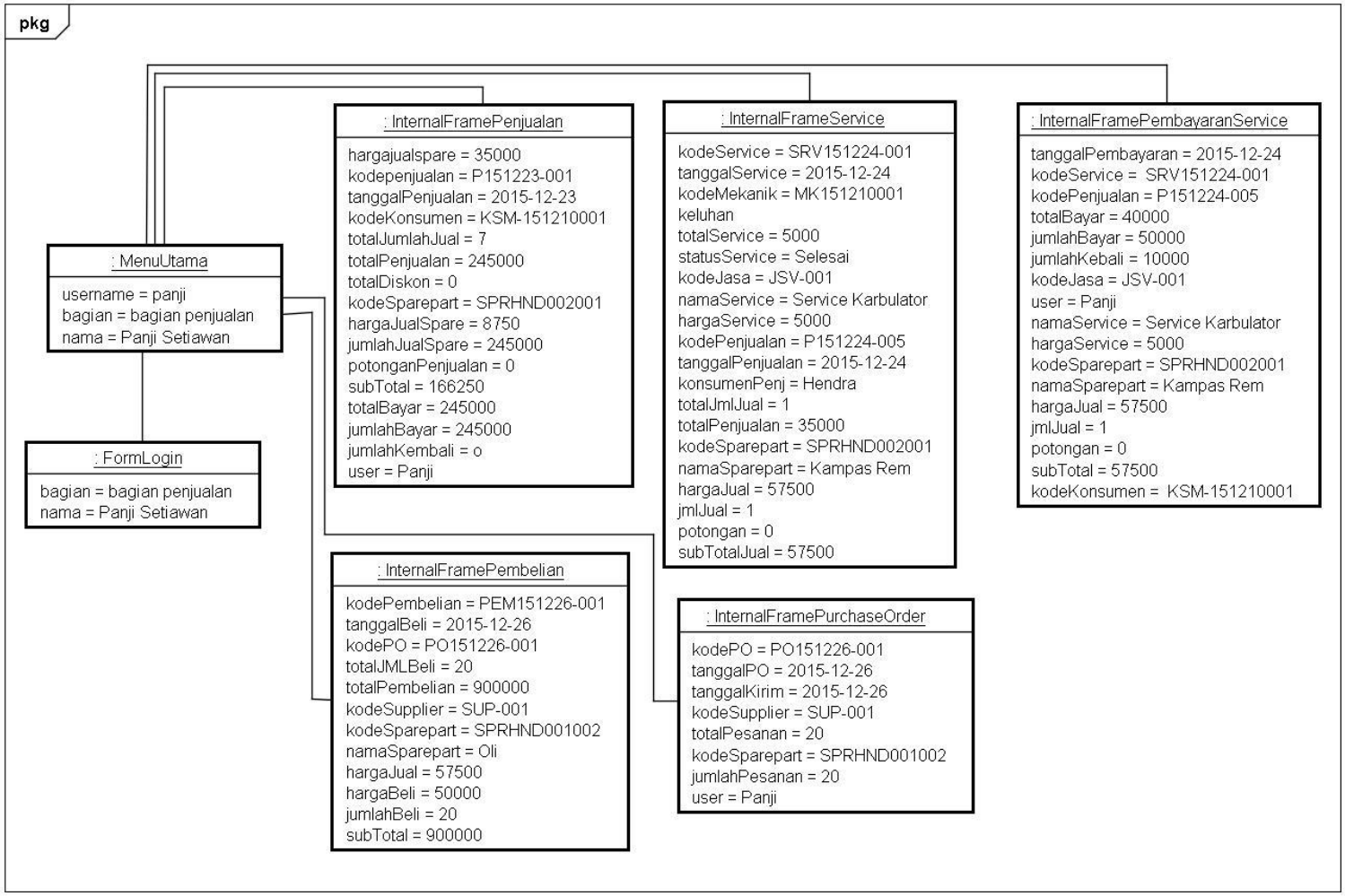


Gambar 4.20 *Sequence Diagram* Laporan yang diusulkan

4.1.4 Perancangan Data

4.1.4.1 *Class diagram*

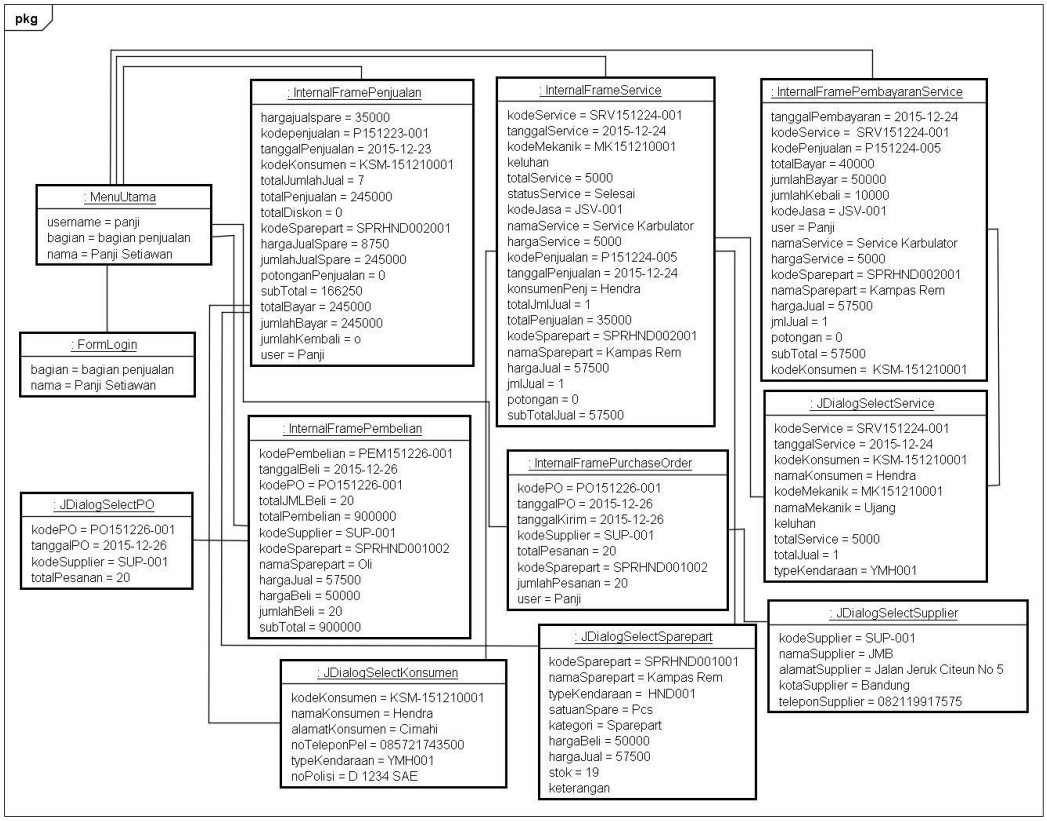
Class Diagram membantu kita dalam visualisasi struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class Diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas didalam model desain dari suatu sistem.



Gambar 4.21 *Class Diagram* Sistem Informasi Penjualan *Sparepart* dan *Service* Motor pada Bengkel Djaya Motor

4.1.4.2 *Object diagram*

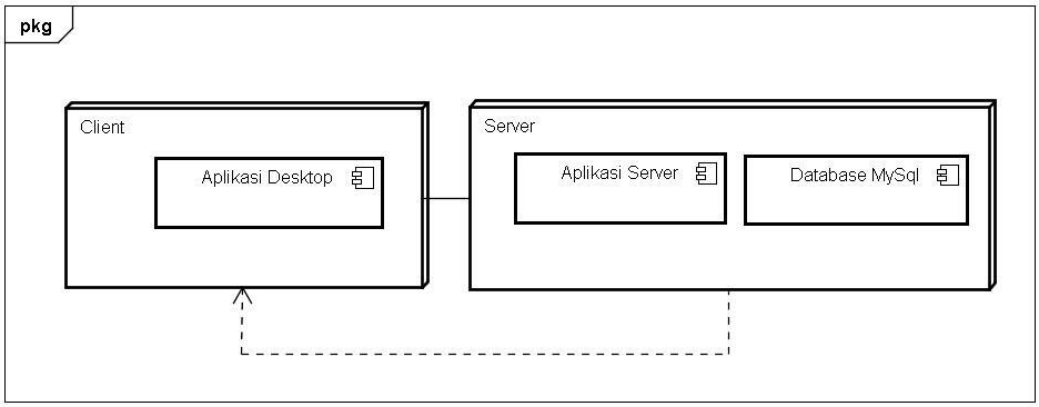
Objek diagram adalah diagram yang memberikan gambaran struktur model sebuah sistem, dalam kurun waktu tertentu. Berikut adalah objek diagram yang Sistem Informasi Penjualan *Sparepart* dan *Service* Motor pada bengkel Djaya Motor.



Gambar 4.22 *Object Diagram* Sistem Informasi Penjualan *Sparepart* dan *Service* Motor pada Bengkel Djaya Motor

4.1.4.3 *Deployment diagram*

Berikut ini *Deployment* diagram yang menggambarkan susunan fisik dan perangkat lunak dalam aplikasi :



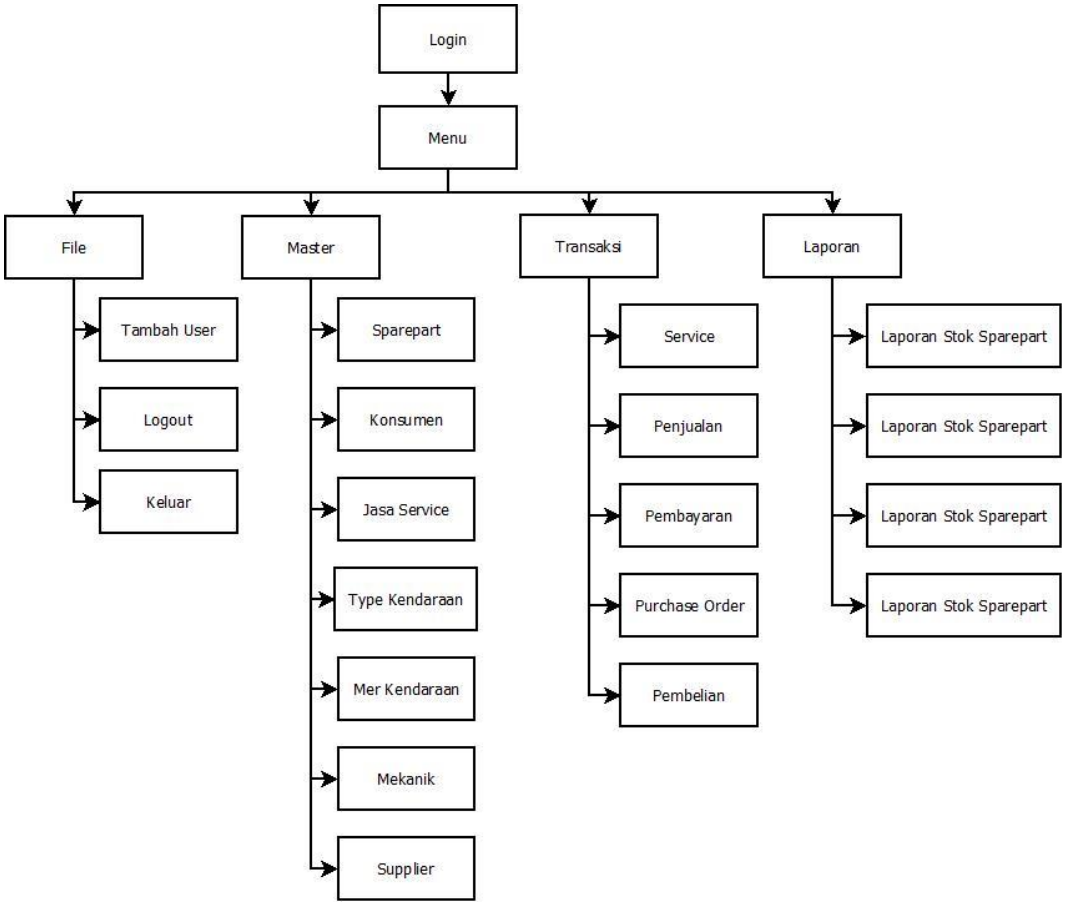
Gambar 4.23 *Deployment Diagram* Sistem Informasi Penjualan *Sparepart* dan *Service* Motor pada Bengkel Djaya Motor

4.2 Perancangan Antar Muka

Sub bab ini membahas mengenai struktur menu, perancangan *input output*, yang akan digunakan pada pembuatan Sistem Informasi Penjualan *Spareaprt* dan *Service* Motor pada Bengekel Djaya Motor.

4.2.1 Struktur menu

Struktur menu digunakan untuk memudahkan pemakai dan juga sebagai petunjuk dalam mengoperasikan sistem informasi Penjualan dan pembelian ini, agar pemakai tidak mengalami kesulitan dalam memilih menu-menu yang diinginkan. Menu-menu tersebut akan tampil sesuai hak akses dari masing-masing *User*.



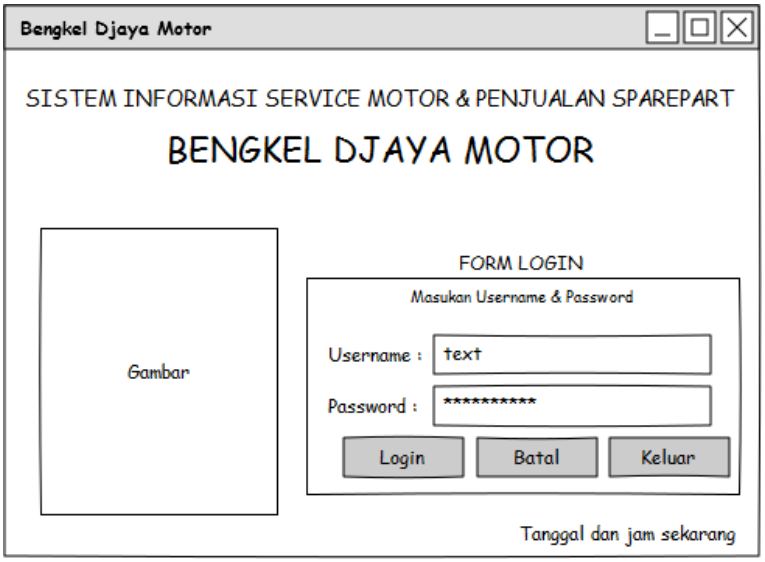
Gambar 4.24 Struktur Menu

4.2.2 Perancangan *input*

Perancangan *input* pada perangkat lunak ini terdiri dari beberapa rancangan antar muka. Berikut rancangan tampilan *input* dari perangkat lunak ini :

1. Rancangan Menu *Login*

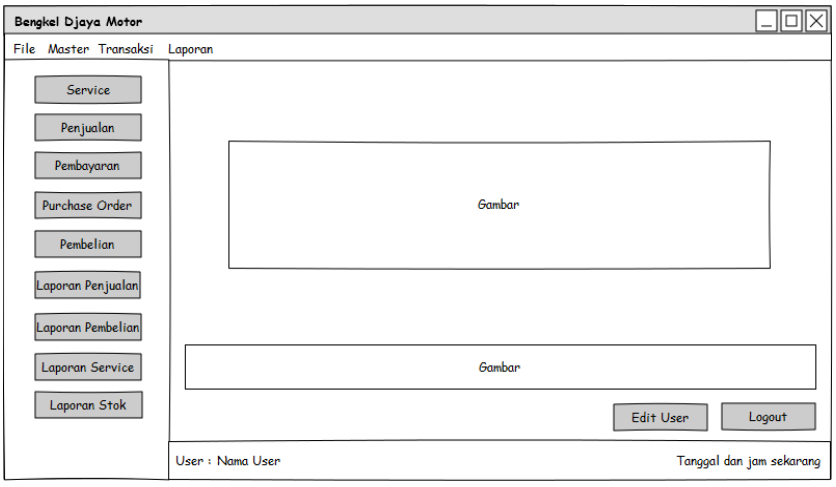
   Form ini berfungsi sebagai portal untuk masuk ke dalam Sistem Informasi Penjualan *Sparepart* dan *Service* Motor pada Bengkel Djaya Motor.



Gambar 4.25 Perancangan Antar Muka Menu *Login*

2. Rancangan Menu Utama

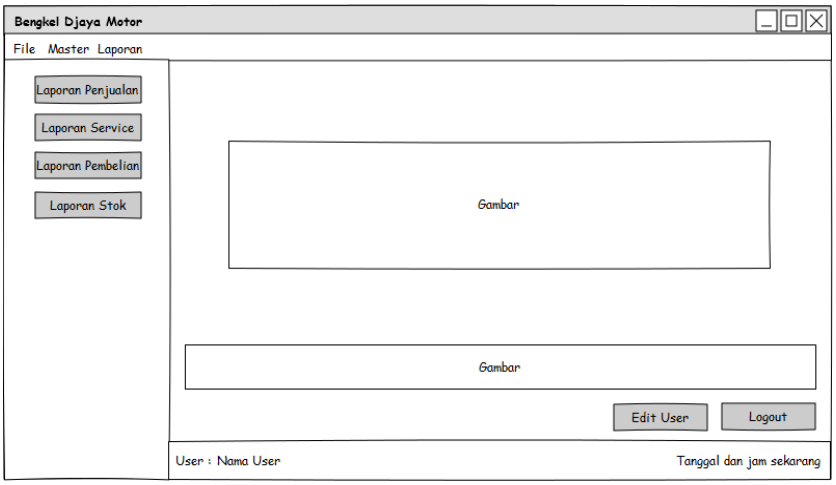
   Form ini berfungsi sebagai menu utama dari User yang memiliki hak akses sebagai Bagian Penjualan pada Bengkel Djaya Motor.



Gambar 4.26 Perancangan Antar Muka Menu Utama

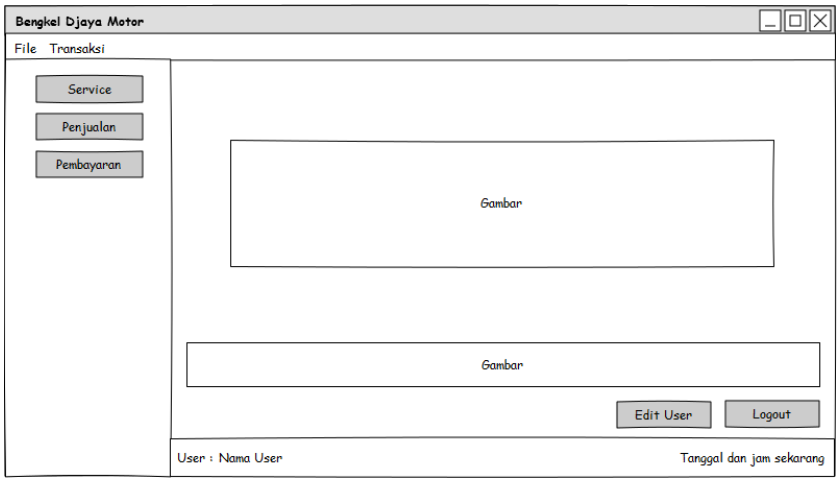
3. Rancangan Menu Utama Pemilik

   Form ini berfungsi sebagai menu utama dari User yang memiliki hak akses sebagai Bagian Pemilik pada Bengkel Djaya Motor.



Gambar 4.27 Perancangan Antar Muka Menu Utama Pemilik

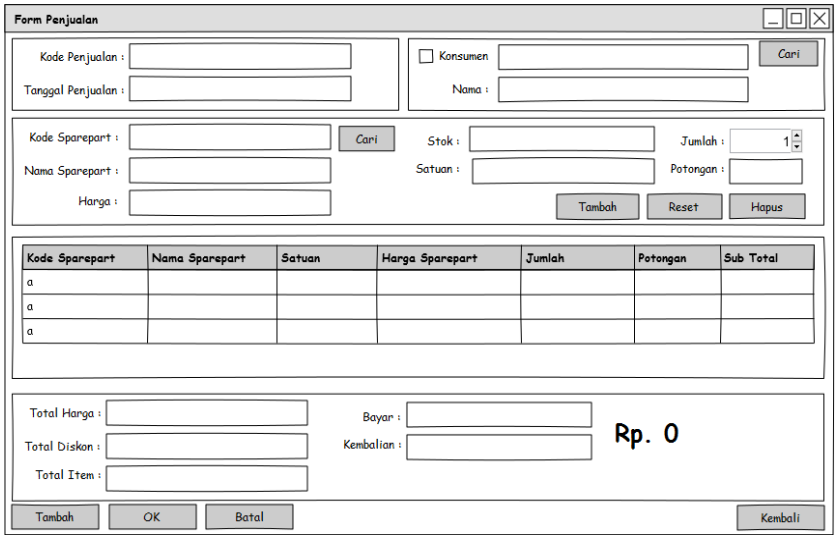
4. Rancangan Menu Utama *Service Counter*



Gambar 4.28 Perancangan Antar Muka Menu Utama *Service Counter*

5. Rancangan Menu Penjualan

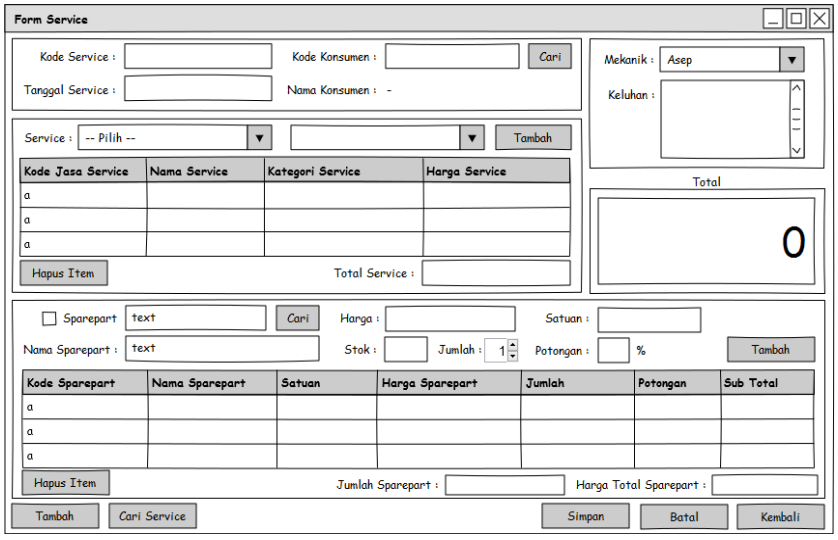
   Form ini berfungsi sebagai sarana pengolahan *data* Penjualan pada Bengkel Djaya Motor.



Gambar 4.29 Perancangan Antar Muka Menu Penjualan

6. Rancangan Menu *Service*

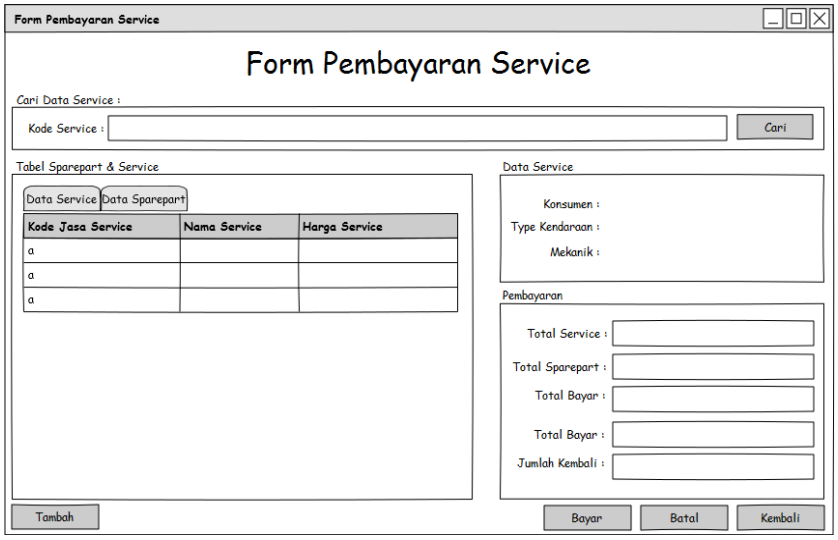
   Form ini berfungsi sebagai sarana pengolahan *data Service* pada Bengkel Djaya Motor.



Gambar 4.30 Perancangan Antar Muka Menu *Service*

7. Rancangan Menu Pembayaran

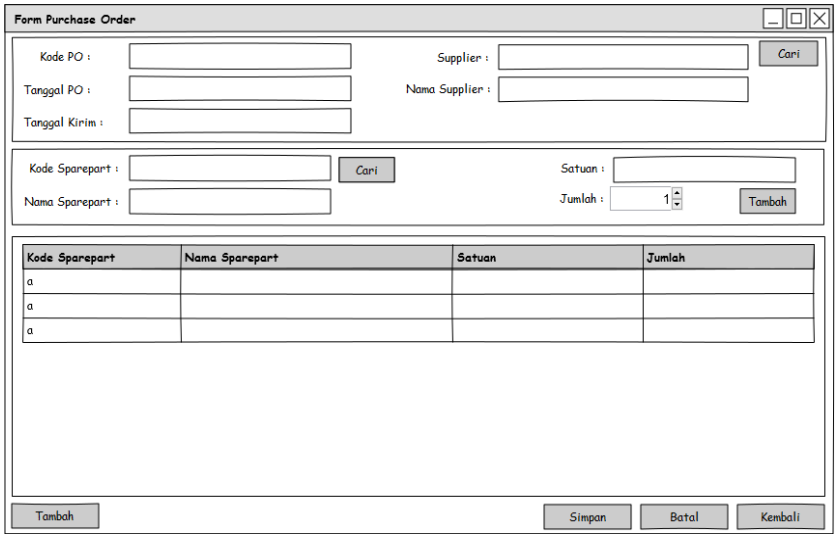
   Form ini berfungsi sebagai sarana pengolahan *data* Pembayaran *Service* pada Bengkel Djaya Motor.



Gambar 4.31 Perancangan Antar Muka Menu Pembayaran

8. Rancangan Menu *Purchase Order*

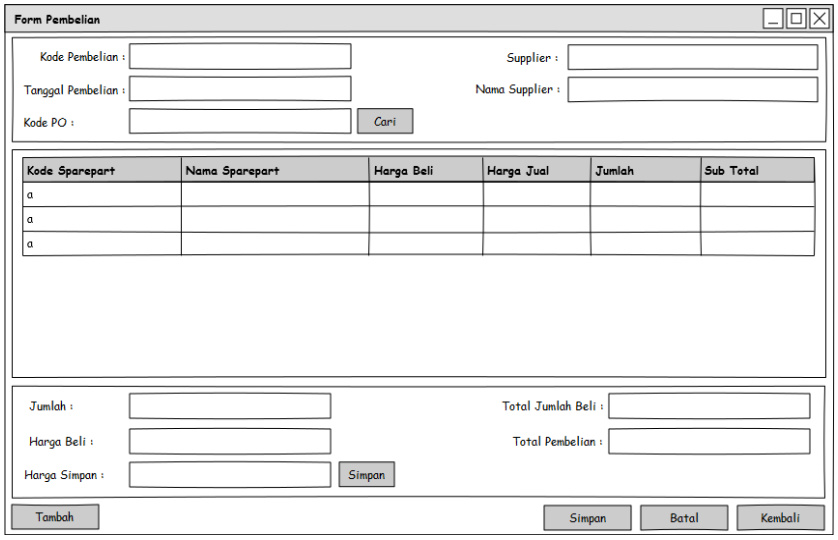
   Form ini berfungsi sebagai sarana pengolahan *data Purchase Order* pada Bengkel Djaya Motor.



Gambar 4.32 Perancangan Antar Muka Menu *Purchase Order*

9. Rancangan Menu Pembelian

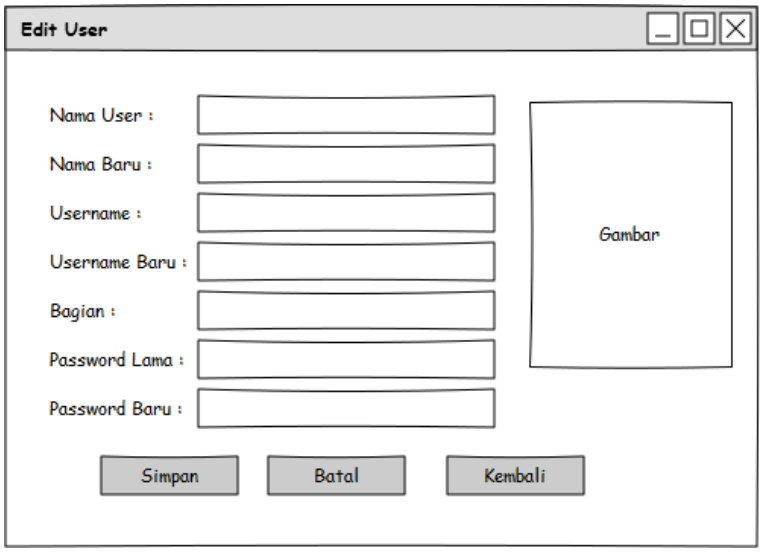
   Form ini berfungsi sebagai sarana pengolahan *data* Pembelian pada Bengkel Djaya Motor.



Gambar 4.33 Perancangan Antar Muka Menu Pembelian

10. Rancangan Menu *Edit User*

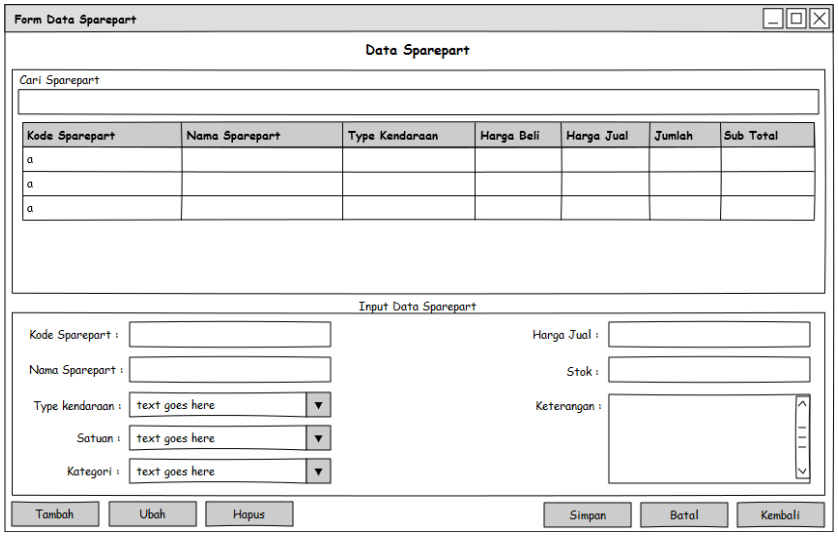
    Form ini berfungsi sebagai sarana pengolahan *data User* pada Bengkel Djaya Motor.



Gambar 4.34 Perancangan Antar Muka Menu *Edit User*

11. Rancangan Menu Data *Sparepart*

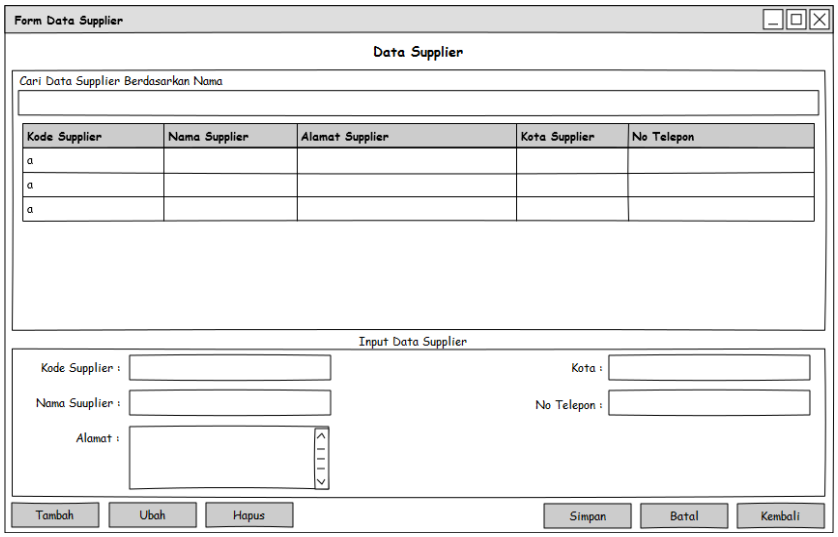
    Form ini berfungsi sebagai sarana pengolahan *data Sparepart* pada Bengkel Djaya Motor.



Gambar 4.35 Perancangan Antar Muka Menu *Data Sparepart*

12. Rancangan Menu Data *Supplier*

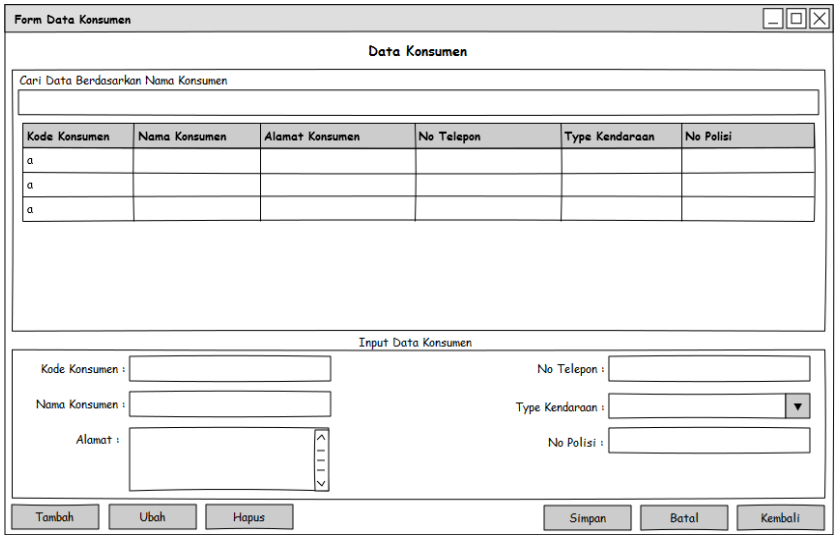
Form ini berfungsi sebagai sarana pengolahan *data Supplier* pada Bengkel Djaya Motor.



Gambar 4.36 Perancangan Antar Muka Menu *Data Supplier*

13. Rancangan Menu Data Konsumen

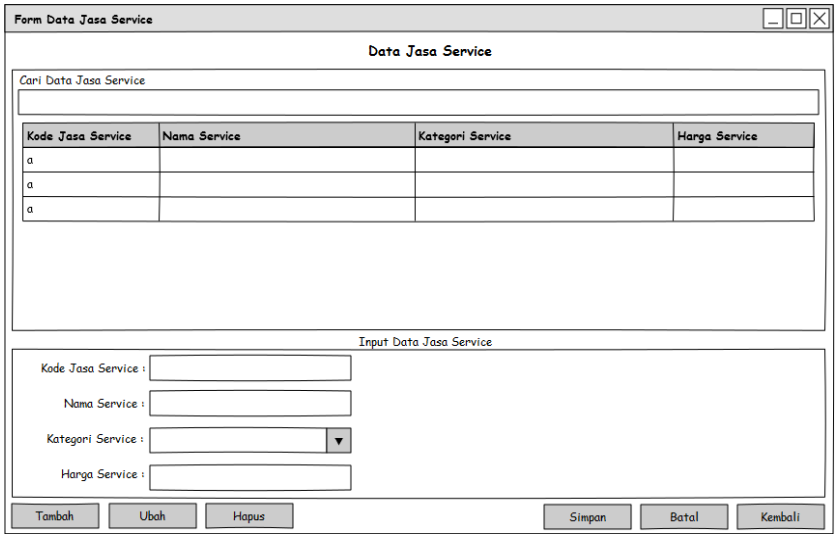
Form ini berfungsi sebagai sarana pengolahan *data* Konsumen pada Bengkel Djaya Motor.



Gambar 4.37 Perancangan Antar Muka Menu *Data* Konsumen

14. Rancangan Menu Jasa *Service*

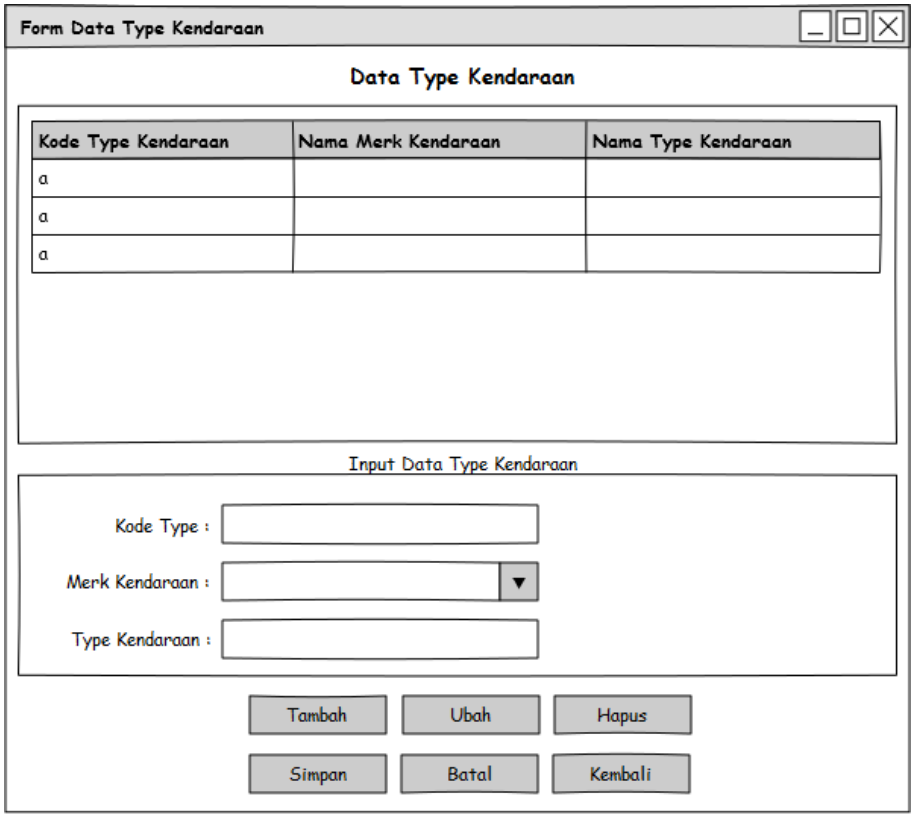
Form ini berfungsi sebagai sarana pengolahan *data* Jasa *Service* pada Bengkel Djaya Motor.



Gambar 4.38 Perancangan Antar Muka Menu *Data* Jasa *Service*

15. Rancangan Menu *Type* Kendaraan

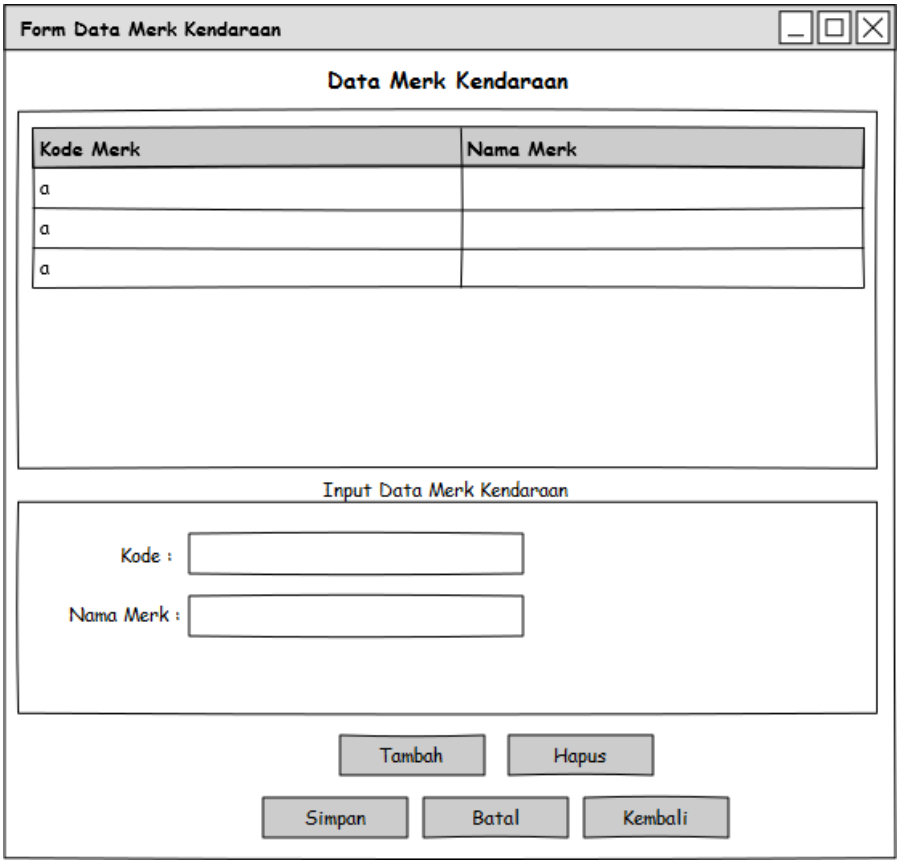
Form ini berfungsi sebagai sarana pengolahan *data Type* Kendaraan pada Bengkel Djaya Motor.



Gambar 4.39 Perancangan Antar Muka Menu *Data Type* Kendaraan

16. Rancangan Menu Merk Kendaraan

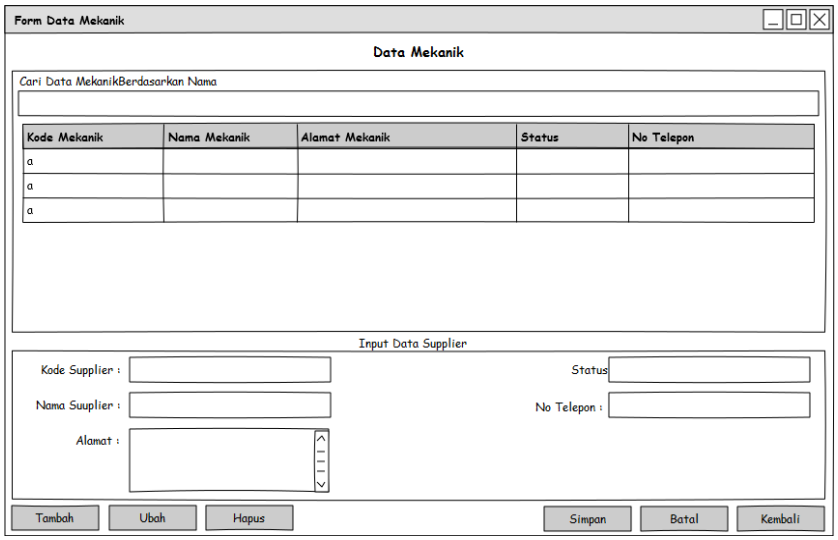
    Form ini berfungsi sebagai sarana pengolahan *data* Merk kendaraan pada Bengkel Djaya Motor.



Gambar 4.40 Perancangan Antar Muka Menu *Data* Merk Kendaraan

17. Rancangan Menu Mekanik

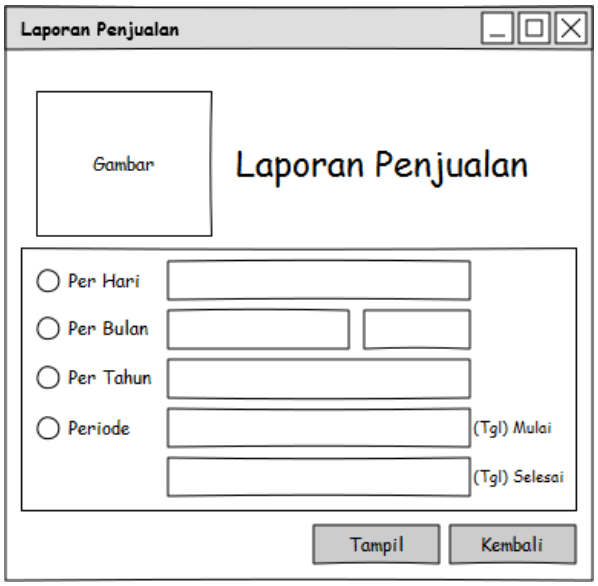
Form ini berfungsi sebagai sarana pengolahan *data* Mekanik pada Bengkel Djaya Motor.



Gambar 4.41 Perancangan Antar Muka Menu *Data* Mekanik

18. Rancangan Menu Laporan Penjualan

Form ini berfungsi sebagai sarana pengolahan pembuatan laporan penjualan pada Bengkel Djaya Motor.



Gambar 4.42 Perancangan Antar Muka Menu Laporan Penjualan

19. Rancangan Menu Laporan *Service*

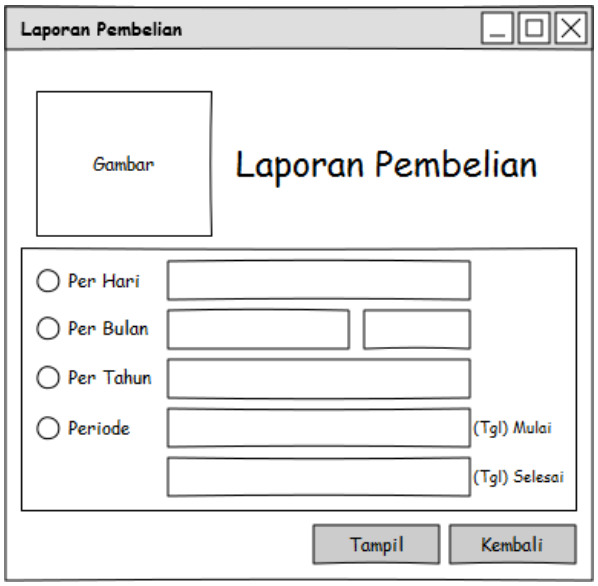
Form ini berfungsi sebagai sarana pengolahan pembuatan laporan *service* pada Bengkel Djaya Motor.



Gambar 4.43 Perancangan Antar Muka Menu Laporan *Service*

20. Rancangan Menu Pembelian

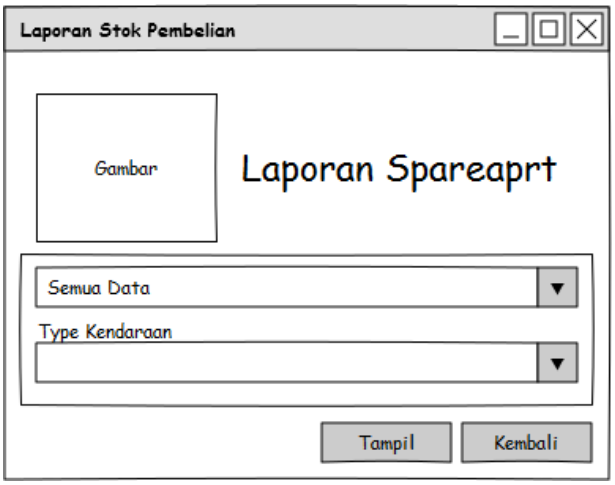
Form ini berfungsi sebagai sarana pengolahan pembuatan laporan pembelian pada Bengkel Djaya Motor.



Gambar 4.44 Perancangan Antar Muka Menu Laporan Pembelian

21. Rancangan Menu Stok *Sparepart*

Form ini berfungsi sebagai sarana pengolahan pembuatan laporan stok *sparepart* pada Bengkel Djaya Motor.



Gambar 4.45 Perancangan Antar Muka Menu Laporan Stok *Sparepart*

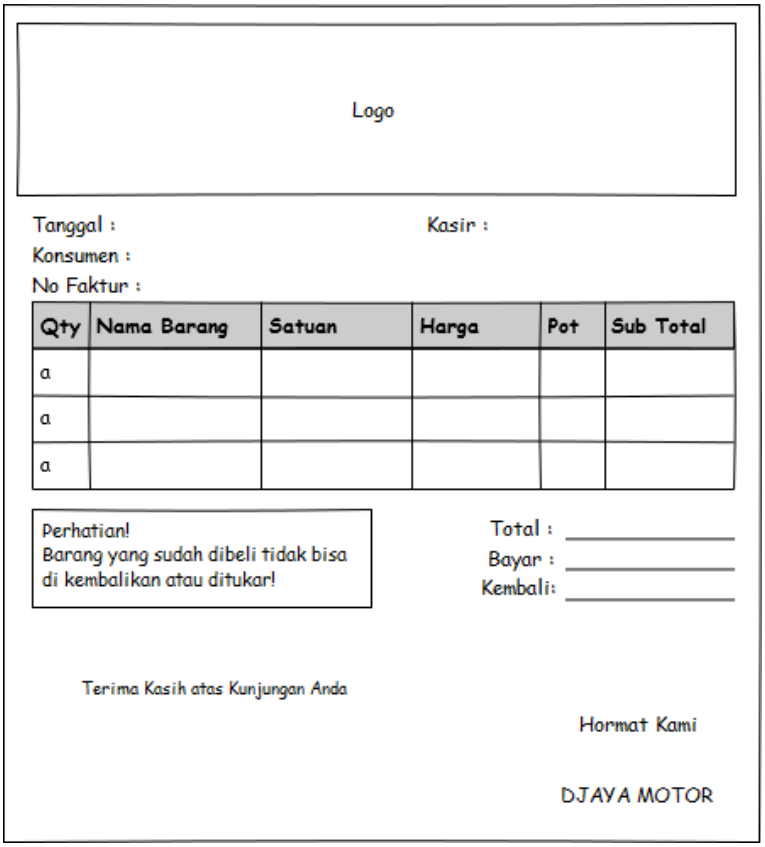
4.2.3 Perancangan *output*

Perancangan *output* dalam perangkat lunak ini berupa Laporan Penjualan *Sparepart*, Laporan *Service* Motor, Laporan Pembelian *Sparepart*, Surat Perintah Kerja, Surat *Purchase Order*, Faktur Penjualan *Sparepart*. Berikut perancangan tampilan *output* dalam perangkat lunak ini:

Laporan ini dicetak oleh Bagian Penjualan, kemudian dapat diserahkan ke Pemilik sebagai Laporan Penjualan *Sparepart* dan *Service* Motor dengan periode tertentu. Berikut perancangan *Output* Laporan Penjualan *Sparepart* dan *Service* Motor pada Bengkel Djaya Motor.

1. Perancangan Output Faktur Penjualan

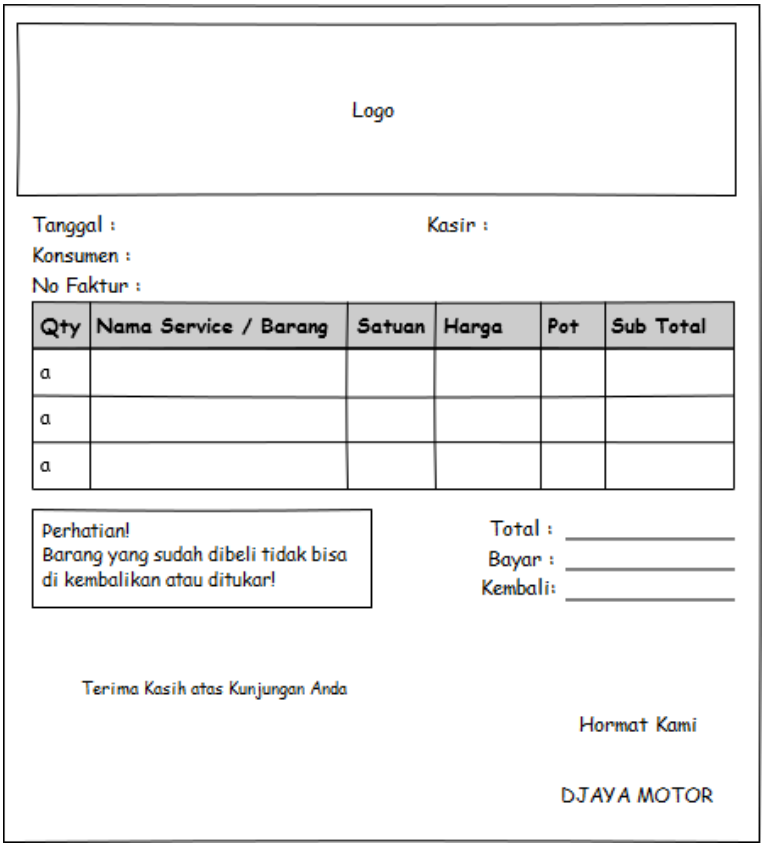
   Faktur ini dicetak oleh Bagian Penjualan , kemudian dapat diserahkan kepada Konsumen terkait sebagai tanda bukti Pembelian Barang oleh konsumen di Bengkel Djaya Motor. Berikut perancangan *Output* Faktur Penjualan Barang pada Bengkel Djaya Motor.



Gambar 4.46 Perancangan *Output* Faktur Penjualan

2. Perancangan Output Faktur *Service*

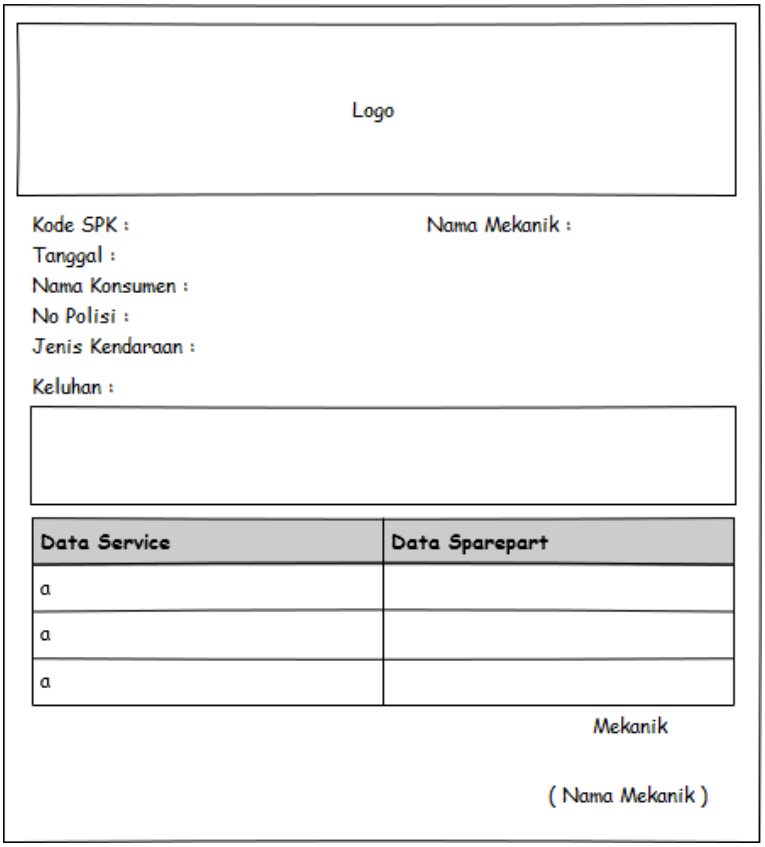
   Faktur ini dicetak oleh Bagian Penjualan , kemudian dapat diserahkan kepada Konsumen terkait sebagai tanda bukti *Service* Motor oleh konsumen di Bengkel Djaya Motor. Berikut perancangan *Output* Faktur *Service* Motor pada Bengkel Djaya Motor.



Gambar 4.47 Perancangan *Output* Faktur *Service*

3. Perancangan Output Surat Perintah Kerja

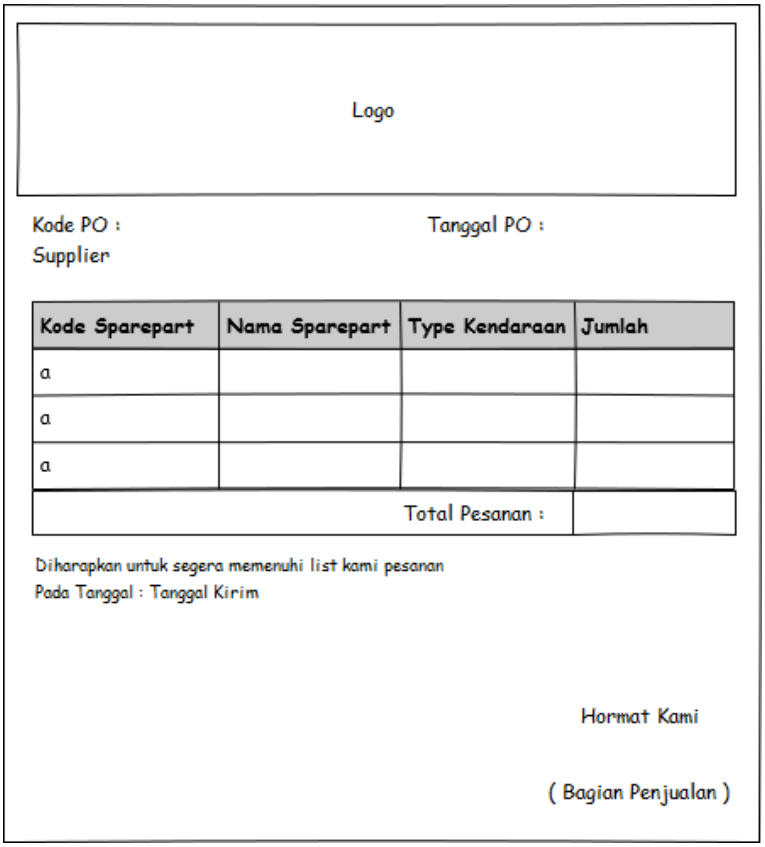
   Surat ini dicetak oleh Bagian Penjualan , kemudian dapat diserahkan kepada Mekanik sebagai perintah kerja. Berikut perancangan *Output* Surat Perintah Kerja pada Bengkel Djaya Motor.



Gambar 4.48 Perancangan *Output* Surat Perintah Kerja

4. Perancangan Output Surat *Purchase Order*

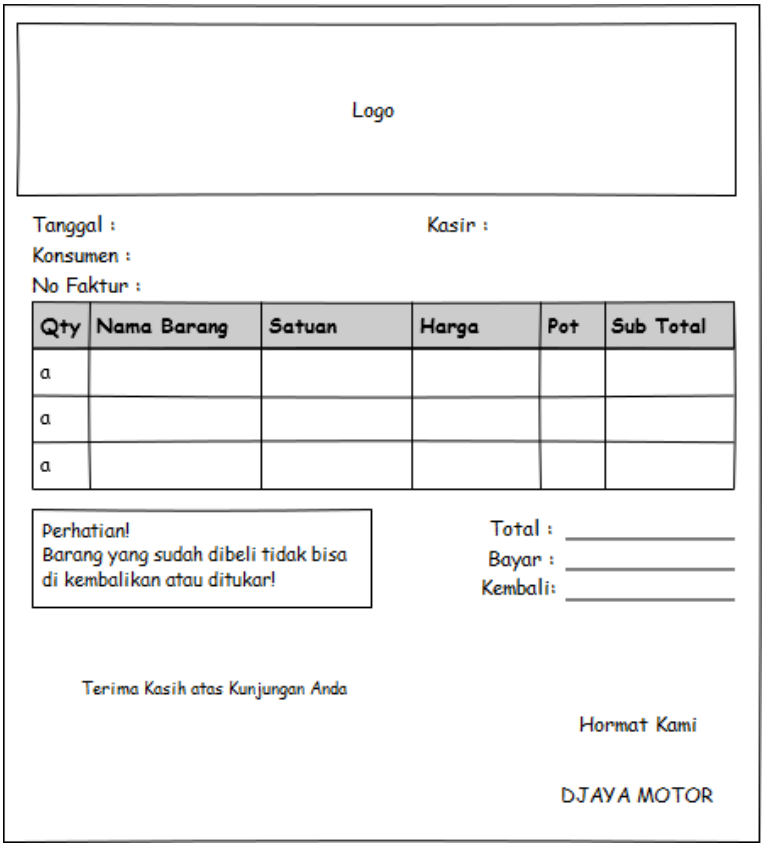
   Surat ini dicetak oleh Bagian Penjualan , kemudian dapat diserahkan kepada *Supplier* sebagai permintaan pengiriman barang sesuai dengan surat *purchase order*. Berikut perancangan *Output* Surat *Purchase Order* pada Bengkel Djaya Motor.



Gambar 4.49 Perancangan *Output* Surat *Purchase Order*

5. Perancangan Output Laporan Penjualan

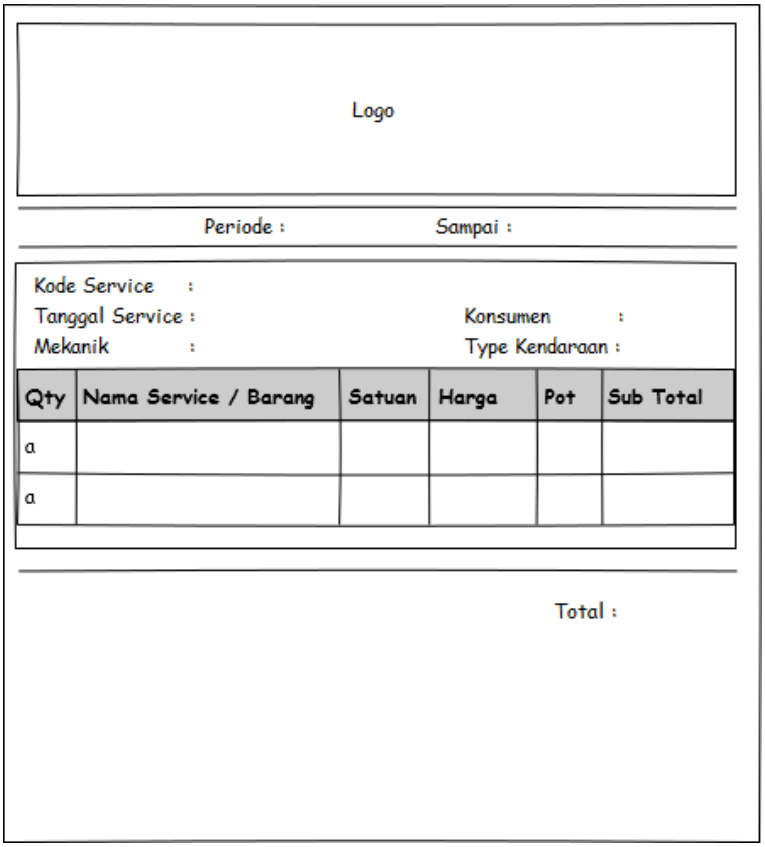
   Laporan ini dicetak oleh Bagian Penjualan, kemudian dapat diserahkan ke Pemilik sebagai Laporan Penjualan Barang dengan periode tertentu. Berikut perancangan *Output* Laporan Penjualan Barang pada Bengkel Djaya Motor.



Gambar 4.50 Perancangan *Output* Laporan Penjualan

6. Perancangan Output Laporan *Service*

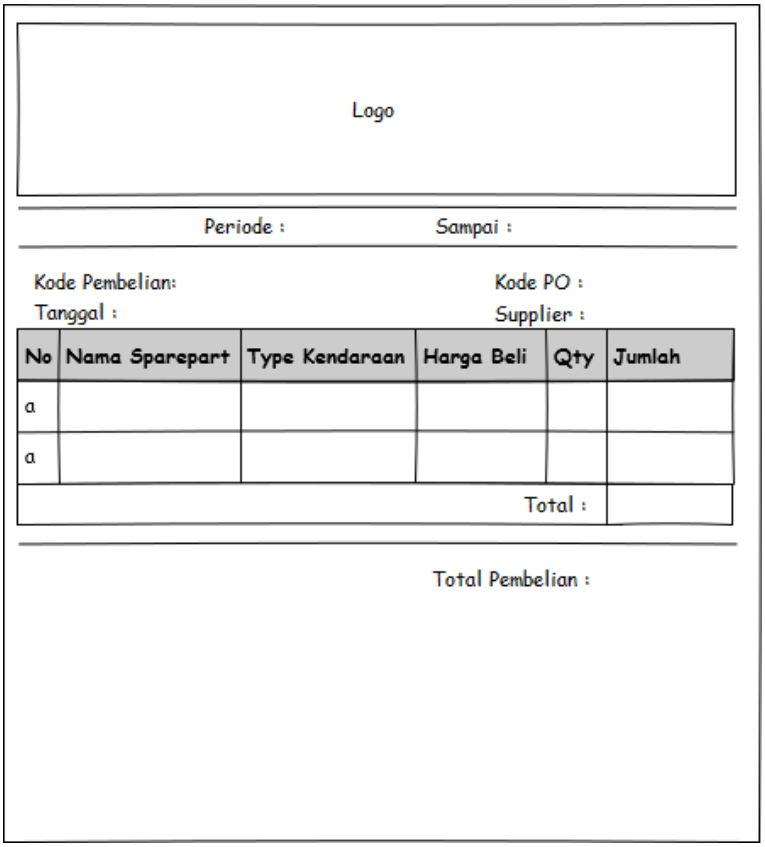
   Laporan ini dicetak oleh Bagian Penjualan, kemudian dapat diserahkan ke Pemilik sebagai Laporan *Service* Motor dengan periode tertentu. Berikut perancangan *Output* Laporan *Service* Motor pada Bengkel Djaya Motor.



Gambar 4.51 Perancangan *Output* Laporan *Service*

7. Perancangan Output Laporan Pembelian

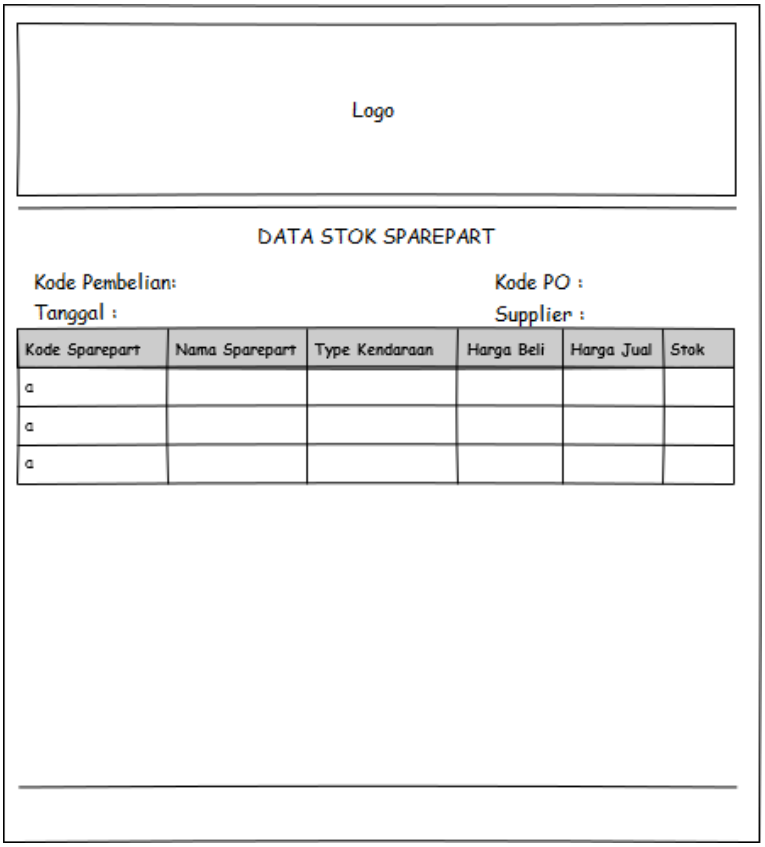
   Laporan ini dicetak oleh Bagian Penjualan, kemudian dapat diserahkan ke Pemilik sebagai Laporan Pembelian Barang dengan periode tertentu. Berikut perancangan *Output* Laporan Pembelian Barang pada Bengkel Djaya Motor.



Gambar 4.52 Perancangan *Output* Laporan Pembelian

8. Perancangan Output Laporan Stok *Sparepart*

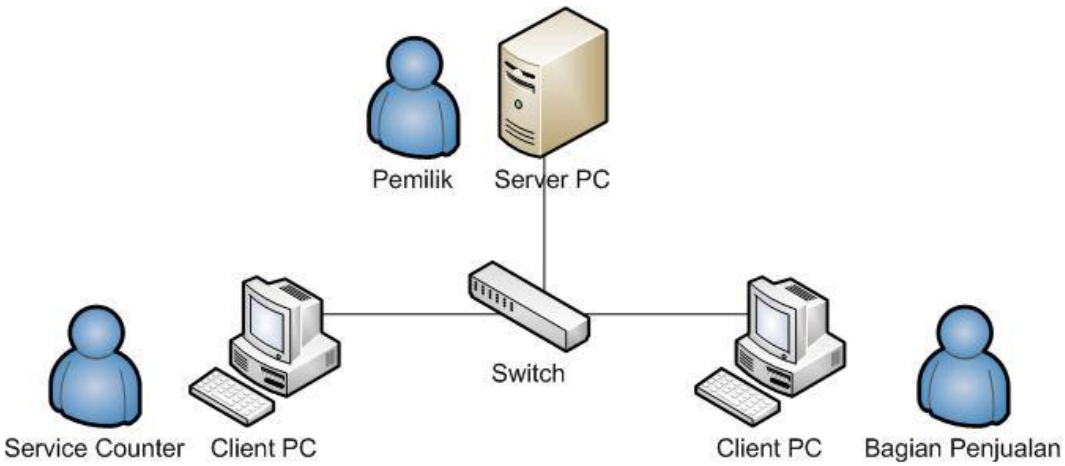
   Laporan ini dicetak oleh Bagian Penjualan, kemudian dapat diserahkan ke Pemilik sebagai Laporan Stok Barang. Berikut perancangan *Output* Laporan Stok Barang pada Bengkel Djaya Motor.



Gambar 4.53 Perancangan *Output* Laporan Stok *Sparepart*

4.3 Perancangan Arsitektur Jaringan

Arsitektur jaringan yang akan digunakan di bengkel Djaya Motor 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 4.54 Perancangan Arsitektur Jaringan

Gambar diatas menunjukan arsitektur jaringan dimana Bagian Penjualan bertugan mengelola transaksi penjualan, pembelian *sparepart* dan *service* motor. Sedangkan Pemilik mengelola data *user*, data mekanik dan menerima laporan penjualan, laporan pembelian, laporan *service* motor dan laporan stok *sparepart*.

4.4 Pengujian

Pengujian sistem perlu dilakukan untuk memastikan sistem dapat berjalan dengan semestinya. Terdapat dua teknik pengujian sistem perangkat lunak yaitu dengan menggunakan *white box* dan *black box testing.* Secara garis besar *white box testing* digunakan untuk menguji algoritma dari suatu sistem dengan kata lain melihat alur proses pada *source kode* yang dimiliki suatu sistem. Sedangkan *black box testing* digunakan untuk menguji fungsionalitas suatu sistem dengan kata lain mengabaikan algoritma pada suatu sistem atau alur proses pada *source kode*.

Teknik yang akan digunakan pada pengujian ini adalah teknik *black box testing*.

4.4.1 Rencana pengujian

Tahapan-tahapan yang akan dilakukan pada rencana pengujian ini adalah sebagai berikut :

Tabel 4.9 Rencana Pengujian Sistem

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Verifikasi *Username*	*Black Box*
Penjualan	Fungsionalitas penjualan sparepart	*Black Box*
Service	Fungsionalitas *Service*	*Black Box*
Purchase Order	Fungsionalitas *Purchase order*	*Black Box*
Pembelian	Fungsionalitas Pembelian	*Black Box*
Laporan	Fungsionalitas Laporan	*Black Box*

4.4.2 Kasus dan hasil pengujian

Tabel 4.10 Pengujian Proses Login

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Username “panji” Password “1234”	Mucul pesan “berhasil login, lalu masuk ke halaman utama dengan hak akses bagian penjualan.	Berhasil

2	Username “iwan” Password “1234”	Mucul pesan “berhasil login, lalu masuk ke halaman utama hak akses pemilik.	Berhasil

Tabel 4.11 Pengujian Proses Penjualan

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik tombol Tambah	*Form* penjualan aktif.	Berhasil
2	Klik tombol Cari *Sparepart*	Menampilkan data *sparepart*.	Berhasil
3	Klik tombol Pilih *Sparepart*	Mengambil data *sparepart* ke dalam *form* penjualan.	Berhasil
4	Klik tombol *Reset*	Menghapus inputan data *sparepart*.	Berhasil
5	Klik tombol Hapus Item	Menghapus item (*sparepart*) yang telah di pilih.	Berhasil
6	Klik tombol OK	Menampilkan pesan “Apakah data yang akan disimpan sudah benar?” dan mencetak faktur.	Berhasil
7	Klik tombol Batal	Menghapus semua inputan dan *form* penjualan tidak aktif.	Berhasil

Tabel 4.12 Pengujian Proses *Service*

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik tombol Tambah	Form *service* aktif.	Berhasil
2	Klik tombol Cari *Service*	Mencari *service* dengan status menunggu atau belum mekakukan transaksi pembayaran	Berhasil
3	Klik tombol tambah konsumen	Menampilkan data konsumen	Berhasil
4	Klik *combobox* kategori *service*	Menampilkan data jasa *service* yang bisa dipilih sesuai katagori	Berhasil
4	Klik *combobox* pilih jasa *service*	Menampilkan jasa *service* yang sudah di pilih.	Berhasil
5	Klik tombol tambah jasa *service*	Menambahkan jasa *service* ke *form service*	Berhasil
6	Klik tombol hapus item jasa *service*	Menghapus jasa *service* yang telah di pilih.	Berhasil
7	Klik *checkbox sparepart*	Membuka *form* penjualan *sparepart* di *form service*	Berhasil
8	Klik tombol pilih *sparepart*	Menampilkan data *sparepart*.	Berhasil
9	Klik tombol tambah *sparepart*	Menambahkan *sparepart* ke *form* penjualan didalam *form service*	Berhasil
10	Pilih *combobox* mekanik	Menampilkan mekanik yang bisa di pilih	Berhasil

11	Klik tombol simpan	Menyimpan transaksi *service*.	Berhasil
12	Klik tombol Batal	Mengosongkan *form service*	Berhasil
13	Klik tombol keluar	Keluar dari *form service*	Berhasil

Tabel 4.13 Pengujian Proses Pembayaran

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik tombol tambah pembayaran	*Form* pembayaran *service* terbuka	Berhasil
2	Klik tombol cari	Menampilkan data *service* yang ber status menunggu.	Berhasil
3	Klik tombol bayar	Input pembayaran sukses dengan mencetak faktur *service* motor.	Berhasil
4	Batal	Mengosongkan form pembayaran *service*.	Berhasil
5	Keluar	Keluar dari *form* pembayarn *service*.	Berhasil

Tabel 4.14 Pengujian Proses *Purchase Order*

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik tombol tambah *purchase order*	*Form purchase order* terbuka	Berhasil
2	Klik tombol cari *supplier*	Menampilkan *form supplier* untuk memilih	Berhasil

3	Klik tombol cari *sparepart*	Menampilkan *form sparepart* untuk memilih *sparepart*	Berhasil
4	Klik tombol tambah *sparepart*	Mengambil data *sparepart* dari *form sparepart* ke *form* pemesanan (*Purcahse Order*)	Berhasil
5	Hapus item	Menghapus *sparepart* yang telah dipilih	Berhasil
6	Simpan	Menyimpan data *purchase order*.	Berhasil
7	Batal	Mengosongkan *form purcahse order* (pemesanan)	Berhasil
8	keluar	Keluar dari *form purchase order*	Berhasil

Tabel 4.15 Pengujian Proses Pembelian

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik tombol tambah pembelian	*Form* pembelian terbuka	Berhasil
2	Klik tombol cari data *purchase order* yang telah di buat	Menampilkan data *purchase order* yang telah di buat.	Berhasil
3	Klik tombol simpan *edit* harga	menyimpan data pembelian yang telah di *edit* (harga beli/jumlah)	Berhasil

4	Simpan pembelian	Menyimpan data pembelian.	Berhasil
5	Batal	Mengosongkan *form* pembelian	Berhasil
6	Keluar	Keluar dari form pembelian	Berhasil

Tabel 4.16 Pengujian *Form* Laporan Penjualan

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik *Radiobutton* perhari, pilih tanggal	Menampilkan laporan penjualan secara berkala per hari	Berhasil
2	Klik *Radiobutton* perbulan, pilih bulan	Menampilkan laporan penjualan secara berkala per bulan	Berhasil
3	Klik *Radiobutton* pertahun, pilih tahun	Menampilkan laporan penjualan secara berkala per tahun	Berhasil
4	Klik *Radiobutton* perpriode, klik *JDate Chosser*	Menampilkan laporan penjualan secara berkala per periode yang telah di pilih	Berhasil
5	Klik tombol Tampilkan	Menampilkan laporan sesuai dengan *radio button* yang dipilih	Berhasil

Tabel 4.17 Pengujian *Form* Laporan *Service*

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik *Radiobutton* perhari, pilih tanggal	Menampilkan laporan *Service* secara berkala per hari	Berhasil
2	Klik *Radiobutton* perbulan, pilih bulan	Menampilkan laporan *Service* secara berkala per bulan	Berhasil
3	Klik *Radiobutton* pertahun, pilih tahun	Menampilkan laporan Service secara berkala per tahun	Berhasil
4	Klik *Radiobutton* perpriode, klik *JDate Chosser*	Menampilkan laporan *Service* secara berkala per periode yang telah di pilih	Berhasil
5	Klik tombol Tampilkan	Menampilkan laporan sesuai dengan *radio button* yang dipilih	Berhasil

Tabel 4.18 Pengujian *Form* Laporan Pembelian

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik *Radiobutton* perhari, pilih tanggal	Menampilkan laporan pembelian secara berkala per hari	Berhasil
2	Klik *Radiobutton* perbulan, pilih bulan	Menampilkan laporan pembelian secara berkala per bulan	Berhasil

3	Klik *Radiobutton* pertahun, pilih tahun	Menampilkan laporan pembelian secara berkala per tahun	Berhasil
4	Klik *Radiobutton* perpriode, klik *JDate Chosser*	Menampilkan laporan pembelian secara berkala per periode yang telah di pilih	Berhasil
5	Klik tombol Tampilkan	Menampilkan laporan sesuai dengan *radio button* yang dipilih	Berhasil

Tabel 4.19 Pengujian *Form* Laporan Stok *Sparepart*

No	Data Masukan	Hasil yang diharapkan	Kesimpulan
1	Klik *ComboBox*,	Dapat memilih berdasrakan semua data, per jenis kendaraan, stok minimum	Berhasil
2	Klik tombol Tampilkan	Menampilkan laporan sesuai dengan yang dipilih di *combobox*	Berhasil

4.4.3 Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus sample uji yang telah dilakukan memberikan kesimpulan bahwa perangkat lunak ini dapat digunakan dengan baik, namun pengujian tersebut dapat dikatakan belum sempurna, karena hanya dilakukan pada satu sisi pengujian. Dari semua yang telah dilakukan dalam pengujian ini diharapkan dapat mewakili pengujian fungsi yang lain dalam program

Sistem Informasi Penjualan Sparepart dan *Service* Motor pada Bengkel Djaya Motor ini.

4.5 Implementasi

Implementasi merupakan sebuah akhir dari pengembangan sistem informasi. Implementasi sistem tentunya mengacu pada rancangan yang telah dibuat, pada umumnya implementasi sistem diperlukan sebuah konfigurasi pada perangkat lunak dan perangkat kerasnya supaya sistem informasi yang dibuat bisa berjalan dengan sebagaimana mestinya.

4.5.1 Implementasi perangkat lunak

Perangkat lunak *(Software)* digunakan untuk mendukung sistem operasi (SO) dan aplikasi *Database*. Perangkat lunak yang dibutuhkan oleh Komputer *server* maupun Komputer *client*, adapun kebutuhan perangkat lunak yang digunakan sebagai berikut :

1. Komputer *Server*

   Komputer *server* adalah komputer yang mempunyai kemampuan yang lebih dari komputer *client*, dimana didalamnya tersimpan data-data yang akan didistribusikan. Dimana *software* yang dibutuhkan diantaranya :

   a. Sistem Operasi *Windows* 7 Pro 64/32bit (*recomended*).
   b. *XAMPP* 1.8.3-2 (*recomended*).
   c. *MySQL*.
   d. *Netbeans IDE* 8.0 (*recomended*).
   e. *Jaspersoft Studio*-5.5.2.final (*recomended*).
   f. *Java SE Development Kit* 8.0.200 64/32bit (*recomended*).

2. Komputer *Client*

   Komputer *Client* adalah komputer yang didalamnya tersimpan data-data yang dapat digunakan secara perorangan, mengumpulkan dan menampilkan *data*, serta menyimpan *data* ke *server*. Dimana *software* yang dibutuhkan diantaranya :

   a. Sistem Operasi *Windows* XP SP 2 (*minimum*).
   b. *Java SE Development Kit* 8.0.200 64/32bit (*recomended*).
   c. *MySQL*.

4.5.2 Implementasi perangkat keras

Untuk dapat menjalankan aplikasi yang dirancang maka dibutuhkan suatu perangkat keras sebagai penunjangnya. Adapun perangkat kebutuhan perangkat keras yang dibutuhkan tersebut adalah sebagai berikut :

1. Komputer *Server* :
   a. Processor Minimal Pentium IV atau yang sekelasnya.
   b. Menggunakan minimal RAM 1gb.
   c. *Harddisk* minimal 80gb.
   d. *Keyboard*, *Monitor* dan *Mouse*.
2. Komputer *Client* :
   a. *Processor* Minimal Pentium IV atau yang sekelasnya.
   b. Menggunakan minimal RAM 512mb.
   c. *Harddisk* minimal 40gb.
   d. *Keyboard*, *Monitor* dan *Mouse*.
3. Perangkat Jaringan

a. 1 Switch 4 port.

b. Kabel UTP dan RJ45.

c. Crimping Tools.

4.5.3 Implementasi basis data

Pada bagian implementasi basis data di uraikan sintaks data sebagai berikut:

a. Sistak pembuatan Database

```
CREATE DATABASE djayamotor_db;
```

b. Sintak untuk membuat tabel konsumen

```
CREATE TABLE IF NOT EXISTS `konsumen` (
  `kode_konsumen` varchar(10) NOT NULL,
 `nama_konsumen` varchar(50) NOT NULL,
  `alamat_konsumen` text NOT NULL,
  `no_tlp_konsumen` varchar(15) NOT NULL,
  `type_kendaraan` varchar(60) NOT NULL,
  `no_polisi` varchar(10) NOT NULL,
  PRIMARY KEY (`kode_konsumen`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


ALTER TABLE `konsumen`
  ADD CONSTRAINT `konsumen_ibfk_1` FOREIGN KEY
(`type_kendaraan`) REFERENCES `type_kendaraan` (`kode_type`);
```

c. Sintak untuk membuat tabel mekanik

```
CREATE TABLE IF NOT EXISTS `mekanik` (
```

```
  `kode_mekanik` varchar(10) NOT NULL,
  `nama_mekanik` varchar(50) NOT NULL,
  `tgl_lahir` date NOT NULL,
  `alamat_mekanik` text NOT NULL,
  `status_mekanik` varchar(20) NOT NULL,
  `no_tlp_mekanik` varchar(15) NOT NULL,
  PRIMARY KEY (`kode_mekanik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

d. Sintak untuk membuat tabel supplier

```
CREATE TABLE IF NOT EXISTS `supplier` (
  `kode_supplier` varchar(10) NOT NULL,
  `nama_supplier` varchar(50) NOT NULL,
  `alamat_supplier` text NOT NULL,
  `kota_supplier` varchar(30) NOT NULL,
  `no_tlp_supplier` varchar(15) NOT NULL,
  PRIMARY KEY (`kode_supplier`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

e. Sitak untuk membuat tabel sparepart

```
CREATE TABLE IF NOT EXISTS `sparepart` (
  `kode_sparepart` varchar(20) NOT NULL,
  `nama_sparepart` varchar(50) NOT NULL,
  `type_kendaraan` varchar(10) NOT NULL,
  `satuan_sparepart` varchar(50) DEFAULT NULL,
```

```
  `kategori` varchar(30) NOT NULL,
  `harga_beli` bigint(20) NOT NULL,
  `harga_jual` bigint(20) NOT NULL,
  `stok` int(11) NOT NULL,
  `ket` text,
  PRIMARY KEY (`kode_sparepart`),
  KEY `FK_kode_type1` (`type_kendaraan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `sparepart`
  ADD CONSTRAINT `FK_kode_type1` FOREIGN KEY
(`type_kendaraan`) REFERENCES `type_kendaraan` (`kode_type`);
```

f. Sitak untuk membuat tabel po

```
CREATE TABLE IF NOT EXISTS `po` (
  `kode_po` varchar(10) NOT NULL,
  `tgl_po` date NOT NULL,
  `kode_supplier` varchar(10) NOT NULL,
  `total_pesanan` int(11) NOT NULL,
  PRIMARY KEY (`kode_po`),
  KEY `FK_kode_supplier2` (`kode_supplier`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `po`
```

```
  ADD CONSTRAINT `FK_kode_supplier2` FOREIGN KEY
(`kode_supplier`) REFERENCES `supplier` (`kode_supplier`);
```

g. Sintak untuk membuat detail_po

```
CREATE TABLE IF NOT EXISTS `detail_po` (
  `kode_po` varchar(10) NOT NULL,
  `kode_sparepart` varchar(10) NOT NULL,
  `jumlah_pesanan` int(11) NOT NULL,
  KEY `FK_kode_po1` (`kode_po`),
  KEY `FK_kode_sparepart2` (`kode_sparepart`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


ALTER TABLE `detail_po`
  ADD CONSTRAINT `FK_kode_po1` FOREIGN KEY (`kode_po`)
REFERENCES `po` (`kode_po`),
  ADD CONSTRAINT `FK_kode_sparepart2` FOREIGN KEY
(`kode_sparepart`) REFERENCES `sparepart` (`kode_sparepart`);
```

h. Sintak untuk membuat tabel pembelian

```
CREATE TABLE IF NOT EXISTS `pembelian_spare` (
  `kode_pem_spare` varchar(10) NOT NULL,
  `tanggal_pem` date NOT NULL,
  `kode_po` varchar(10) NOT NULL,
  `kode_supplier` varchar(10) NOT NULL,
  `total_jml_beli` int(11) NOT NULL,
```

```
  `total_pem` double NOT NULL,
  PRIMARY KEY (`kode_pem_spare`),
  KEY `FK_kode_supplier1` (`kode_supplier`),
  KEY `FK_kode_po2` (`kode_po`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `pembelian_spare`
  ADD CONSTRAINT `FK_kode_po2` FOREIGN KEY (`kode_po`)
REFERENCES `po` (`kode_po`),
  ADD CONSTRAINT `FK_kode_supplier1` FOREIGN KEY
(`kode_supplier`) REFERENCES `supplier` (`kode_supplier`);
```

i. Sintak untuk membuat tabel detail_pembelian

```
CREATE TABLE IF NOT EXISTS `detail_pembelian` (
  `kode_pem_spare` varchar(10) NOT NULL,
  `kode_sparepart` varchar(10) NOT NULL,
  `harga_beli_lama` bigint(20) NOT NULL,
  `jumlah_beli_spare` int(11) NOT NULL,
  `sub_total_pem` double NOT NULL,
  KEY `FK_kode_pem_spare1` (`kode_pem_spare`),
  KEY `FK_kode_sparepart1` (`kode_sparepart`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `detail_pembelian`
```

```
 ADD CONSTRAINT `FK_kode_pem_spare1` FOREIGN KEY
(`kode_pem_spare`) REFERENCES `pembelian_spare`
(`kode_pem_spare`),
 ADD CONSTRAINT `FK_kode_sparepart1` FOREIGN KEY
(`kode_sparepart`) REFERENCES `sparepart` (`kode_sparepart`);
```

j. Sintak untuk membuat tabel penjualan

```
CREATE TABLE IF NOT EXISTS `penjualan_spare` (
 `kode_penjualan` varchar(10) NOT NULL,
 `tgl_penjualan` date NOT NULL,
 `kode_konsumen` varchar(10) DEFAULT NULL,
 `total_jml_jual` int(11) NOT NULL,
 `total_penjualan` double NOT NULL,
 PRIMARY KEY (`kode_penjualan`),
 KEY `FK_kode_konsumen2` (`kode_konsumen`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `penjualan_spare`
 ADD CONSTRAINT `FK_kode_konsumen1` FOREIGN KEY
(`kode_konsumen`) REFERENCES `konsumen` (`kode_konsumen`),
 ADD CONSTRAINT `FK_kode_konsumen2` FOREIGN KEY
(`kode_konsumen`) REFERENCES `konsumen` (`kode_konsumen`);
```

k. Sintak untuk membuat tabel detail_penjualan

```
CREATE TABLE IF NOT EXISTS `detail_penjualan` (
```

```
  `kode_penjualan` varchar(10) NOT NULL,
  `kode_sparepart` varchar(10) NOT NULL,
  `harga_jual_spare` double NOT NULL,
  `jml_jual_spare` int(11) NOT NULL,
  `potongan_penjualan` bigint(20) NOT NULL,
  `subtotal` double NOT NULL,
  KEY `FK_kode_penjualan1` (`kode_penjualan`),
  KEY `FK_kode_sparepart4` (`kode_sparepart`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `detail_penjualan`
  ADD CONSTRAINT `FK_kode_penjualan1` FOREIGN KEY (`kode_penjualan`) REFERENCES `penjualan_spare` (`kode_penjualan`),
  ADD CONSTRAINT `FK_kode_sparepart4` FOREIGN KEY (`kode_sparepart`) REFERENCES `sparepart` (`kode_sparepart`);
```

l. Sintak untuk membuat tabel *Service*

```
CREATE TABLE IF NOT EXISTS `service` (
  `kode_service` varchar(10) NOT NULL,
  `tgl_service` date NOT NULL,
  `kode_konsumen` varchar(10) NOT NULL,
  `kode_mekanik` varchar(10) NOT NULL,
  `keluhan` text NOT NULL,
```

```
  `total_service` bigint(20) NOT NULL,
  `status_service` varchar(20) NOT NULL,
  PRIMARY KEY (`kode_service`),
  KEY `FK_kode_konsumen3` (`kode_konsumen`),
  KEY `FK_kode_mekanik1` (`kode_mekanik`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `service`
  ADD CONSTRAINT `FK_kode_konsumen3` FOREIGN KEY
(`kode_konsumen`) REFERENCES `konsumen` (`kode_konsumen`),
  ADD CONSTRAINT `FK_kode_mekanik1` FOREIGN KEY
(`kode_mekanik`) REFERENCES `mekanik` (`kode_mekanik`);
```

m. Sintak untuk membuat tabel detail service

```
CREATE TABLE IF NOT EXISTS `detail_service` (
  `kode_service` varchar(10) NOT NULL,
  `kode_jasa_service` varchar(10) NOT NULL,
  `harga_service` varchar(10) NOT NULL,
  KEY `FK_kode_service1` (`kode_service`),
  KEY `FK_kode_jasa_service1` (`kode_jasa_service`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `detail_service`
```

```
 ADD CONSTRAINT `FK_kode_jasa_service1` FOREIGN KEY
(`kode_jasa_service`) REFERENCES `jasa_service`
(`kode_jasa_service`),
 ADD CONSTRAINT `FK_kode_service1` FOREIGN KEY
(`kode_service`) REFERENCES `service` (`kode_service`);
```

n. Sintak untuk membuat tabel jasa_service

```
CREATE TABLE IF NOT EXISTS `jasa_service` (
 `kode_jasa_service` varchar(10) NOT NULL,
 `nama_service` varchar(50) NOT NULL,
 `kategori_service` varchar(16) NOT NULL,
 `harga_service` bigint(20) NOT NULL,
 PRIMARY KEY (`kode_jasa_service`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

o. Sintak untuk membuat tabel type kendaraan

```
CREATE TABLE `type_kendaraan` (
 `kode_type` varchar(10) NOT NULL,
 `kode_merk` varchar(10) NOT NULL,
 `nama_type` varchar(50) NOT NULL,
 PRIMARY KEY (`kode_type`),
 KEY `FK_kode_merk1` (`kode_merk`)
) ENGINE=InnoDB;


ALTER TABLE `type_kendaraan`
```

```
ADD CONSTRAINT `FK_kode_merk1` FOREIGN KEY (`kode_merk`) REFERENCES `merk_kendaraan` (`kode_merk`);
```

p. Sintak untuk membuat tabel merk kendaraan

```
CREATE TABLE IF NOT EXISTS `merk_kendaraan` (
  `kode_merk` varchar(10) NOT NULL,
  `nama_merk` varchar(20) NOT NULL,
  PRIMARY KEY (`kode_merk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

q. Sintak untuk membuat tabel pembayaran

```
CREATE TABLE IF NOT EXISTS `pembayaran` (
  `kode_pembayaran` int(10) NOT NULL AUTO_INCREMENT,
  `tgl_pembayaran` date NOT NULL,
  `kode_service` varchar(10) DEFAULT NULL,
  `kode_penjualan` varchar(10) DEFAULT NULL,
  `total_bayar` double NOT NULL,
  `jumlah_bayar` double NOT NULL,
  `jml_kembali` double NOT NULL,
  `type_transaksi` int(11) NOT NULL,
  PRIMARY KEY (`kode_pembayaran`),
  KEY `FK_kode_service3` (`kode_service`),
  KEY `FK_kode_penjualan3` (`kode_penjualan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

```
ALTER TABLE `pembayaran`
 ADD CONSTRAINT `FK_kode_penjualan3` FOREIGN KEY
(`kode_penjualan`) REFERENCES `penjualan_spare`
(`kode_penjualan`),
 ADD CONSTRAINT `FK_kode_service3` FOREIGN KEY
(`kode_service`) REFERENCES `service` (`kode_service`);
```

4.5.4 Implementasi antar muka

Dibawah ini merupakan Implementasi antar muka yang di usulkan untuk Sistem Informasi bengkel Kegan Putra *Sparepart*.

Tabel 4.20 Implementasi Menu Utama

Sub Menu	Deskripsi	Nama *File*
File	Untuk memangil sub menu tambah *user*, *login*, *logout*, keluar	MenuUtama.java
Master	Untuk memanggil sub menu *sparepart*, konsumen, jasa *service*, *type* kendaraan, merk kendaraan, *supplier*, mekanik	MenuUtama.java
Transaksi	Untuk memanggil sub menu *service*, penjualan, pembayaran, *purchase order*, pembelian	MenuUtama.java

Laporan	Untuk memanggil sub menu laporan penjualan, pembelian, *service*, *sparepart*	MenuUtama.java

Tabel 4.21 Implementasi Sub Menu *File*

Sub Menu	Deskripsi	Nama File
Tambah *User*	Menambah *user*	InternalFrameTambahUser.java
Login	Membuka hak akses	FormLogin.java
Logout	Menutup hak akses	
Keluar	Keluar dari program	Dispose();

Tabel 4.22 Implementasi Sub Menu Master

Sub Menu	Deskripsi	Nama File
Konsumen	Membuat *input* data konsumen	InternalFrameKonsumen.java
Supplier	Membuat *input* data *supplier*	InternalFrameSupplier.java
Sparepart	Membuat *input* data *Sparepart*	InternalFrameSparepart.java
Jasa *Service*	Membuat *input* data jasa *service*	InternalFrameJasaService.java
Mekanik	Membuat *input* data mekanik	InternalFrameMekanik.java
Type Kendaraan	Membuat *input* data type motor	InternalFrameTypeKendaraan.java
Merk Kendaraan	Membuat *input* data merk motor	InternalFrameMerkKendaraan.java

Tabel 4.23 Implementasi Sub Menu Transaksi

Sub Menu	Deskripsi	Nama File
Service	Melakukan transaksi *service* motor	InternalFrameService.java
Penjualan	Melakukan transaksi penjualan	InternalFramePenjualan.java
Pembayaran	Melakukan transaksi pembayaran *service*	InternalFramePembayaran.java
Purchase Order	Melakukan transaksi *purchase order*	InternalFramePurchaseOrder.java
Pembelian	Melakukan transaksi pembelian	InternalFramePembelian.java

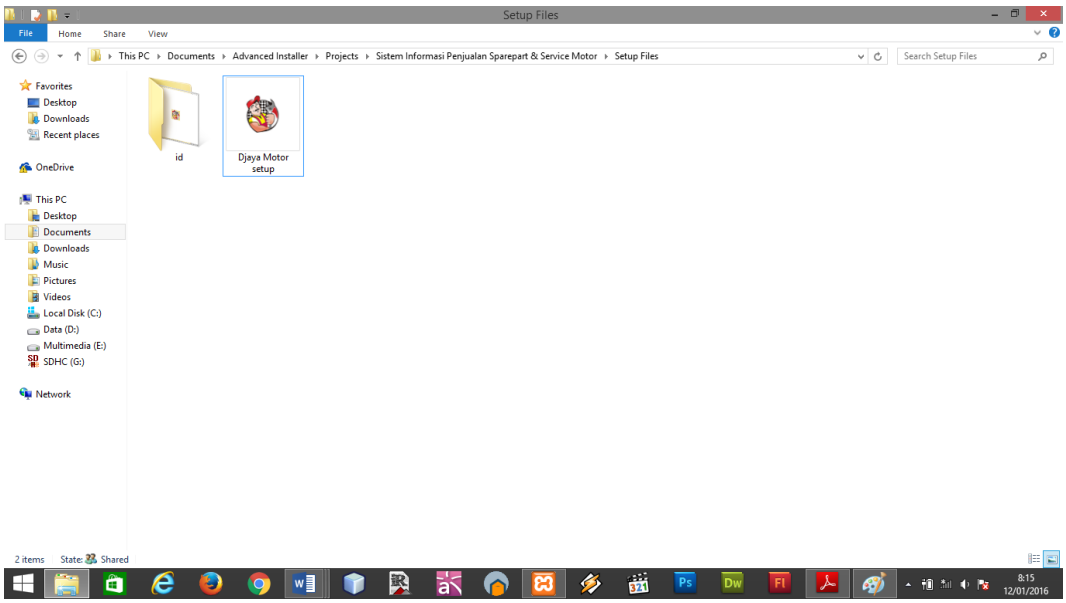
Tabel 4.24 Implementasi Sub Menu Laporan

Sub Menu	Deskripsi	Nama File
Laporan penjualan	Laporan data *sparepart* yang terjual	FormLaporanPenjualan.java
Laporan Pembelian	Laporan data pembelian *sparepart*	FormLaporanPembelian.java
Laporan *Service*	Laporan data *service* yang terjadi di bengkel	FormLaporanService.java
Laporan Data Stok *Sparepart*	Laporan data stok *sparepart*	FormLaporanSparepart.java

4.5.5 Implementasi instalasi program

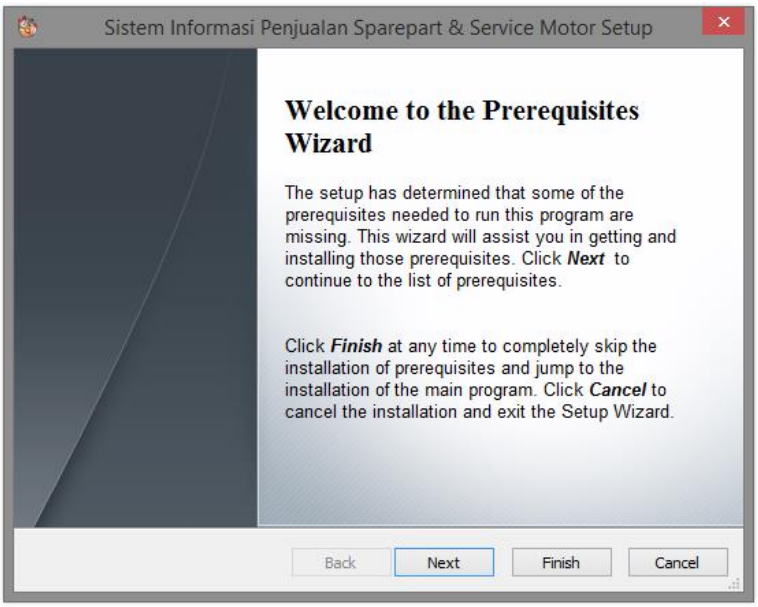
Dibawah ini adalah langkah-langkah untuk instalasi program atau aplikasi dari Sistem Informasi Penjualan dan Persediaan *Sparepart* dan Service Motor.

1. Buka folder Direktori penyimpanan dan klik dua kali Djaya Motor Setup.exe.



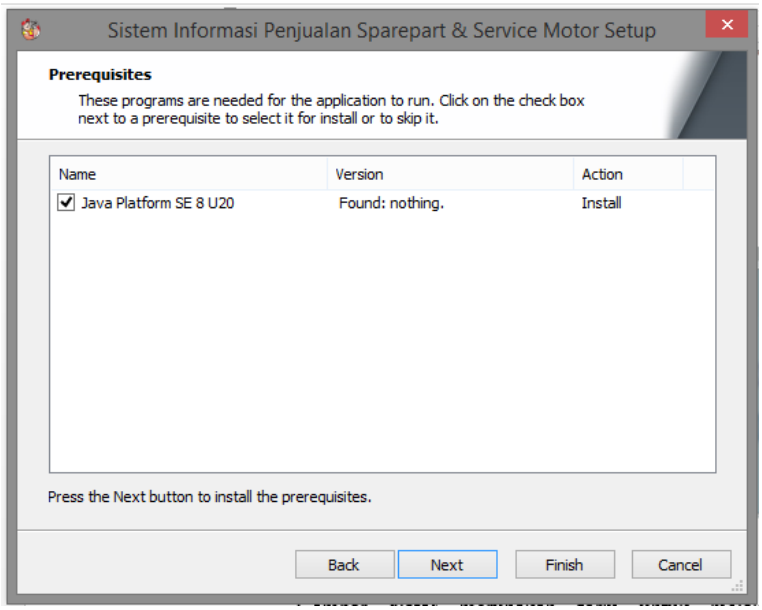
Gambar 4.55 Instalasi step 1

2. Setelah muncul lalu klik *next.*



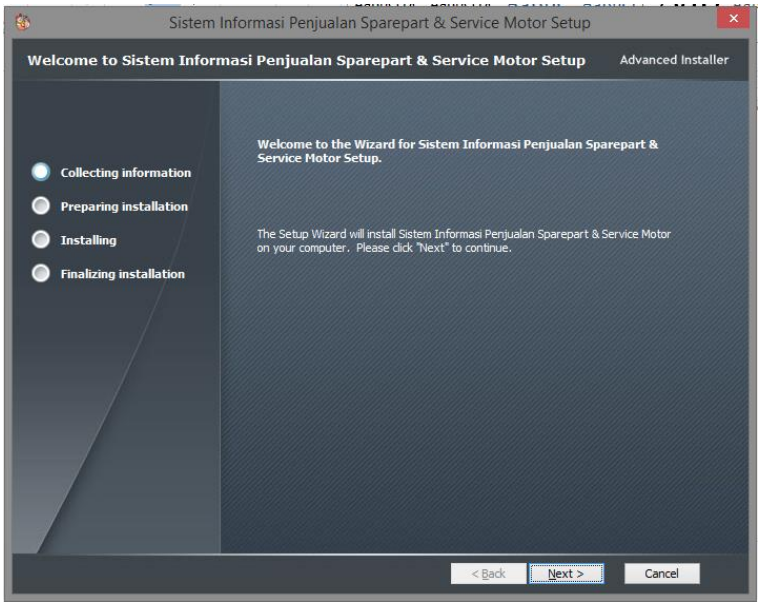
Gambar 4.56 Instalasi step 2

3. Jika belum terisntal jdk centang pada *checkbox*, jika sudah jangan di centang dan klik *finish*



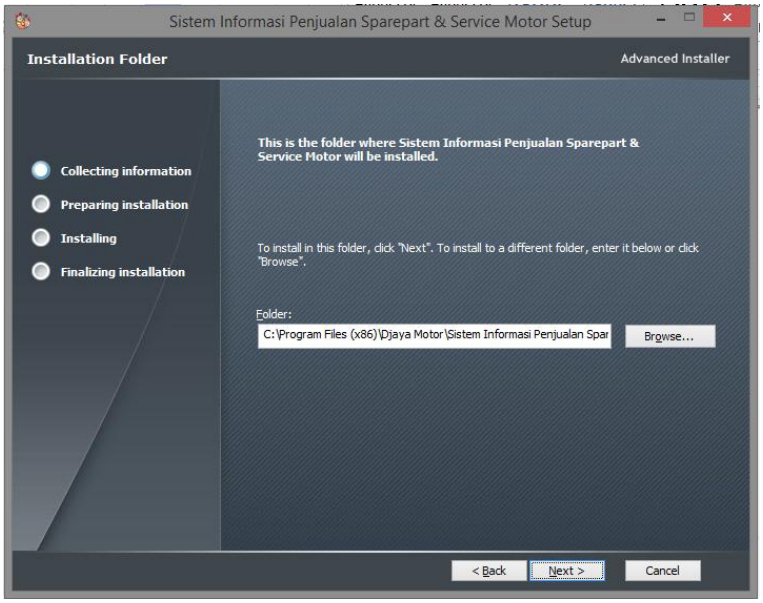
Gambar 4.57 Instalasi step 3

4. Maka akan menuju tahap Collecting information klik next



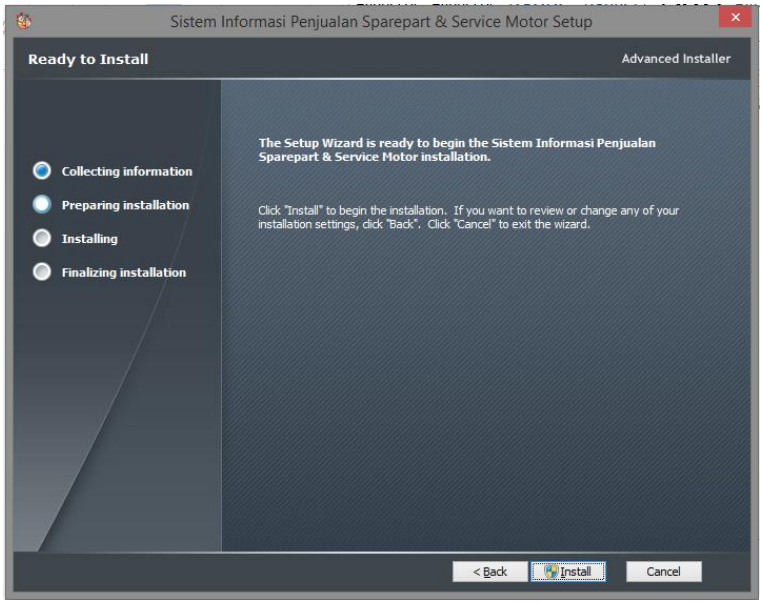
Gambar 4.58 Instalasi step 4

5. Pilih direktori penyimpanan data instalasi, lalu klik next



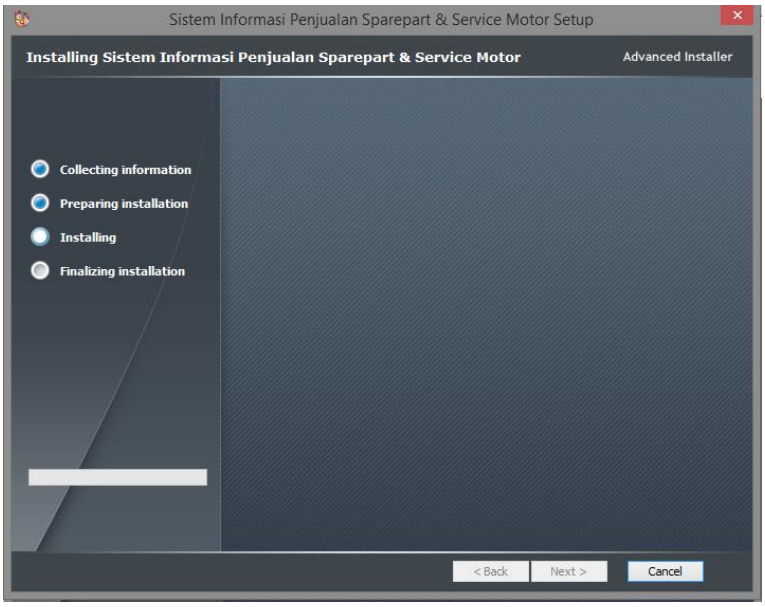
Gambar 4.59 Instalasi step 5

6. Lalu akan muncul pada tahan instalasi, klik instal



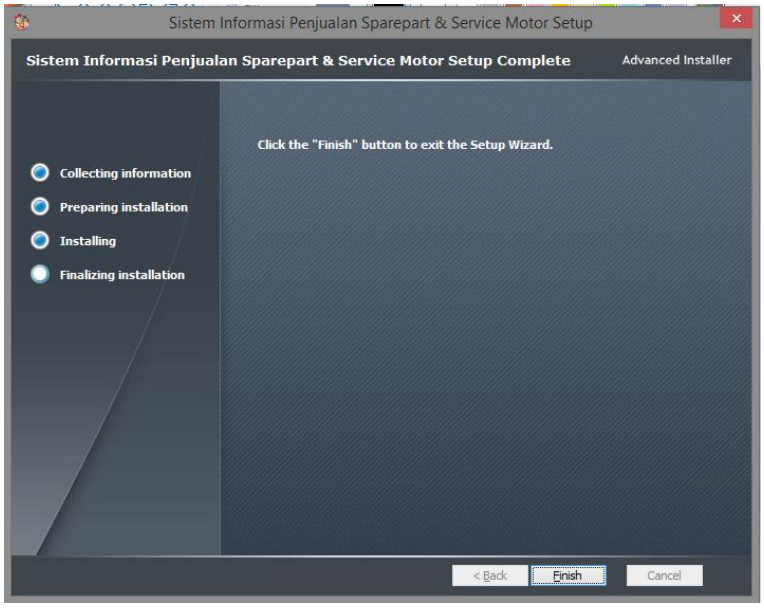
Gambar 4.60 Instalasi step 6

7. Pada tahapan ini sedang proses intalasi, tunggu sampai intalasi selesai



Gambar 4.61 Instalasi step 7

8. Jika sudah pada tahapan ini proses intalasi sudah selesai, dan klik finish



Gambar 4.62 Instalasi step 8

4.5.6 Penggunaan program

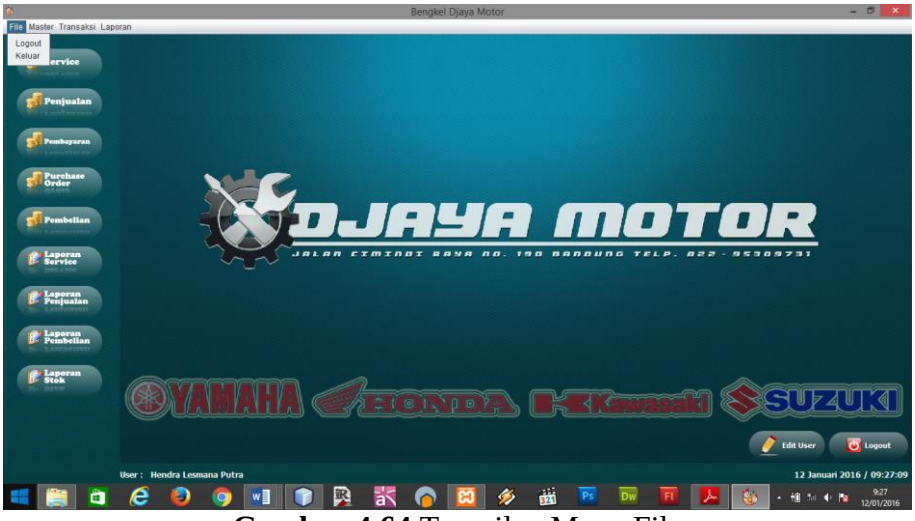
1. Tampilan awal



Gambar 4.63 Tampilan Login

Gambar diatas merupakan form untuk melakukan login dengan memasukan username dan password yang telah di buat.

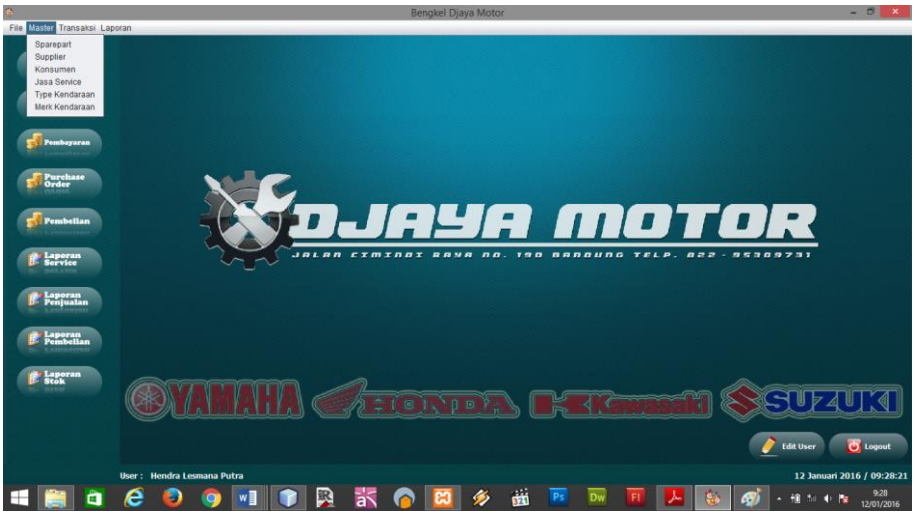
2. Tampilan menu file



Gambar 4.64 Tampilan Menu File

Gambar diatas merupakan isi dari menu file yang berisi beberapa sub menu logout dan keluar

3. Tampilan menu master



Gambar 4.65 Tampilan Menu Master

Gambar diatas merupakan isi dari menu master yang berisi beberapa sub menu diantaranya adalah konsumen, supplier, sparepart, merk kendaraan, type kendaraan, dan mekanik, jasa service yang hanya dimiliki oleh type user owner.

4. Tampilan menu transaksi



Gambar 4.66 Tampilan Menu Transaksi

Gambar diatas merukan menu transaksi yang memiliki beberapa submenu adalah sebagai berikut purchaseorder atau pemesanan, pembelian, penjulan, service dan pembayaran.

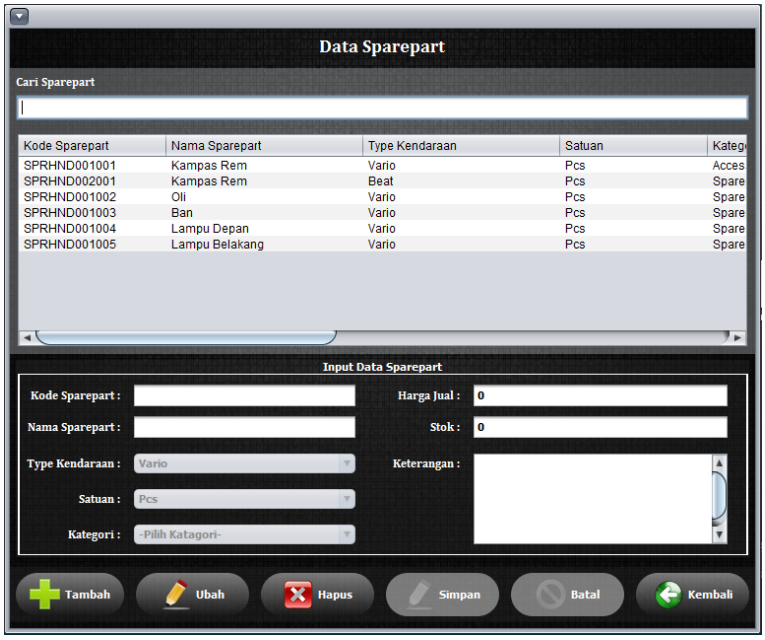
5. Tampilan menu laporan



Gambar 4.67 Tampilan Menu Laporan

Gambar diatas merupakan menu laporan yang didalamnya mempunyai beberapa submenu beberapa submenu itu adalah laporan pembelian, laporan penjualan, laporan service/pelayanan perbaikan motor, dan laporan stok sparepart.

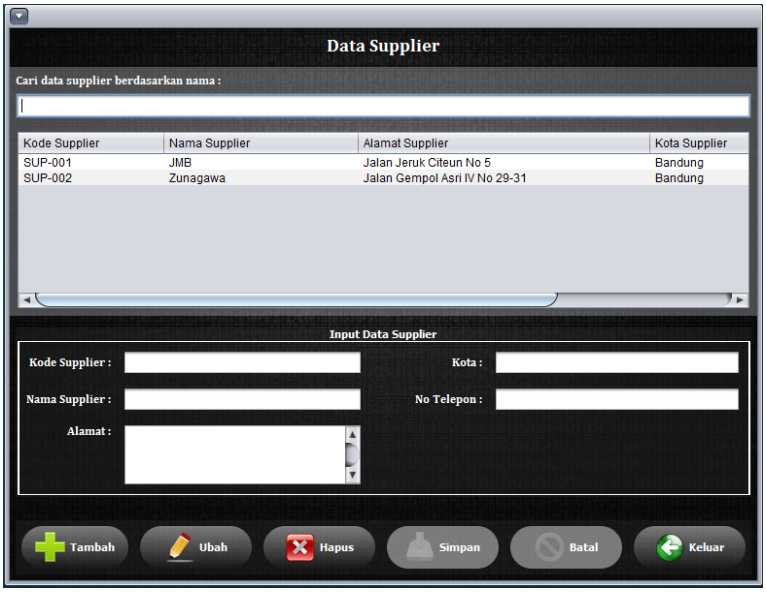
6. Tampilan sub menu sparepart



Gambar 4.68 Tampilan Menu Sparepart

Gambar diatas merupakan data *sparepart* yang berisi tentang data-data *sparepart* dan untuk menambahakan data *sparepart* baru atau mengedit data *sparepart* yang ada ataupun menghapus data *sparepart* yang ada.

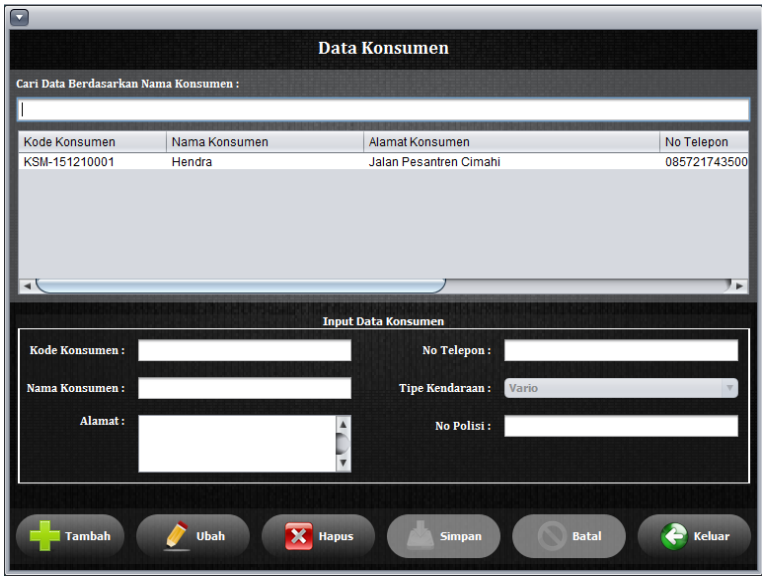
7. Tampilan sub menu supplier



Gambar 4.69 Tampilan Menu Supplier

Gambar diatas merupakan data *supplier* yang berisi semua data *supplier*, dan juga untuk menambahakan data *supplier*, mengedit ataupun menghapus data *supplier* yang ada.

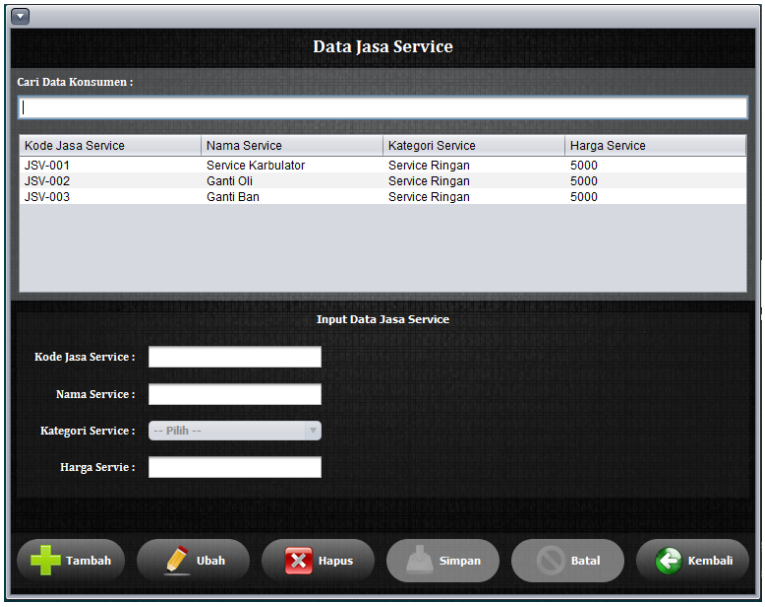
8. Tampilan sub menu konsumen



Gambar 4.70 Tampilan Menu Konsumen

Gambar diatas adalah data konsumen yang berisi data-data konsumen, menambahkan konsumen baru, mngubah konsumen yang ada tau menghapus konsumen.

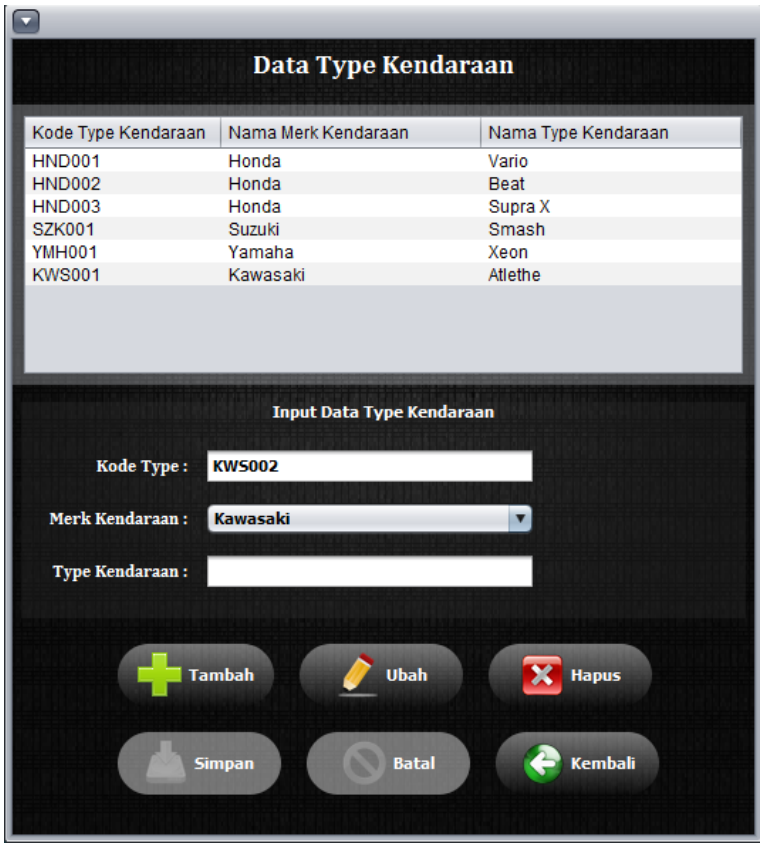
9. Tampilan sub menu jasa service



Gambar 4.71 Tampilan Menu Jasa Service

Gambar diatas adalah data jasa *service* yang berisi beberapa jenis *service* yang di sajikan oleh pihak bengkel dan juga menyertakan harga tiap jenis *type* jasa *service*, *form* ini hanya bisa diakses oleh *type user owner*.

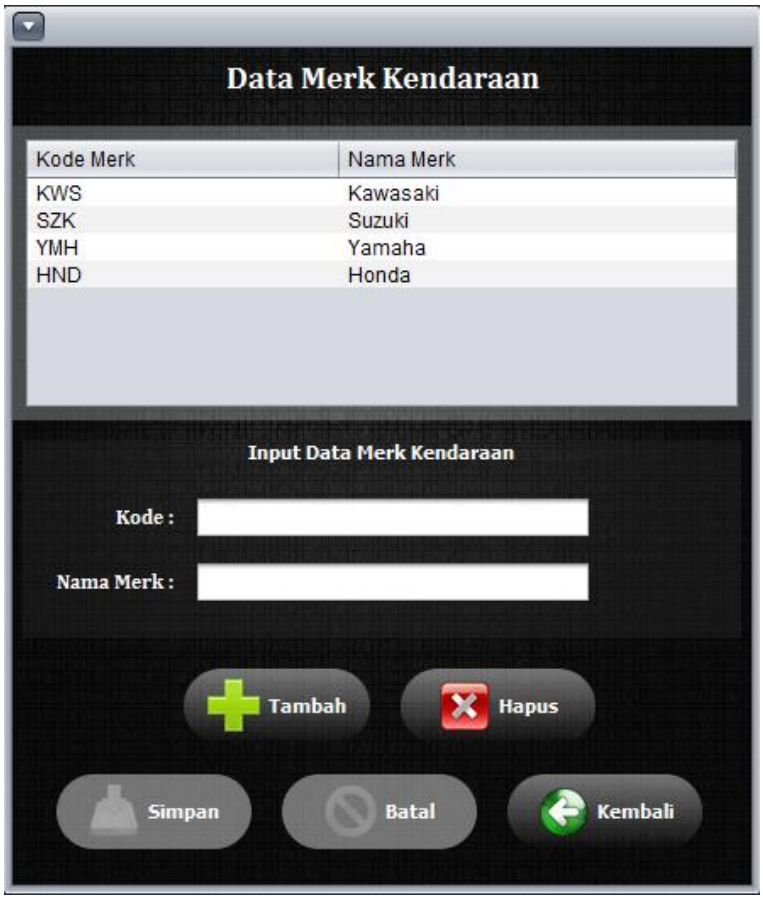
10. Tampilan sub menu type kendaraan



Gambar 4.72 Tampilan Menu Type Kendaraan

Gambar diatas merupakan data *type* motor yang berisi beberapa data *type* motor dan juga untuk menambahkan *type* motor yang baru, mengubah data *type* motor, atau menghapus *type* motor yang ada.

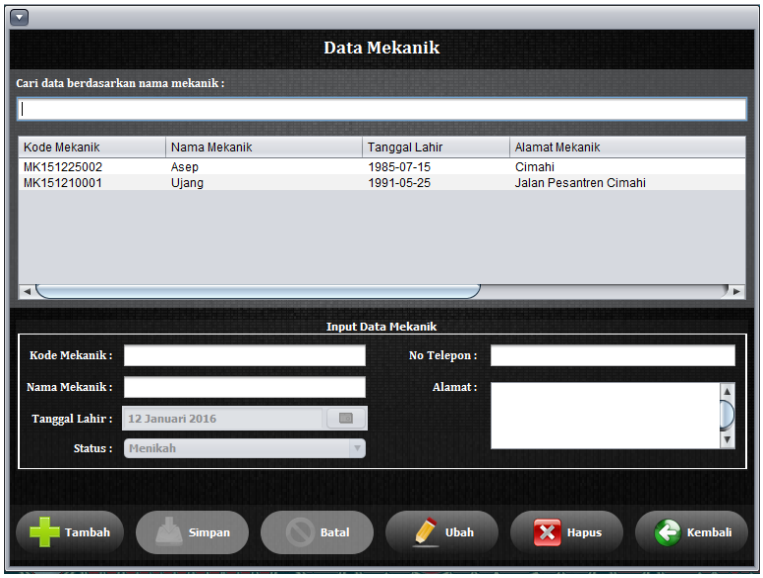
11. Tampilan sub menu merk kendaraan



Gambar 4.73 Tampilan Menu Merk Kendaraan

Gambar diatas merupakan data merek motor yang berisi beberapa dat merek motor dan untuk memasukan data merek motor yang baru ataupun menghapus data yang ada.

12. Tampilan sub menu mekanik



Gambar 4.74 Tampilan Menu Mekanik

Gambar diatas adalah form mekanik yang berisi data-data mekanik *form* ini hanya bisa diakses oleh *type user owner* karena *owner* yang berhak memecat atau menambahkan mekanik.

13. Tampilan sub menu service



Gambar 4.75 Tampilan Menu Service

Gambar diatas adalah *form service* atau pelayanan perbaiakn motor, berisi beberapa data tentang *service*, bila diperlukan *sparepart* bisa menambahkan *sparepart* atau melakukan service saja, dan akan menampilkan surat perintah kerja untuk mekanik jika input *service* selesai dilakukan bisa juga untuk mengubah data *service* dengan status menunggu atau sedang dilakukan proses perbaikan.

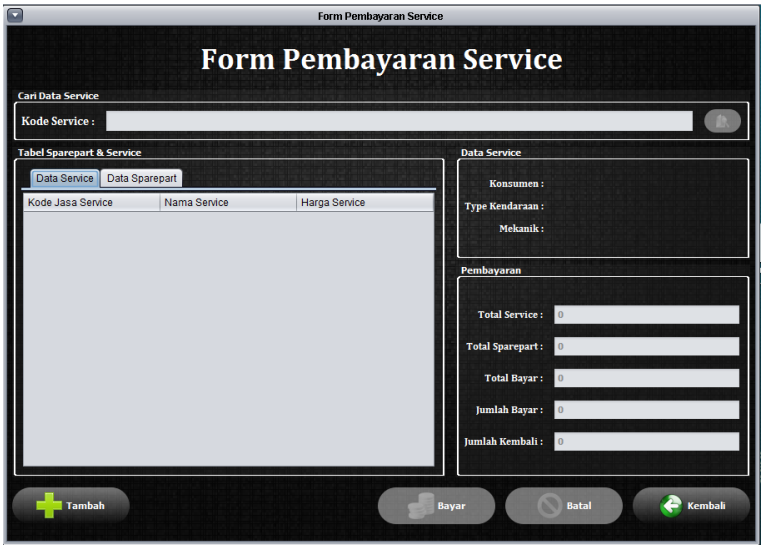
14. Tampilan sub menu penjualan



Gambar 4.76 Tampilan Menu Penjualan

Gambar diatas adalah *form* penjualan yaitu untuk melakukan transaksi penjualan *sparepart*.

15. Tampilan sub menu pembayaran



Gambar 4.77 Tampilan Menu Pembayaran

Gambar diatas adalah tampilan dari form pembayaran *service* yang berisi data-data *service* yang belum melakukan pembayaran.

16. Tampilan sub menu purchase order



Gambar 4.78 Tampilan Menu Purchase Order

Gambar diatas adalah *form purchaseorder* merupakan *form* untuk menambahkan pesanan beberapa *sparepart*.

17. Tampilan sub menu pembelian



Gambar 4.79 Tampilan Menu Pembelian

Gambar diatas adalah *form* pembelian dimana untuk membuat transaksi pembelian dengan mengload data po yang telah dibut sebelumnya, disana bisa untuk menambahkan harga beli dan otomatis akan muncul harga jual yaitu 15% dari harga beli, dan juga merubah jumlah *sparepart* jika barang tidak sesuai dengan po.

18. Tampilan sub menu laporan penjualan



Gambar 4.80 Tampilan Menu Laporan Penjualan

Diatas adalah *form* laporan penjualan berisi laporan penjualan dan bisa di pilih berdasarkan hari, bulan, tahun atau hanya beberapa priode tertentu.

19. Tampilan sub menu laporan pembelian



Gambar 4.81 Tampilan Menu Laporan Pembelian

Diatas adalah gambar laporan pembelian dengan katagori laporan pembelian perhari, diatas hanya menampilkan transaksi pembelian yang terjadi pada hari tertentu.

20. Tampil sub menu laporan service



Gambar 4.82 Tampilan Menu Laporan Service

Diatas adalah *form* laporan pembelian berisi laporan pembelian dan bisa di pilih berdasarkan hari, bulan, tahun atau hanya beberapa priode tertentu.

21. Tampil sub menu laporan stok



Gambar 4.83 Tampilan Menu Laporan Stok

Diatas adalah gambar *form* laporan stok k sparepart yang berisi semua data sparepart dengan stok, harga jual, harga beli.

Berikut tampilan laporan sparepart :