BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Perancangan sistem bisa diartikan sebagai pengembangan sistem baru dari sistem lama yang ada, dimana masalah-masalah yang terjadi pada sistem lama diharapkan sudah teratasi pada sistem yang baru serta 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.

4.1.1. Tujuan Perancangan Sistem

Tujuan perancangan sistem yang diusulkan yaitu untuk menghasilkan suatu rancangan sistem yang dapat membantu proses pengolahan data dan transaksi dalam menangani proses pembelian, penjualan , persediaan suku cadang, servis motor, pencatatan komisi mekanik dan laporan transaksi yang semuanya terintegrasi dengan *database* dan aplikasi tersebut dapat mencapai sasaran kebutuhan dan tujuan dari perusahaan yaitu kepuasan pelanggan.

4.1.2. Gambaran Umum Sistem Yang Diusulkan

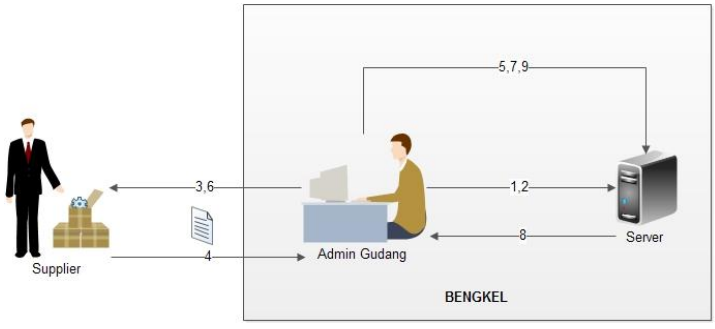
Gambaran umum tentang sistem yang diusulkan pada proses perancangan ini adalah suatu sistem yang merubah seluruh proses bisnis yang manual menjadi terkomputerisasi. Proses bisnis yang dimaksud adalah pengolahan data pelanggan, transaksi penjualan-pembelian suku cadang, pelayanan servis, pencatatan komisi

mekanik 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.

4.1.3. 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).

Proses bisnis transaksi pembelian suku cadang yang diusulkan adalah sebagai berikut :

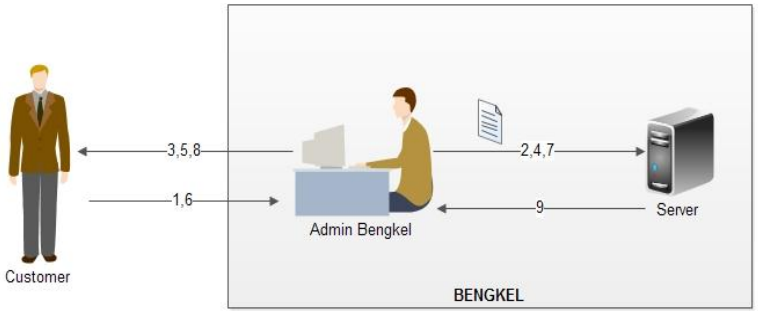


Gambar 4.1 Proses bisnis transaksi pembelian suku cadang yang diusulkan

Keterangan :

1. Admin gudang melakukan login ke sistem.
2. Admin gudang membuat pemesanan pembelian (PO).
3. Admin gudang mencetak dan mengirimkan data pemesanan pembelian (PO) kepada *supplier.*
4. *Supplier* mengirimkan barang dan faktur pemesanan pembelian sesuai PO kepada admin gudang.
5. Admin gudang menerima barang dan faktur pemesanan pembelian dan melakukan pengecekan dan penyesuain barang.
6. Admin gudang melakukan pembayaran secara tunai kepada *supplier.*
7. Admin gudang melakukan input data barang ke dalam transaksi pembelian sesuai dengan No. Pesanan.
8. Sistem menampilkan seluruh data transaksi dari pesanan pembelian.
9. Admin gudang melakukan penyesuaian data transaksi.

Proses bisnis transaksi penjualan suku cadang yang diusulkan adalah sebagai berikut :

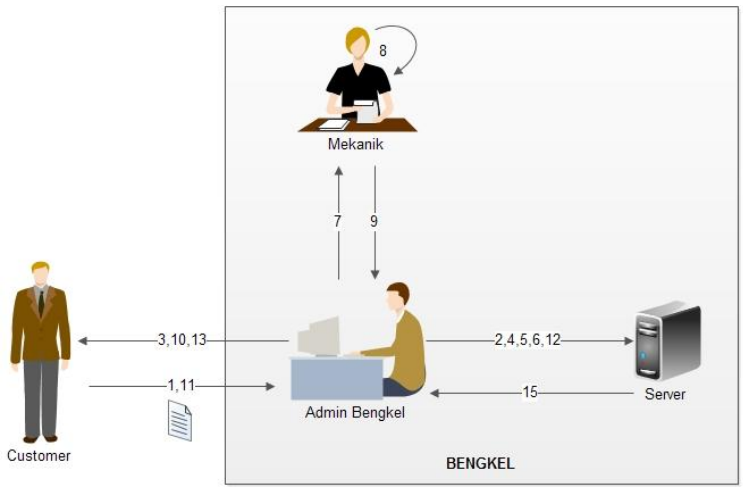


Gambar 4.2 Proses bisnis transaksi penjualan suku cadang yang diusulkan

Keterangan :

1. Pelanggan datang ke bengkel untuk melakukan pembelian kepada admin bengkel.
2. Admin bengkel login ke sistem.
3. Admin bengkel menanyakan kebutuhan suku cadang kepada pelanggan.
4. Admin bengkel memeriksa ketersediaan dan harga suku cadang pada *database*.
5. Admin bengkel memberitahu harga kepada pelanggan.
6. Pelanggan melakukan pembayaran tunai sesuai barang yang dibeli.
7. Admin bengkel membuat dan mencetak faktur penjualan
8. Admin bengkel menyerahkan faktur penjualan dan suku cadang yang dibeli kepada pelanggan.
9. Sistem memberikan respon sesuai data yang diinginkan.

Proses bisnis servis motor yang diusulkan adalah sebagai berikut :



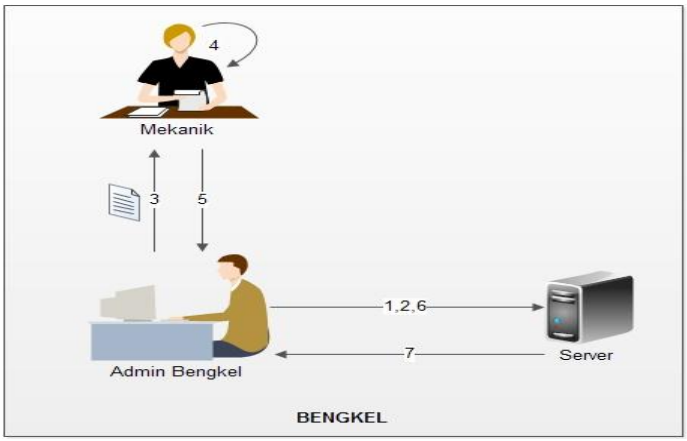
Gambar 4.3 Proses bisnis servis yang diusulkan

Keterangan :

1. Pelanggan datang ke bengkel mendaftar servis kepada admin bengkel dengan menyerahkan STNK untuk input data.
2. Admin bengkel melakukan *login* ke sistem.
3. Admin bengkel menanyakan kebutuhan suku cadang dan keluhan pelanggan
4. Admin bengkel melakukan input data transaksi servis.
5. Admin bengkel melakukan pemeriksaan data pelanggan dan kendaraan di sistem. Jika di *database* tidak ditemukan, maka Admin bengkel melakukan input data dan kendaraan pelanggan pada *database.*
6. Admin bengkel membuat dan mencetak surat perintah kerja untuk mekanik.
7. Admin bengkel menyerahkan surat perintah kerja kepada mekanik.
8. Mekanik melakukan pemeriksaan motor sesuai surat perintah kerja.
9. Mekanik melakukan konfirmasi ke admin bengkel servis sudah selesai.

10. Admin bengkel melakukan konfirmasi kepada pelanggan servis sudah selesai.
11. Pelanggan melakukan pembayaran tunai kepada admin bengkel.
12. Admin bengkel mencetak faktur penjualan (Biaya servis dan harga suku cadang).
13. Admin bengkel menyerahkan faktur penjualan dan jasa servis.
14. Sistem memberikan respon sesuai data yang diinginkan.

Proses bisnis pencatatan komisi mekanik yang diusulkan adalah sebagai berikut :



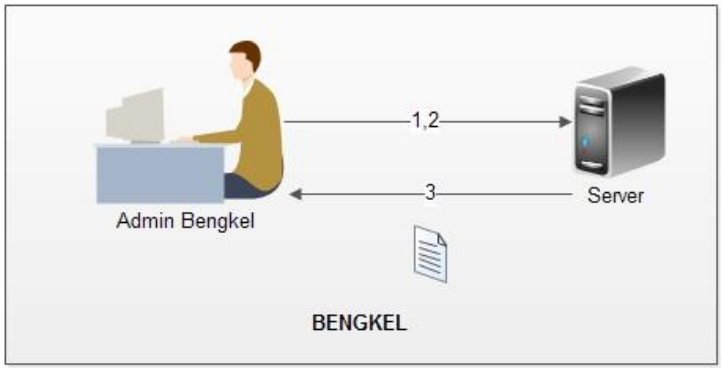
Gambar 4.4 Proses bisnis pencatatan komisi mekanik yang diusulkan

Keterangan :

1. Admin bengkel *login* ke sistem.
2. Admin bengkel melakukan input data mekanik yang tersedia.
3. Admin bengkel mencetak dan menyerahkan surat perintah kerja kepada mekanik.
4. Mekanik menerima surat perintah kerja dari admin bengkel dan melakukan servis sesuai dengan surat perintah kerja.

5. Mekanik konfirmasi perkerjaan selesai kepada admin bengkel.
6. Admin bengkel melakukan penyesuaian dan *update* data transaksi.
7. Sistem memberikan respon sesuai data yang diinginkan.

Proses bisnis pelaporan yang diusulkan adalah sebagai berikut :



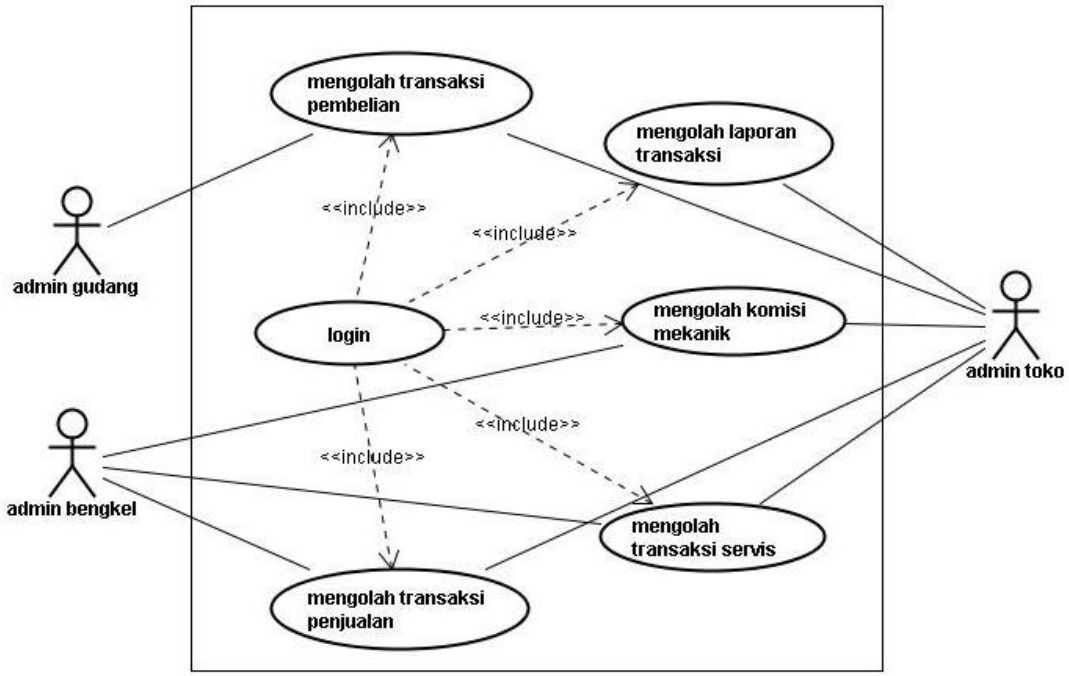
Gambar 4.5 Proses bisnis pelaporan yang diusulkan

Keterangan :

1. Kepala toko melakukan *login* ke sistem.
2. Kepala toko melihat laporan dengan memasukan data periode laporan dan jenis laporan sesuai kebutuhan.
3. Sistem secara terkomputerisasi akan menampikan laporan sesuai dengan kondisi yang diinginkan.

4.1.3.1. Use Case Diagram

Berikut adalah *use case* yang diusulkan pada perancangan sistem informasi pelayanan bengkel ini :



Gambar 4.6 *Use case* diagram yang diusulkan

4.1.3.2. Identifikasi *use case* dan Deskripsi yang diusulkan

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

Table 4.1 Identifikasi *use case* sistem yang diusulkan

No	*Use case*	Deskripsi
UC-01	Login	Fungsionalitas untuk menentukan hak akses user pada saat login ke sistem
UC-02	Mengolah transaksi pembelian	Fungsionalitas untuk melakukan penambahan data transaksi pembelian
UC-03	Mengolah transaksi	Fungsionalitas untuk mengolah data

	penjualan	transaksi penjualan.
UC-04	Mengolah transaksi servis	Fungsionalitas untuk mengolah data transaksi servis
UC-05	Mengolah komisi mekanik	Fungsionalitas untuk mengolah data pencatatan komisi mekanik
UC-06	Mengolah laporan transaksi	Fungsionalitas untuk melakukan pemeriksaan dan pencetakan laporan

4.1.3.3.Identifikasi aktor dan Deskripsi yang diusulkan

Aktor yang berperan dalam menjalankan sistem adalah sebagai berikut :

Table 4.2 Identifikasi aktor sistem yang diusulkan

No	Aktor	Deskripsi
P-01	Admin bengkel	Pihak yang berperan mengatur proses administrasi, transaksi, servis dan pencatatan komisi mekanik di bengkel
P-02	Admin gudang	Pihak yang berfungsi mengatur proses pemesanan pembelian dan persediaan suku cadang
P-03	Kepala toko	Pihak yang berfungsi memeriksa semua laporan dan bertanggung jawab penuh dengan seluruh kegiatan proses bisnis di bengkel
P-04	Mekanik	Pihak yang bertanggung jawab dengan kegiatan transaksi servis di bengkel
P-05	Supplier	Pihak yang melakukan proses pengiriman suku cadang

4.1.3.4. Skenario *Use Case* yang diusulkan

Adapun tahapan dari scenario *use case* pada sistem yang diusulkan adalah sebagai berikut :

1. Skenario *use case Login*

Table 4.3 Skenario *use case* login

Identifikasi	
Judul	Login
No. Use case	UC-01
Deskripsi use case	Aktor login 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
Scenario utama	
Aksi-aktor	**Reaksi sistem**
1. Admin toko, admin bengkel, admin gudang mengisi *field username* dan *password* pada *form* login.	
2. Admin toko, admin bengkel, admin gudang menekan tombol login.	3. Sistem memeriksa inputan berdasarkan *username* dan *password* pada *database.*
Kondisi akhir scenario utama	4. Sistem menampilkan halaman utama berdasarkan hak akses dari user.
Skenario Alternatif – 1	
Aksi-aktor	**Reaksi sistem**
1. Admin toko, admin bengkel, admin gudang menekan tombol login tanpa mengisi *field-field* pada *form.*	2. *Form* login menampilkan pesan kesalahan yang menyatakan *username* atau *password* harus diisi.
	3. *Form* utama tidak ditampilkan
Kondisi akhir scenario alternative -1	Aplikasi menampilkan *form login.*
Skenario Alternatif – 2	
Aksi-aktor	**Reaksi sistem**
1. Admin toko, admin bengkel, admin gudang menekan tombol login dengan mengisi *field username* atau *password* yang salah pada *form.*	2. *Form* login menampilkan pesan kesalahan yang menyatakan *username* atau *password* salah.
	3. *Form* utama tidak ditampilkan
Kondisi akhir scenario alternative -2	Aplikasi menampilkan *form login.*

2. Skenario *use case* mengolah transaksi pembelian

Table 4.4 Skenario *use case* mengolah transaksi pembelian

Identifikasi	
Judul	Mengolah transaksi pembelian
No. Use case	UC-02
Deskripsi use case	Aktor menambah data transaksi pembelian
Aktor	Admin gudang
Trigger	Aktor memilih menu transaksi pembelian
Kondisi awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data transaksi pembelian

Scenario utama	
Aksi-aktor	**Reaksi sistem**
1. Admin gudang mengisi *field* pada *form* transaksi pembelian.	
2. Admin gudang menyimpan data transaksi pembelian	3. Sistem memeriksa apakah *field-field* pada *form* pengisian data telah diisi dan sesuai
	4. Sistem menampilkan respon sukses melakukan transaksi pembelian.
	5. Sistem menambahkan data transaksi penjualan yang baru pada *database.*
	6. Sistem memperbaharuai data suku cadang sesuai dengan data yang di input.
7. Admin gudang menekan tombol cetak	8. Sistem menampikan halaman *preview* data transaksi pembelian yang akan di cetak.
Kondisi akhir scenario utama	Data transaksi pembelian bertambah dan data suku cadang diperbaharuai pada *database* sesuai data yang di input.

Skenario Alternatif – 1	
Aksi-aktor	**Reaksi sistem**
1. Admin gudang menyimpan *form* transaksi pembelian dimana ada *field* yang tidak diisi atau salah dalam melakukan pengisian.	2. Sistem menampilkan pesan bahwa ada data *field* yang harus diisi atau *field* yang tidak sesuai.
	3. Data transaksi pembelian tidak disimpan dalam *database.*

Kondisi akhir scenario alternative -1	*Form* penambahan data transaksi pembelian ditampilkan.

3. Skenario *use case* mengolah transaksi penjualan

Table 4.5 Skenario *use case* mengolah transaksi penjualan

Identifikasi		
Judul	Mengolah transaksi penjualan	
No. Use case	UC-03	
Deskripsi use case	Aktor menambah data transaksi penjualan	
Aktor	Admin bengkel	
Trigger	Aktor memilih menu transaksi penjualan	
Kondisi awal	Aplikasi menampilkan sebuah *form* untuk menambahkan data transaksi penjualan	
Scenario utama		
Aksi-aktor		**Reaksi sistem**
1. Admin bengkel mengisi data penjualan yang baru		
2. Admin bengkel menyimpan data transaksi penjualan		3. Sistem memeriksa apakah *field-field* data yang diperlukan telah diisi dan sesuai.
		4. Data transaksi penjualan ditampilkan pada *form* dengan hasil kalkulasi.
		5. Sistem menambahkan data transaksi penjualan yang baru pada *database.*
		6. Sistem memperbaharui data suku cadang sesuai dengan data pada transaksi penjualan.
7. Admin bengkel menekan tombol cetak		8. Sistem menampikan halaman *preview* data transaksi penjualan yang akan di cetak.
Kondisi akhir scenario utama		Data transaksi pembelian bertambah dan data suku cadang diperbaharuai pada *database* sesuai data yang di input.
Skenario Alternatif – 1		
Aksi-aktor		**Reaksi sistem**
1. Admin bengkel menyimpan *form* transaksi penjualan dimana ada *field* yang tidak diisi.		2. Sistem menampilkan pesan bahwa ada data *field* yang harus diisi

	3. Data transaksi penjualan tidak disimpan dalam *database*
Kondisi akhir scenario alternative -1	*Form* penambahan data transaksi penjualan ditampilkan
Skenario Alternatif – 2	
Aksi-aktor	**Reaksi sistem**
1. Admin bengkel menekan tombol pending karena ada *customer* lain melakukan transaksi.	2. Sistem menyimpan data transaksi penjualan pending pada *database* sementara.
	3. Data transaksi penjualan tidak disimpan dalam *database.*
Kondisi akhir scenario alternative -2	Data transaksi penjualan tidak disimpan dalam *database.*

4. Skenario *use case* mengolah transaksi servis

Table 4.6 Skenario *use case* mengolah transaksi servis

Identifikasi	
Judul	Mengolah transaksi servis
No. Use case	UC-04
Deskripsi use case	Aktor menambah data transaksi servis
Aktor	Admin bengkel
Trigger	Aktor memilih menu transaksi servis
Kondisi awal	Aplikasi menampilkan sebuah *form* transaksi servis
Scenario utama	
Aksi-aktor	**Reaksi sistem**
1. Admin kasir mengisi data jenis servis yang baru	
2. Admin kasir menyimpan data transaksi penjualan	3. Sistem memeriksa apakah *field-field* pada *form* 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 suku cadang sesuai dengan data pada transaksi servis .
7. Admin kasir menekan tombol cetak	8. Sistem menampikan halaman *preview* data transaksi servis yang akan di

	cetak.
Kondisi akhir scenario utama	Data transaksi pembelian bertambah dan data suku cadang diperbaharuai pada *database* sesuai data yang di input.
Skenario Alternatif – 1	
Aksi-aktor	**Reaksi sistem**
1. Admin kasir menyimpan *form* transaksi servis dimana ada *field* yang tidak diisi.	2. Sistem menampilkan pesan bahwa ada data *field* yang harus diisi
	3. Data transaksi servis tidak disimpan dalam *database*
Kondisi akhir scenario alternative -1	*Form* penambahan data transaksi servis ditampilkan

5. Skenario *use case* pencatatan komisi mekanik

Table 4.7 Skenario *use case* pencatatan komisi mekanik

Identifikasi	
Judul	Pencatatan komisi mekanik
No. Use case	UC-05
Deskripsi use case	Aktor menambah data komisi mekanik
Aktor	Admin bengkel
Trigger	Aktor memilih menu mekanik
Kondisi awal	Aplikasi menampilkan sebuah *form* mekanik
Scenario utama	
Aksi-aktor	**Reaksi sistem**
1. Admin kasir mengisi data mekanik yang tersedia	
2. Admin kasir menyimpan data transaksi servis	3. Sistem memeriksa apakah *field-field* pada *form* data telah diisi dan sesuai.
	4. Data mekanik ditampilkan pada *form* dengan hasil kalkulasi.
	5. Sistem menambahkan data mekanik yang baru pada *database.*
Kondisi akhir scenario utama	Data mekanik diperbaharuai pada *database* sesuai data yang di input.
Skenario Alternatif – 1	
Aksi-aktor	**Reaksi sistem**
1. Admin kasir menyimpan *form* data mekanik dengan *field*	2. Sistem menampilkan pesan bahwa ada data *field* yang harus diisi

mekanik tidak diisi.	
	3. Data mekanik tidak disimpan dalam *database*
Kondisi akhir scenario alternative -1	*Form* penambahan data mekanik ditampilkan

6. Skenario *use case* mengolah pelaporan

Table 4.8 Skenario *use case* mengolah pelaporan

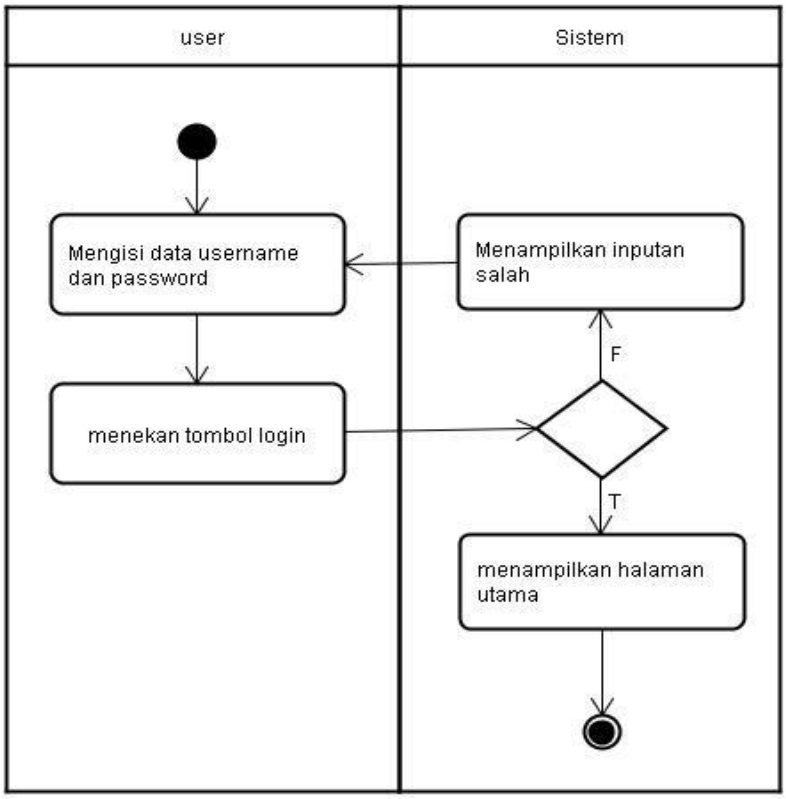
Identifikasi	
Judul	Mengolah laporan transaksi
No. Use case	UC-06
Deskripsi use case	Aktor mencetak laporan yang terjadi pada *range* waktu yang diinginkan
Aktor	Kepala toko
Trigger	Aktor memilih menu untuk mencetak laporan.
Kondisi awal	Aplikasi menampilkan sebuah *form* untuk menentukan periode dan jenis laporan yang diinginkan

Scenario utama	
Aksi-aktor	**Reaksi sistem**
1. Admin kasir mengisi periode waktu dan jenis laporan yang diinginkan	
2. Admin kasir menekan tombol cetak	3. Sistem mencetak laporan berdasarkan dengan jenis dan periode yang dipilih
Kondisi akhir scenario utama	Laporan dicetak
Skenario Alternatif – 1	
Aksi-aktor	**Reaksi sistem**
1. Admin kasir memilih periode yang tidak memiliki data penjualan sama sekali	
2. Admin kasir menekan tombol cetak	3. Sistem menampilkan pesan bahwa transaksi pada pada periode tersebut kosong
Kondisi akhir scenario alternative -1	Sistem menampilkan *form* untuk melakukan pengisian periode laporan yang diinginkan

4.1.3.5. Diagram *Activity*

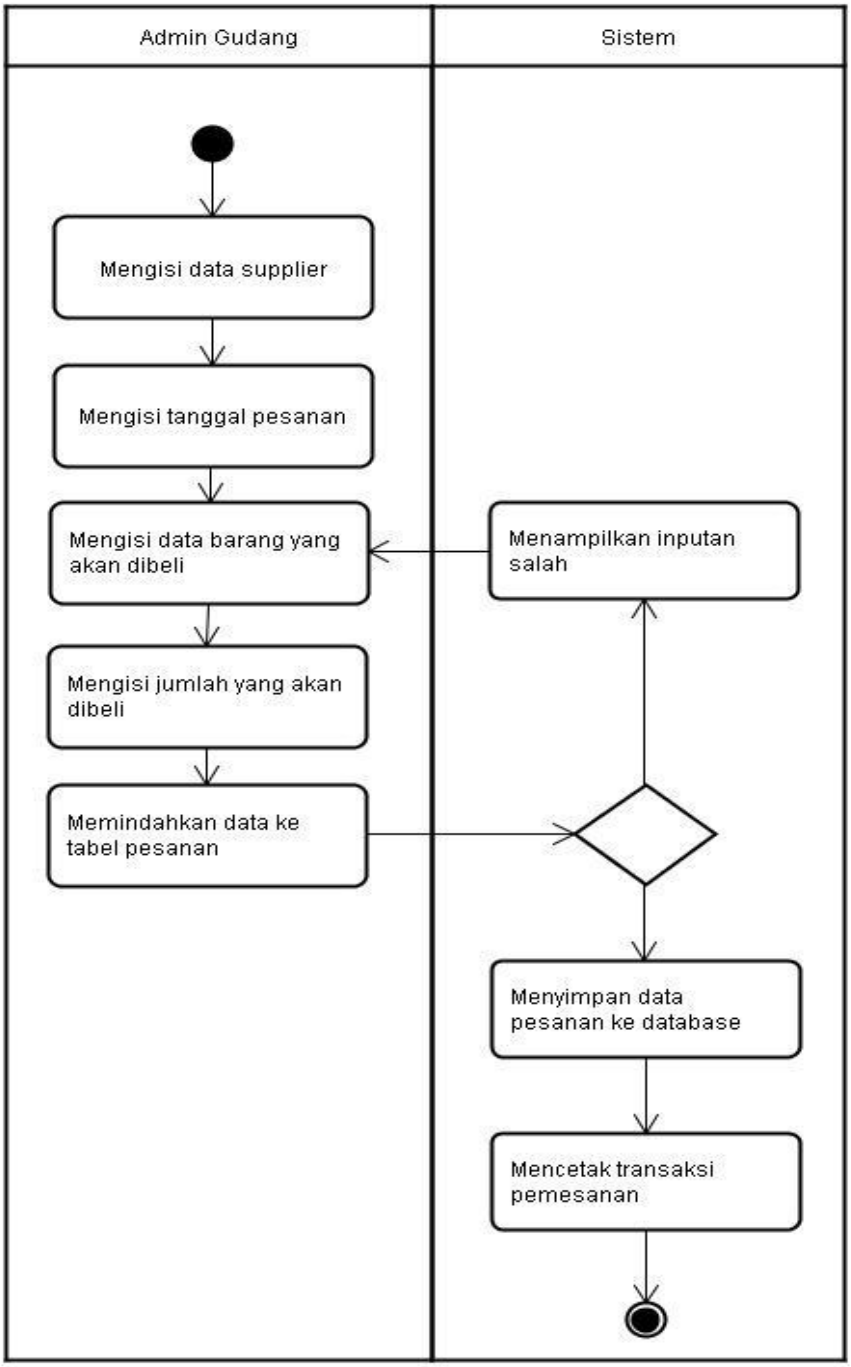
Diagram *activity* yang diusulkan untuk masing-masing bagian dari sistem yang diusulkan akan digambarkan seperti berikut ini :

1. *Activity diagram login user*



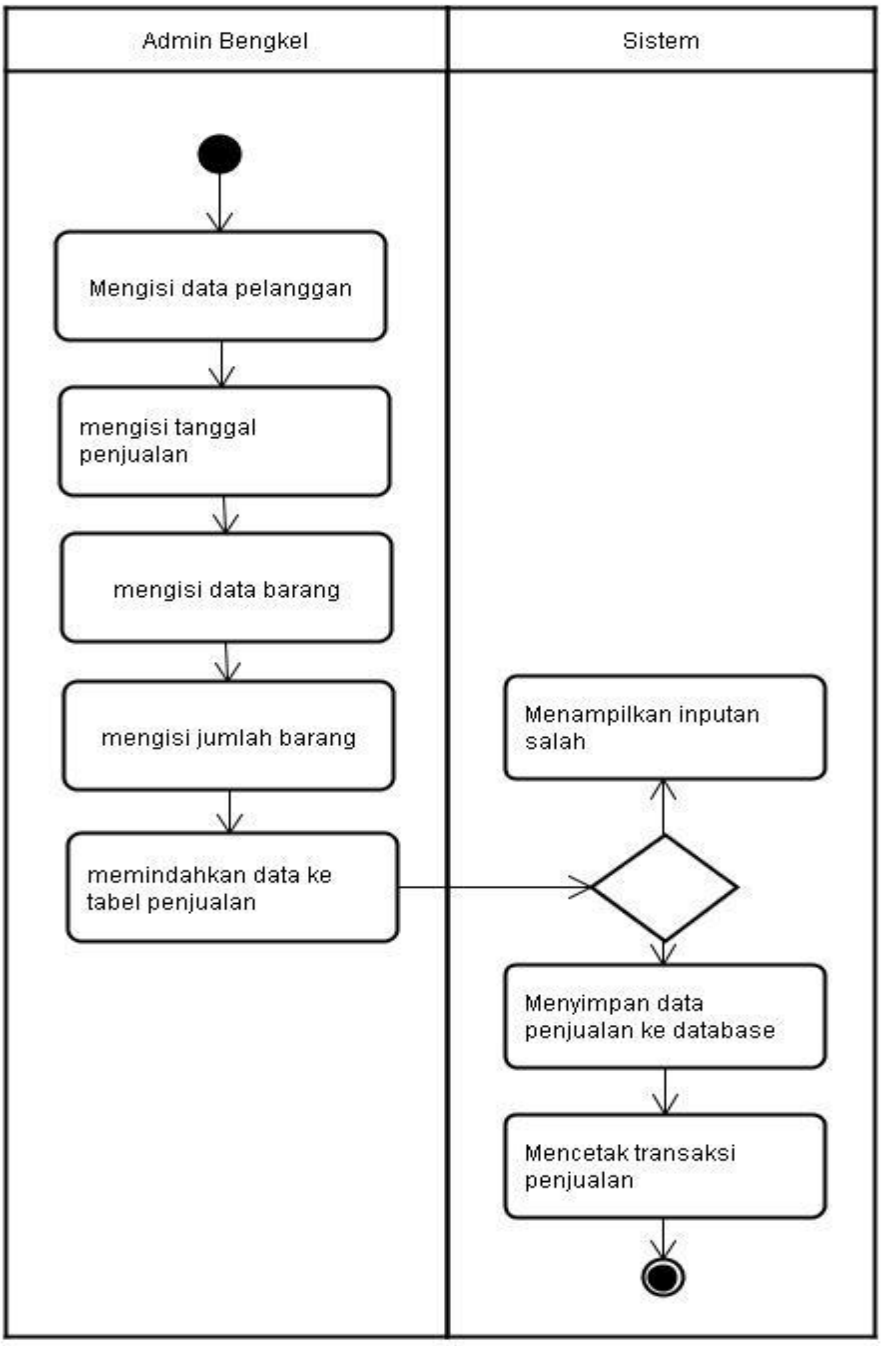
Gambar 4.7 *Activity diagram login user*

2. *Activity diagram* mengolah transaksi pembelian



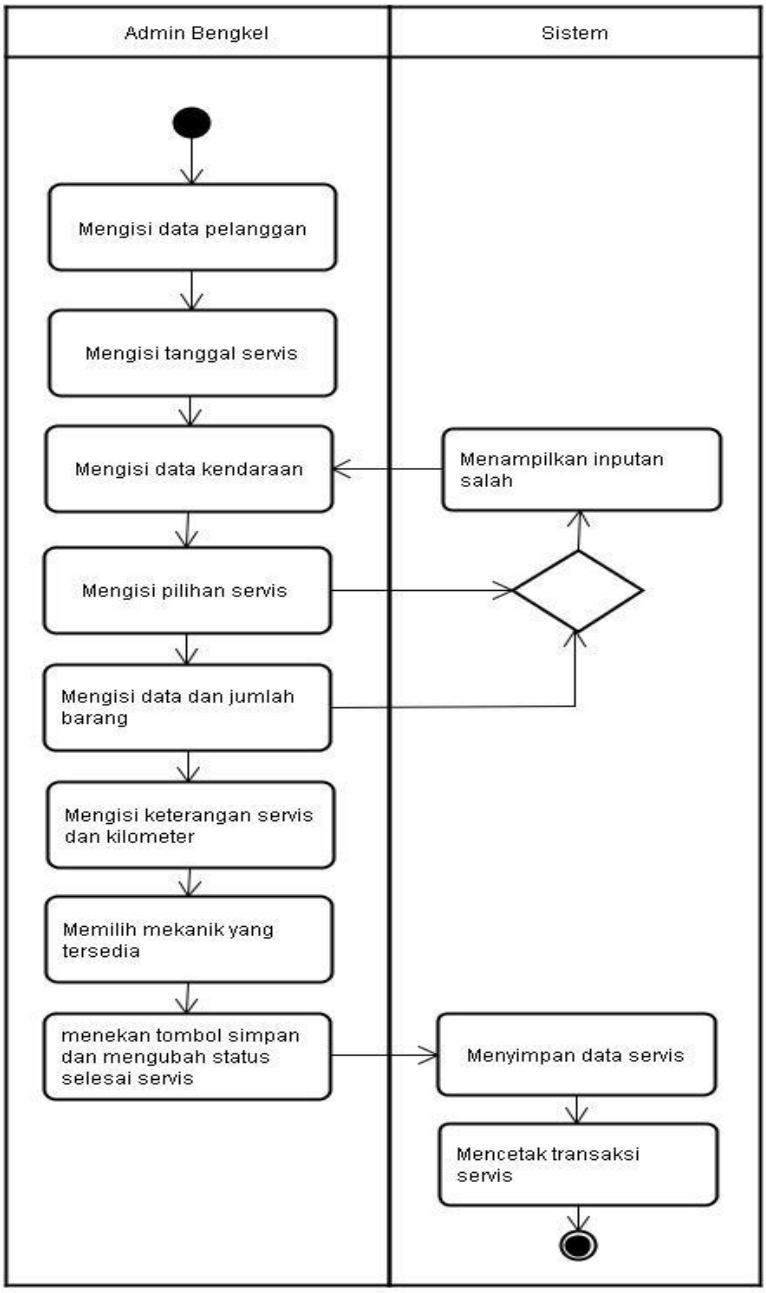
Gambar 4.8 *Activity diagram* mengolah transaksi pembelian

3. *Activity diagram* mengolah transaksi penjualan



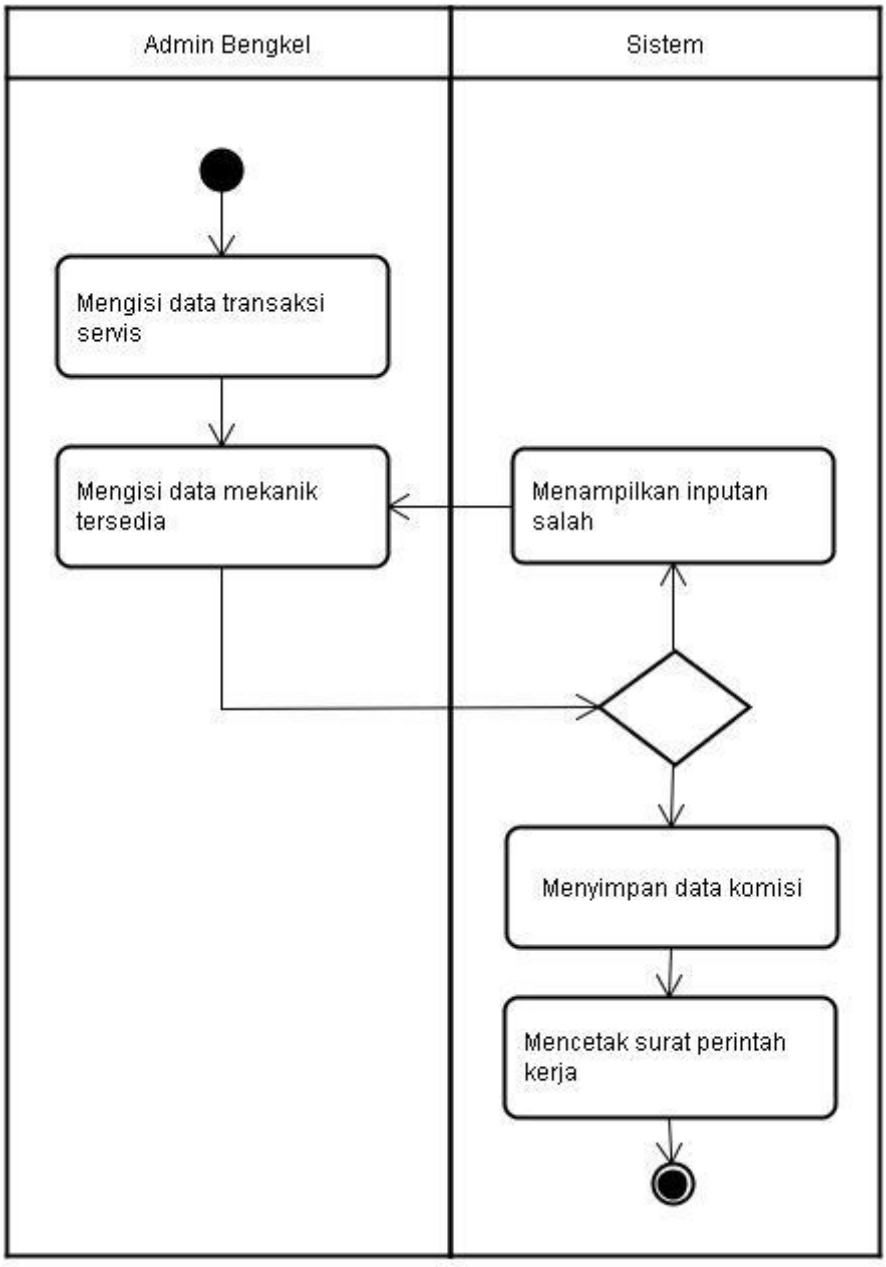
Gambar 4.9 *Activity diagram* mengolah transaksi penjualan

4. *Activity diagram* mengolah transaksi servis



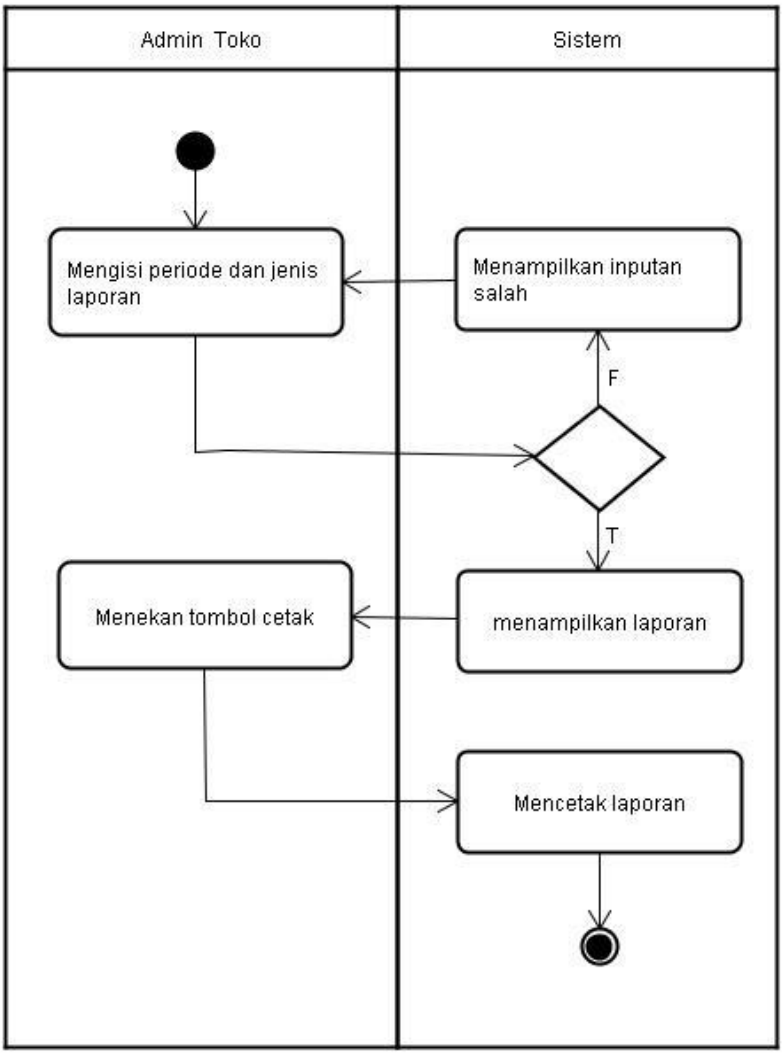
Gambar 4.10 *Activity diagram* mengolah transaksi servis

5. *Activity diagram* pencatatan komisi mekanik



Gambar 4.11 *Activity diagram* mengolah pencatatan komisi mekanik

6. *Activity diagram* mengolah laporan transaksi

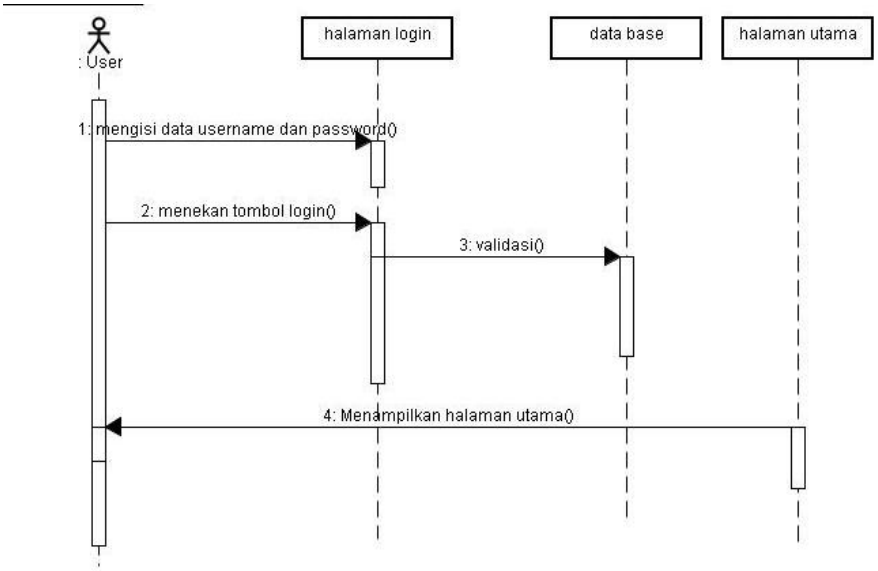


Gambar 4.12 *Activity diagram* mengolah laporan transaksi

4.1.3.6. Diagram *Sequence*

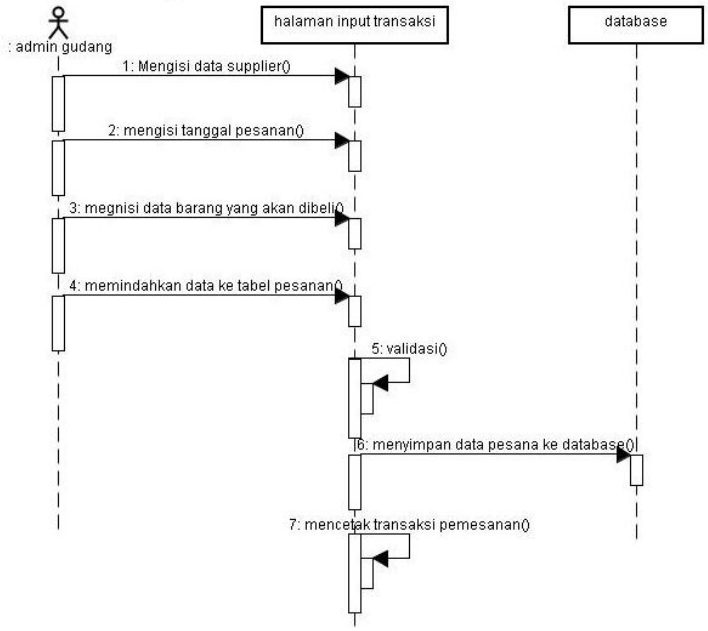
Diagram *activity* yang diusulkan untuk masing-masing bagian dari sistem yang diusulkan akan digambarkan seperti berikut ini :

1. *Sequence diagram* login *user*



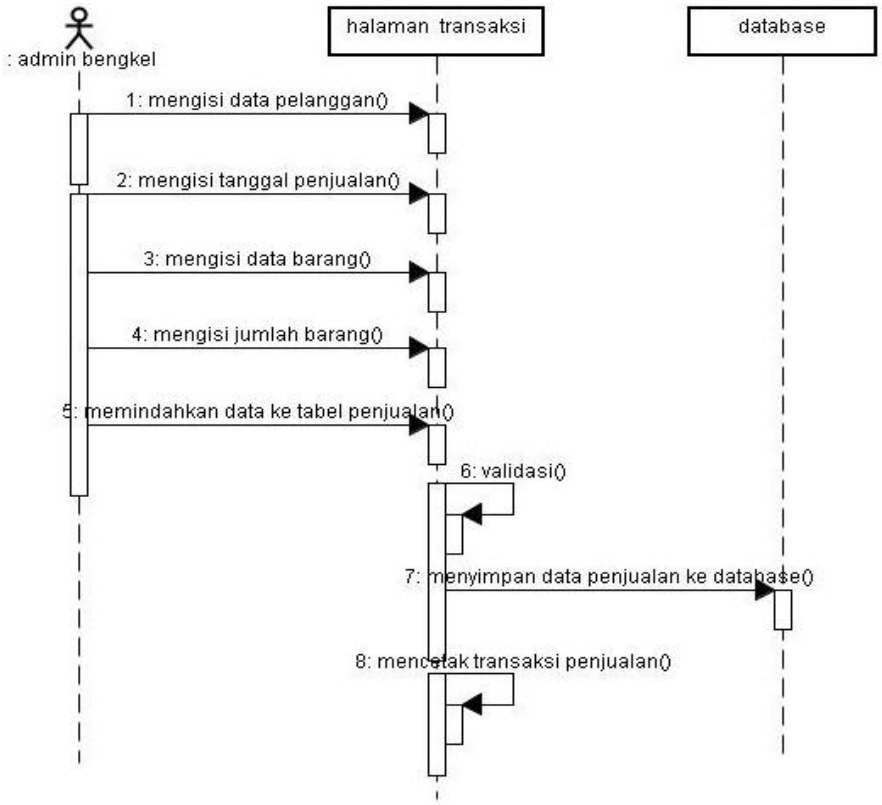
Gambar 4.13 *Sequence diagram* login *user*.

2. *Sequence diagram* transaksi pembelian



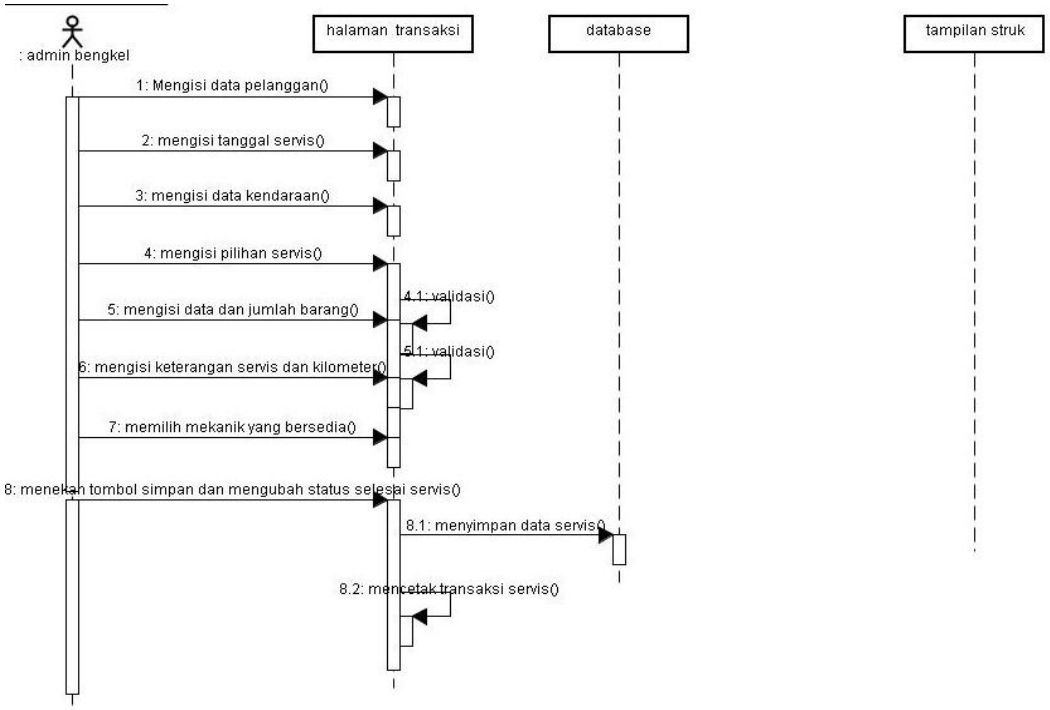
Gambar 4.14 *Sequence diagram* transaksi pembelian

3. *Sequence diagram* transaksi penjualan



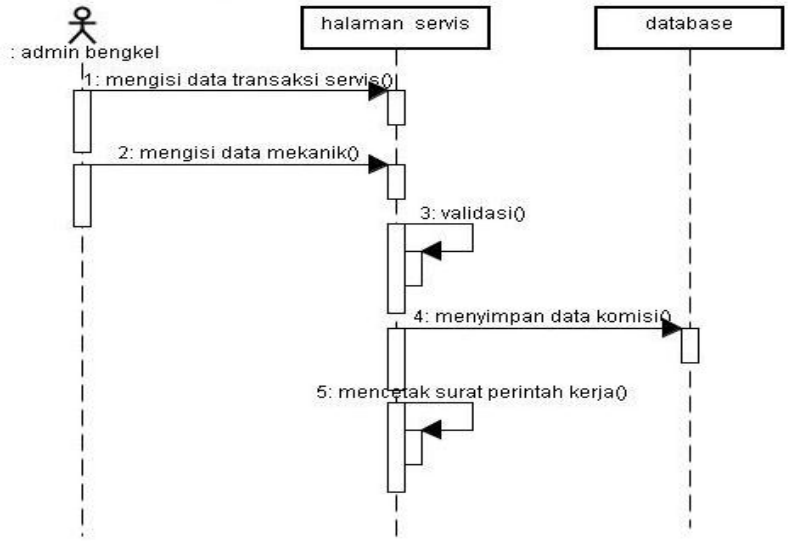
Gambar 4.15 *Sequence diagram* transaksi penjualan

4. *Sequence diagram* transaksi servis



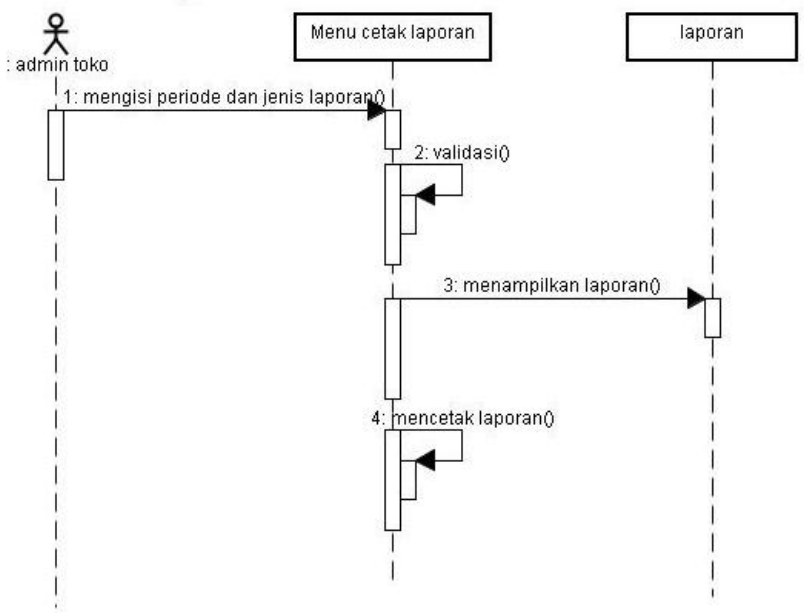
Gambar 4.16 *Sequence diagram* transaksi servis

5. *Sequence diagram* pencatatan komisi mekanik



Gambar 4.17 *Sequence diagram* pencatatan komisi mekanik

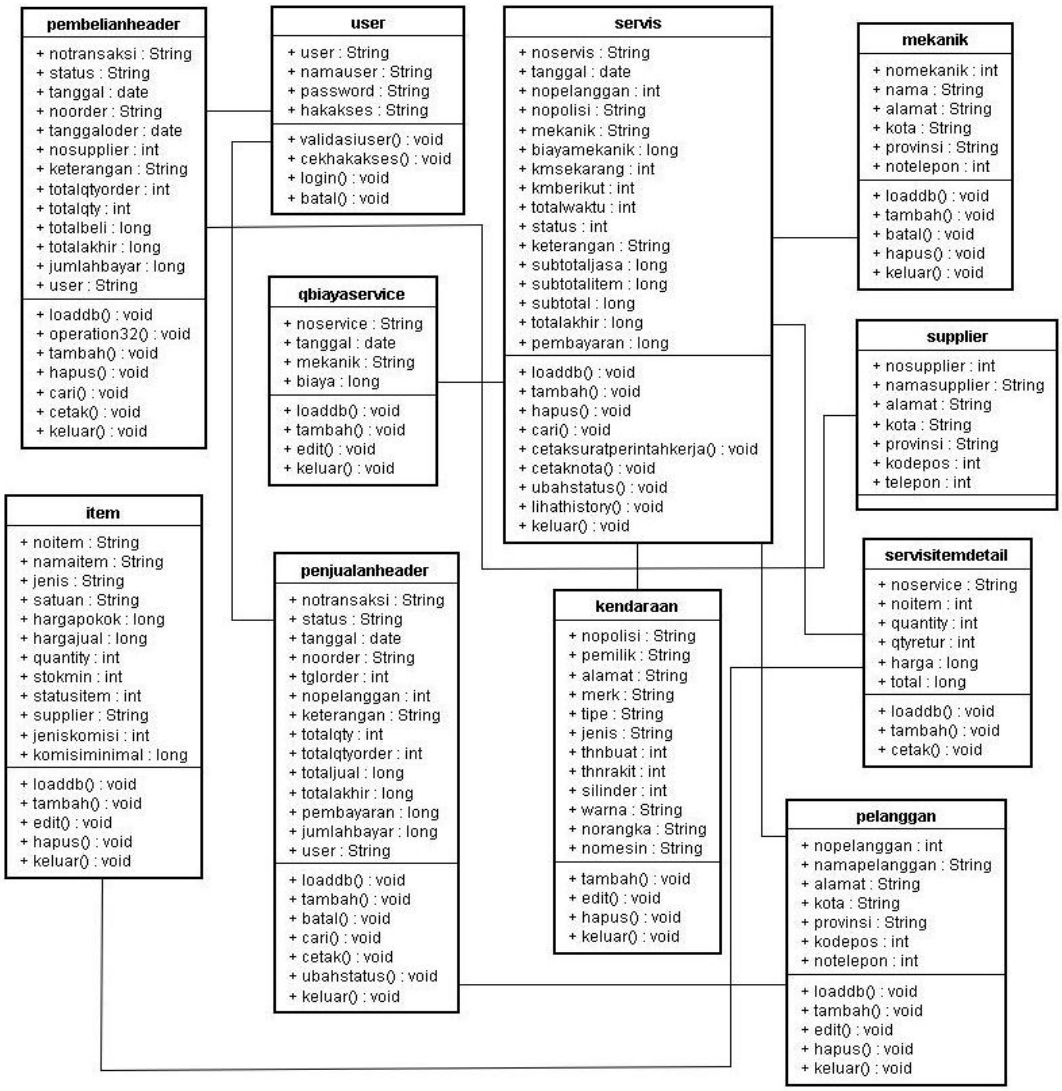
6. *Sequence diagram* pelaporan



Gambar 4.18 *Sequence diagram* pelaporan

4.1.3.7.Diagram *Class*

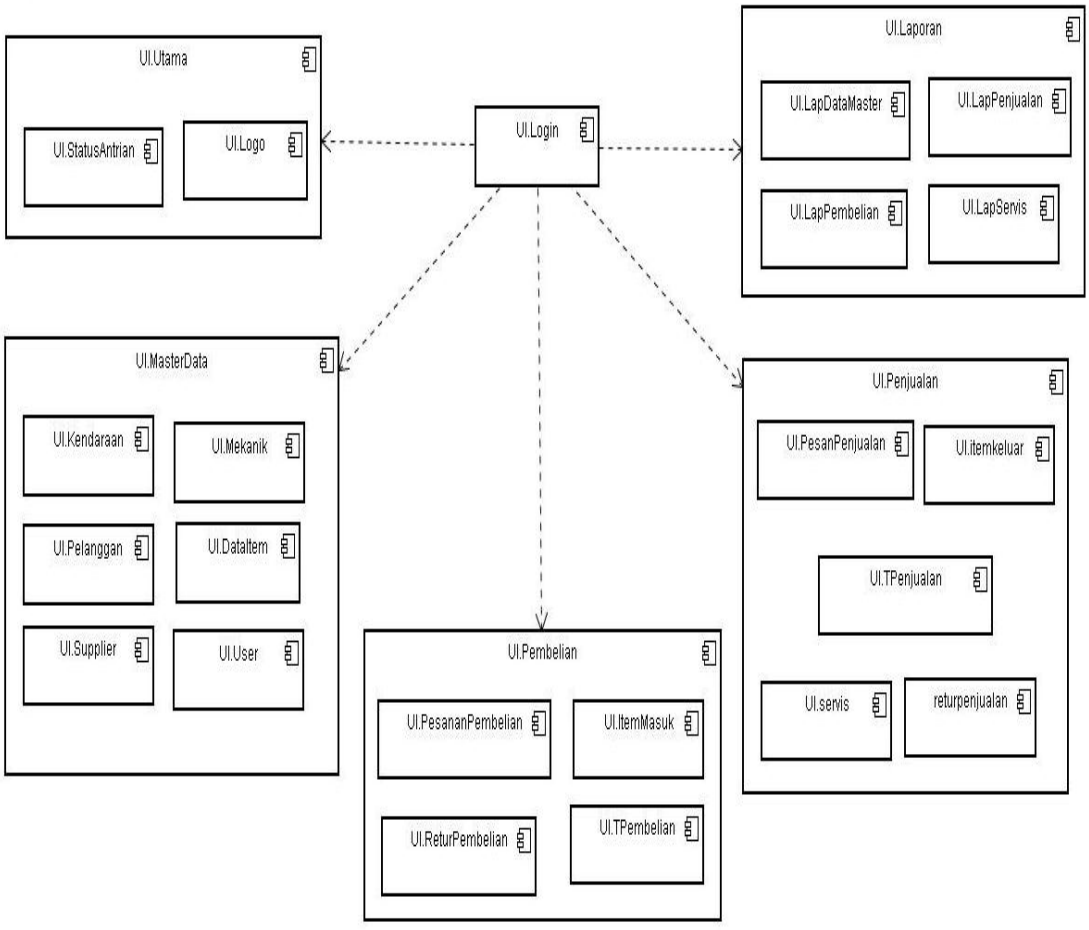
Class diagram digunakan untuk menampilkan kelas-kelas dan paket-paket didalam sistem. *Class* diagram memberikan gambaran sistem secara statis dan relasi antar mereka. Dengan melihat karakteristik sistem beserta proses-proses yang terjadi,maka dapat dibuat *class* diagram sebagai berikut :



Gambar 4.19 *Class diagram* yang diusulkan

4.1.3.8.Diagram *Component*

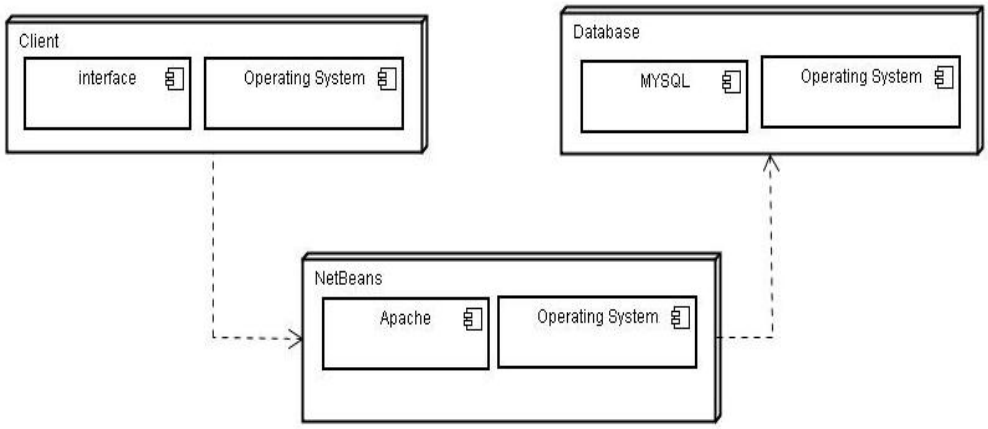
Component diagram menggambarkan struktur dan hubungan antar komponen piranti lunak, termasuk ketergantungan (*dependency*) di antaranya. Komponen piranti lunak adalah modul berisi *code* , baik berisi *source code* maupun *binary code* , baik *library* maupun *executable* , baik yang muncul pada *compile time, link time* , maupun *run time* . Berikut *component* diagram yang diusulkan :



Gambar 4.20 *Component diagram* yang diusulkan**.**

4.1.3.9.*Deployment Diagram*

Deployment / physical diagram menggambarkan detail bagaimana komponen di-deploy dalam infrastruktur sistem, di mana komponen akan terletak (pada mesin, *server* atau piranti keras apa), bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal. Sebuah node adalah *server*, *workstation*, atau piranti keras lain yang digunakan untuk men-deploy komponen dalam lingkungan sebenarnya. Hubungan antar node (misalnya TCP/IP) dan *requirement* dapat juga didefinisikan dalam diagram ini.



Gambar 4.21 *Deployment diagram* yang diusulkan**.**

4.1.3.10. Kodifikasi

Kodifikasi digunakan untuk mengklasifikasikan data. Adapun kodefikasi yang penulis gunakan dalam sistem informasi pelayanan jasa servis dan jual-beli suku cadang ini adalah sebagai berikut :

1. **No Transaksi Pemesanan Pembelian**

   Bentuk umum : AAABBBBBBCCCCC
   1 2 3

   Keterangan :

   1 : Menunjukan singkatan dari No Transaksi Pemesanan Pembelian

   2 : Menunjukan informasi tanggal, bulan, tahun transaksi

   3 : Menunjukan urutan Transaksi Pemesanan Pembelian

   Contoh : NPS09061400001

2. **No Transaksi Pembelian**

   Bentuk umum : AAABBBBBBCCCCC
   1 2 3

   Keterangan :

   1 : Menunjukan singkatan dari No Transaksi Pembelian

   2 : Menunjukan informasi tanggal, bulan, tahun transaksi

   3 : Menunjukan urutan Transaksi Pembelian

   Contoh : TPB09061400001

3. **No Transaksi Retur Pembelian**

   Bentuk umum : AAABBBBBBCCCCC
   1 2 3

   Keterangan :

   1 : Menunjukan singkatan dari No Retur Pembelian

   2 : Menunjukan informasi tanggal, bulan, tahun transaksi

3 : Menunjukan urutan Transaksi Retur Pembelian

Contoh : RIB09061400001

4. No Transaksi Item Masuk

Bentuk umum : AAABBBBBBCCCCC

1 2 3

Keterangan :

1 : Menunjukan singkatan dari No Item Masuk

2 : Menunjukan informasi tanggal, bulan, tahun transaksi

3 : Menunjukan urutan Item Masuk

Contoh : IMS09061400001

5. No Transaksi Pemesanan Penjualan

Bentuk umum : AAABBBBBBCCCCC

1 2 3

Keterangan :

1 : Menunjukan singkatan dari No Pemesanan Penjualan

2 : Menunjukan informasi tanggal, bulan, tahun transaksi

3 : Menunjukan urutan Pemesanan Penjualan

Contoh : PSJ09061400001

6. No Transaksi Penjualan

Bentuk umum : AAABBBBBBCCCCC

1 2 3

Keterangan :

1 : Menunjukan singkatan dari No Transaksi Penjualan

2 : Menunjukan informasi tanggal, bulan, tahun transaksi

3 : Menunjukan urutan Transaksi Penjualan

Contoh : TRP09061400001

7. No Transaksi Servis dan Penjualan

Bentuk umum : AAABBBBBBCCCCC

1 2 3

Keterangan :

1 : Menunjukan singkatan dari No Transaksi Servis dan Penjualan

2 : Menunjukan informasi tanggal, bulan, tahun transaksi

3 : Menunjukan urutan Transaksi Servis dan Penjualan

Contoh : TRS09061400001

8. No Transaksi Retur Penjualan

Bentuk umum : AAABBBBBBCCCCC

1 2 3

Keterangan :

1 : Menunjukan singkatan dari No Retur Penjualan

2 : Menunjukan informasi tanggal, bulan, tahun transaksi

3 : Menunjukan urutan Retur Penjualan

Contoh : RIJ09061400001

9. No Transaksi Item Keluar

Bentuk umum : AAABBBBBBCCCCC

1 2 3

Keterangan :

1 : Menunjukan singkatan dari Item Keluar

2 : Menunjukan informasi tanggal, bulan, tahun transaksi

3 : Menunjukan urutan Item Keluar

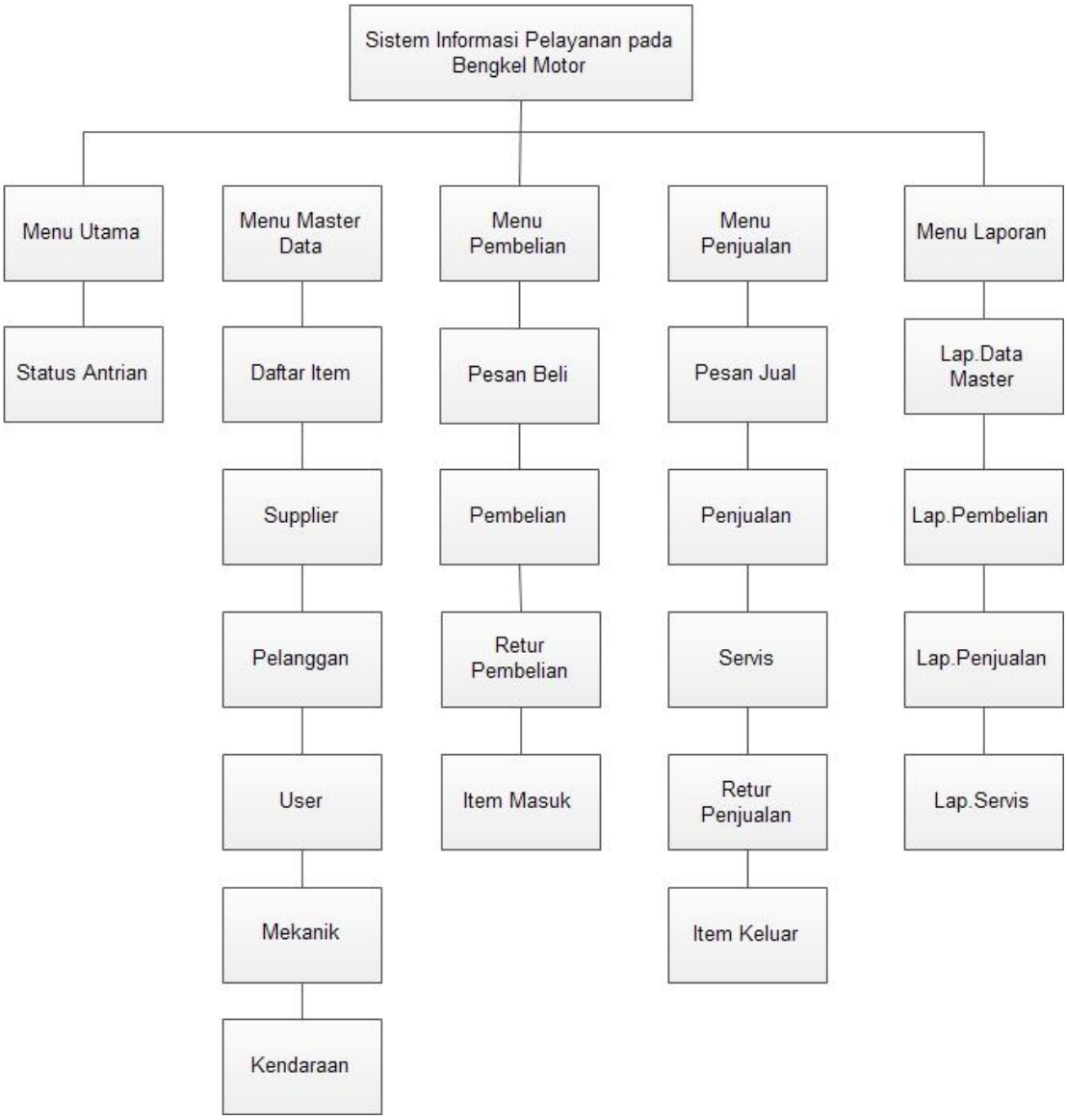
Contoh : IKL09061400001

4.2. Perancangan Antar Muka (*Interface*)

Perancangan antarmuka merupakan perancangan model aplikasi sebelum program dibuat seutuhnya, perancangan antarmuka pada sistem ini bertujuan untuk memudahkan dalam pembuatan program nantinya.

4.2.1. Struktur Menu

Perancangan menu di gunakan untuk memudahkan dalam penelusuran program yang di buat. Berikut adalah gambar perancangan nya :



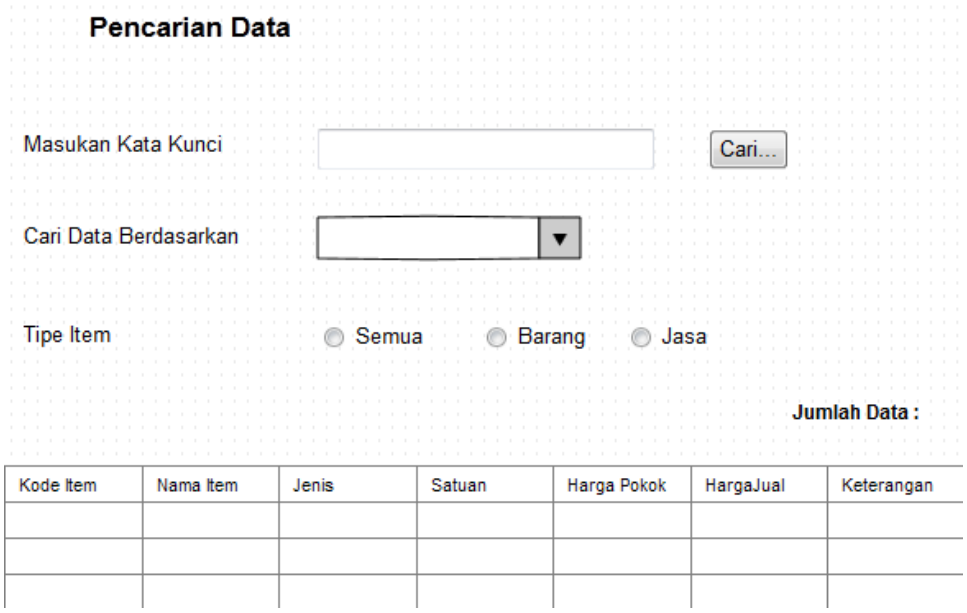
Gambar 4.22 Rancangan Struktur Menu

4.2.2. Perancangan Input

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

1. Rancangan Master item

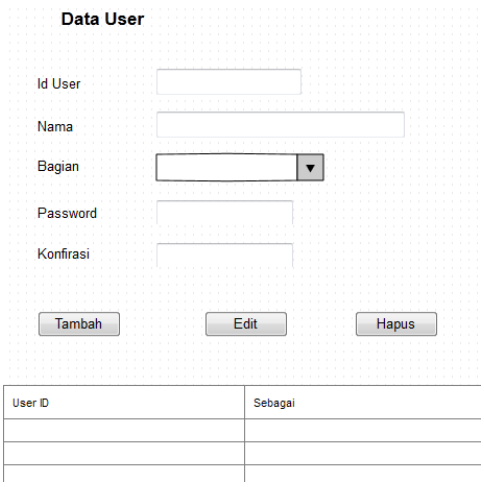
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data barang maupun jasa pada database.



Gambar 4.23 Menu Master Item

2. Rancangan Master User

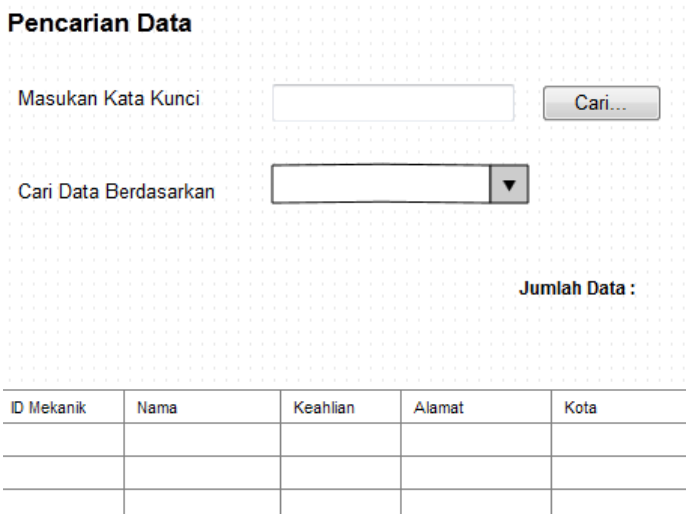
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data *User* pada database. *User* nantinya berfungsi untuk *Login* ke sistem.



Gambar 4.24 Menu Master User

3. Rancangan Master Mekanik

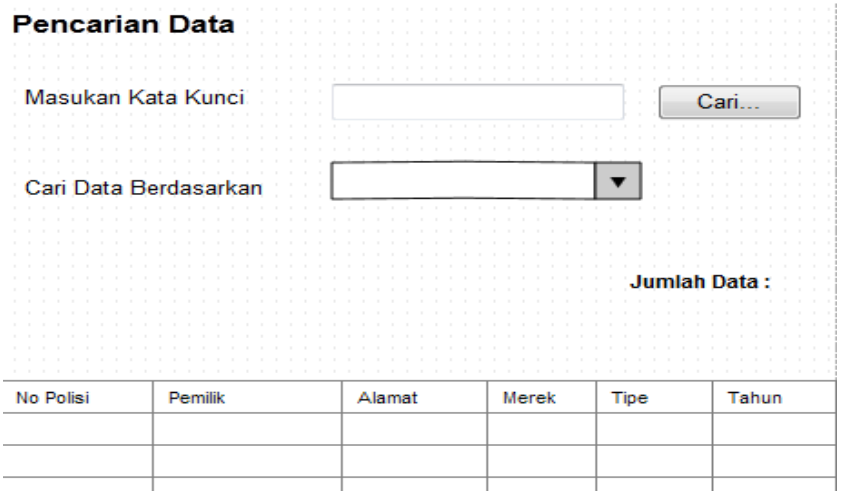
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data *Mekanik* pada database.



Gambar 4.25 Menu Master Mekanik

4. Rancangan Master Kendaraan

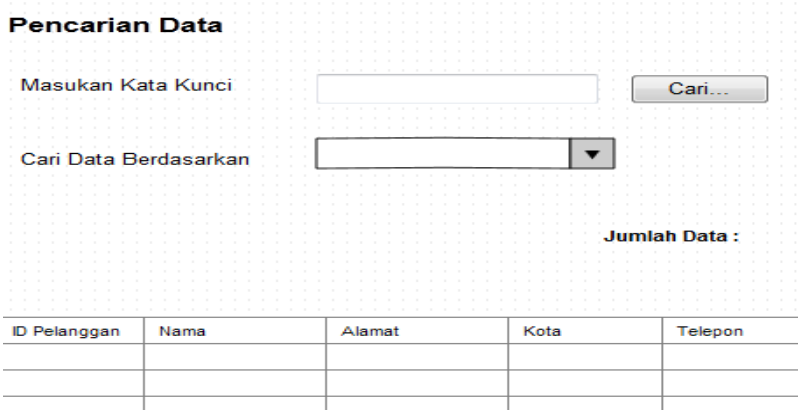
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data kendaraan pada database.



Gambar 4.26 Menu Master Kendaraan

5. Rancangan Master Pelanggan

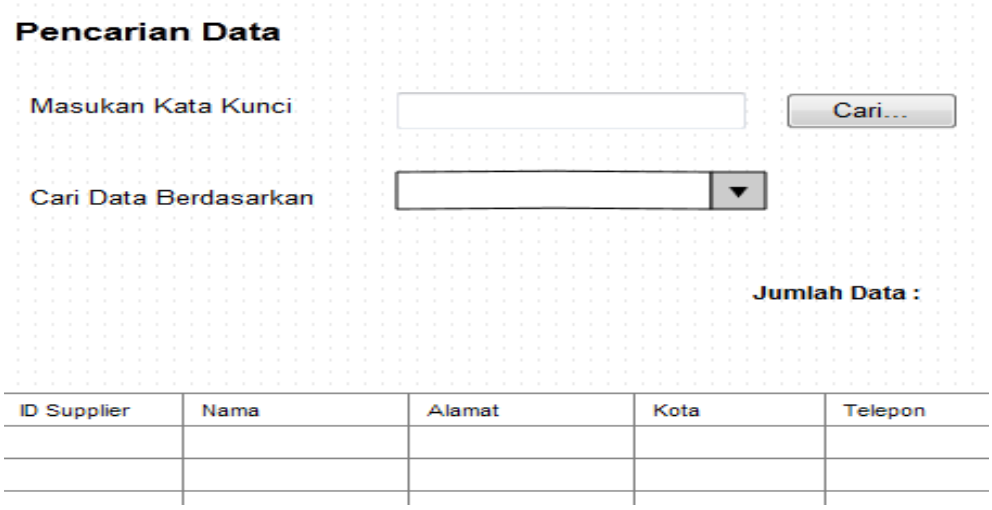
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data pelanggan pada database.



Gambar 4.27 Menu Master Pelanggan

6. Rancangan Master Supplier

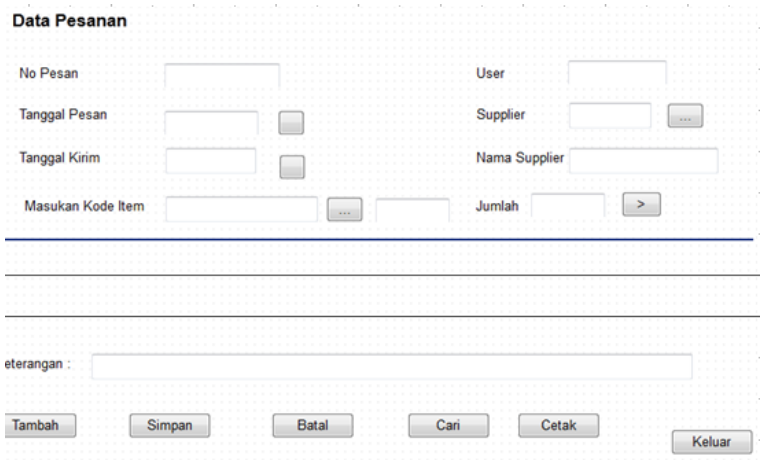
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data supplier pada database.



Gambar 4.28 Menu Master Supplier

7. Rancangan Pesanan Pembelian

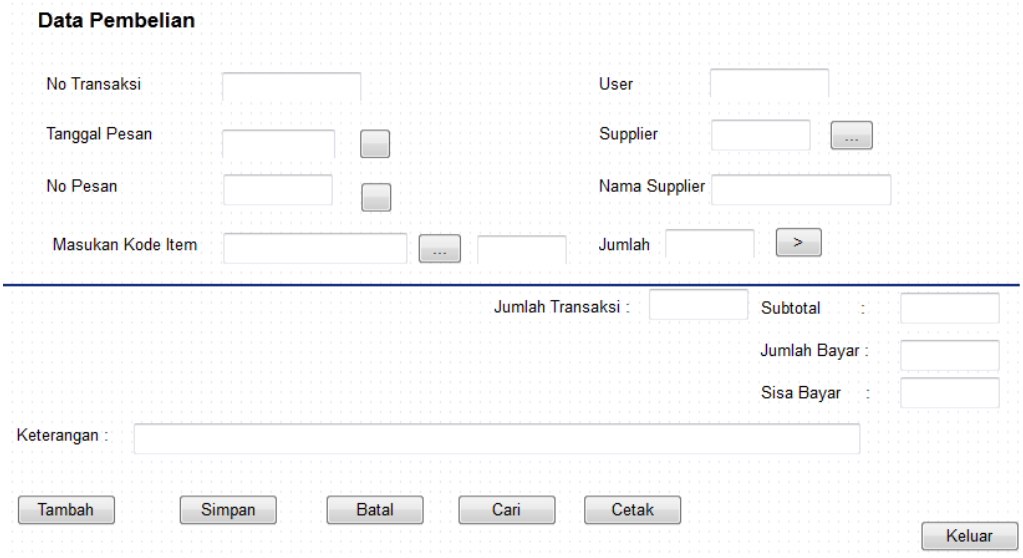
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data pesanan pembelian pada database.



Gambar 4.29 Menu Pesanan Pembelian

8. Rancangan Transaksi Pembelian

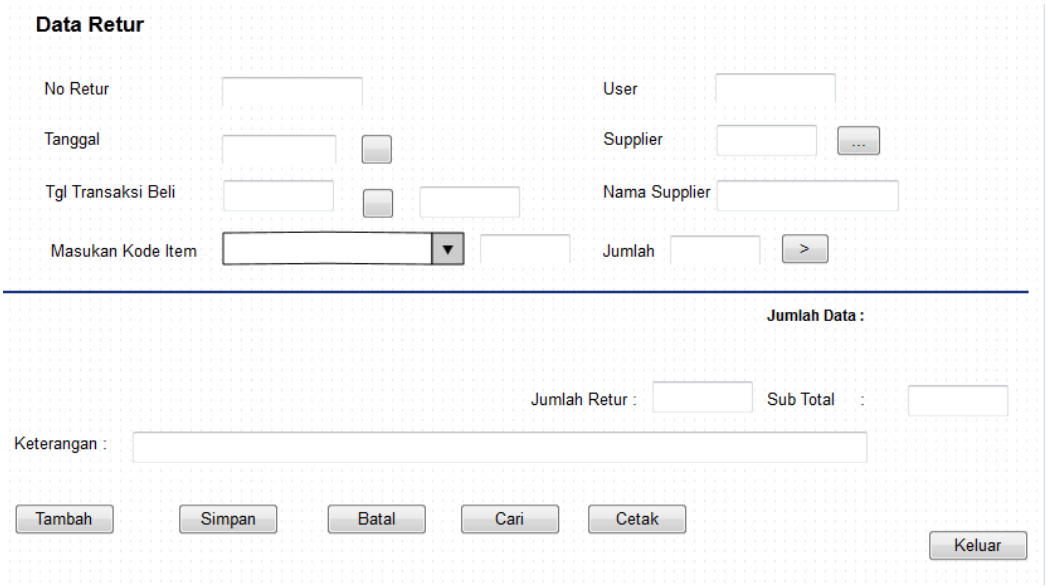
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data transaksi pembelian pada database.



Gambar 4.30 Menu Transaksi Pembelian

9. Rancangan Retur Pembelian

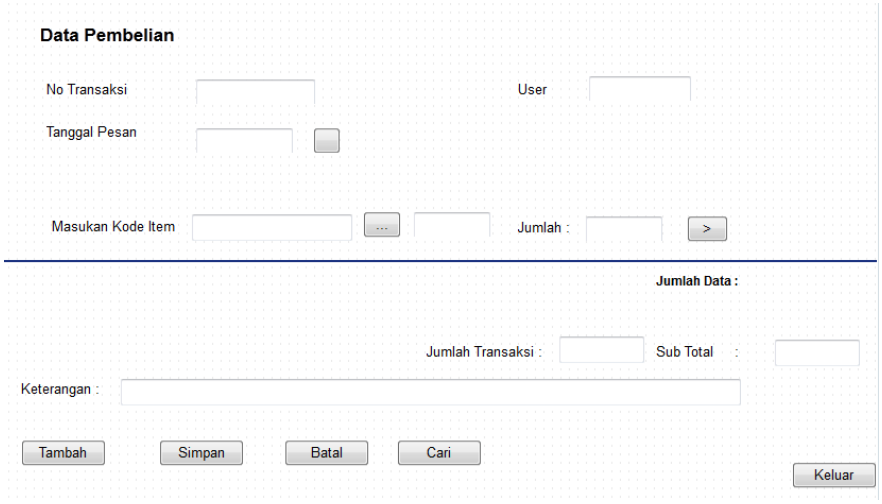
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data retur pembelian pada database.



Gambar 4.31 Menu Retur Pembelian

10. Rancangan Item Masuk

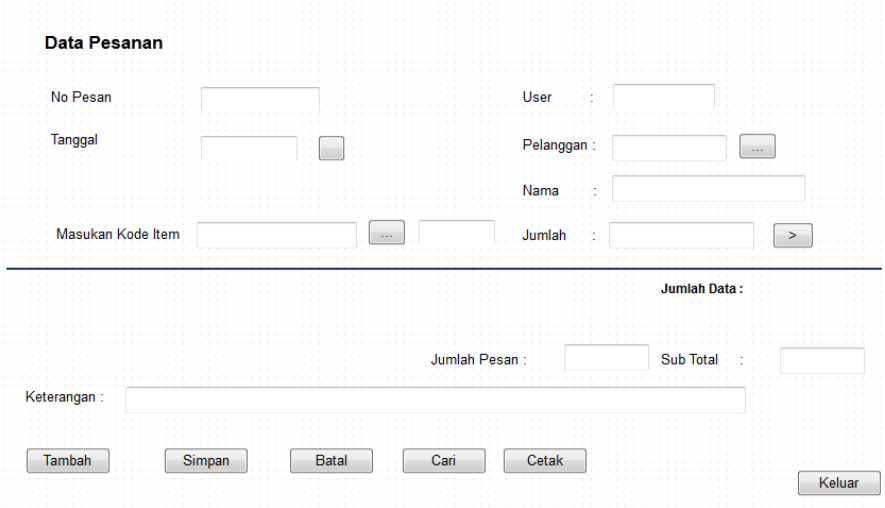
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data item masuk pada database.



Gambar 4.32 Menu Item Masuk

11. Rancangan Pesanan Penjualan

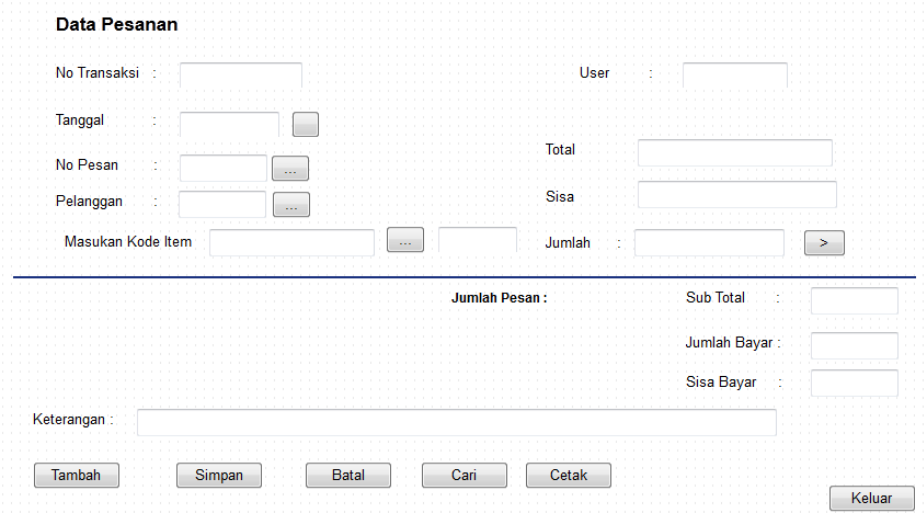
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data pesanan penjualan pada database.



Gambar 4.33 Menu Pesanan Penjualan

12. Rancangan Transaksi Penjualan

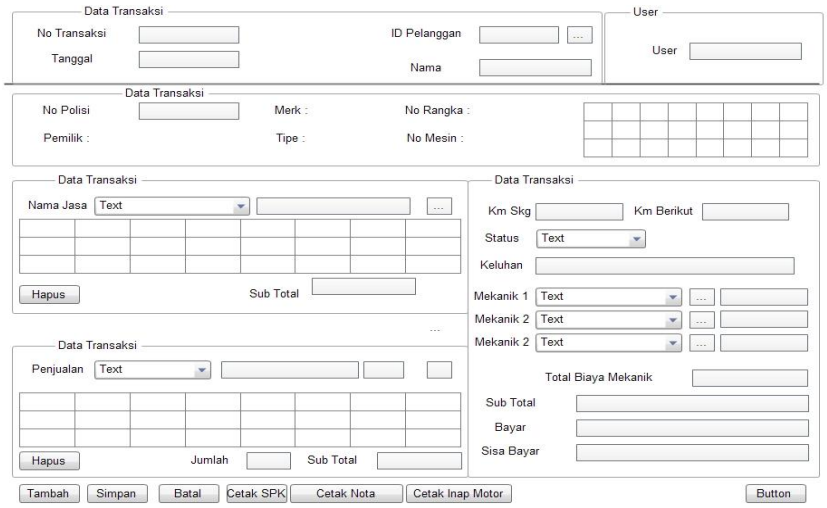
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data transaksi penjualan pada database.



Gambar 4.34 Menu Transaksi Penjualan

13. Rancangan Transaksi Servis dan Penjualan

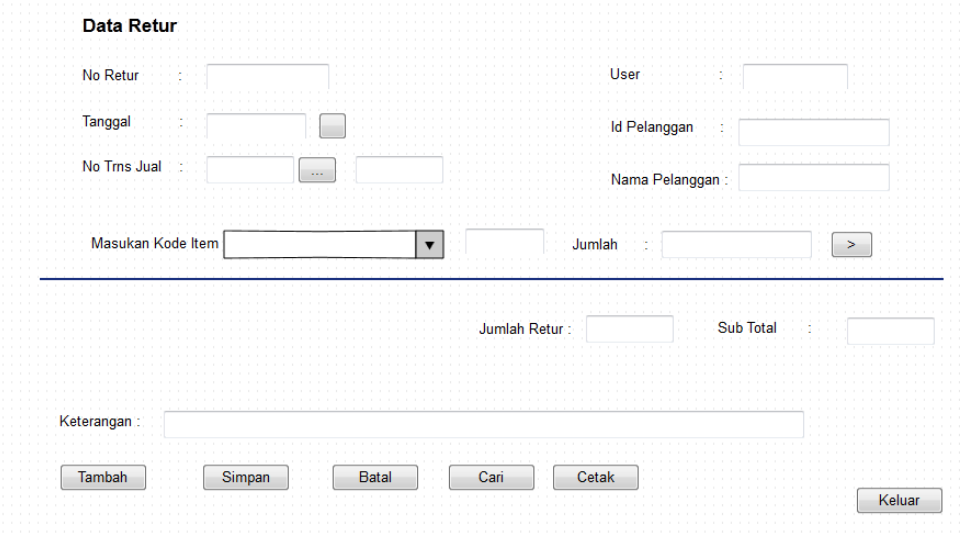
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data transaksi servis dan penjualan pada database.



Gambar 4.35 Menu Transaksi Servis dan Penjualan

14. Rancangan Retur Penjualan

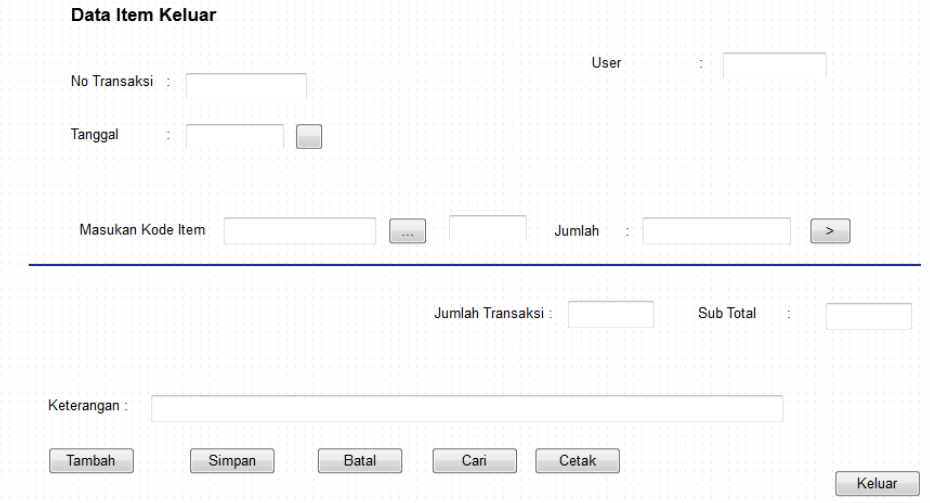
Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data Retur Penjualan pada database.



Gambar 4.36 Menu Retur Penjualan

15. Rancangan Item Keluar

Tampilan yang dirancang berfungsi untuk melakukan penambahan , pengubahan , pencarian atau pun menghapus data Item Keluar pada database.



Gambar 4.37 Menu Item Keluar

4.2.3. Perancangan Output

Perancangan *Output* merupakan hasil dari apa yang kita Inputkan pada program yang dibuat, *Output* yang penulis maksud adalah laporan dari semua data yang telah diinputkan dan disimpan kedalam database kemudian di panggil kembali dalam bentuk laporan.

1. Laporan Master Item

Tampilan laporan Master Item merupakan *Output* yang dihasilkan dari *input* Master Item, Laporan ini berupa informasi tentang seluruh data Master Item yang ada didalam database.

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

DAFTAR ITEM

TIPE : barang

No	Kode Item	Nama Item	Jenis	Satuan	Harga Pokok	Harga Jual	Stok	Keterangan
1	KIB-00001	nama item	OIL	DUS	45000	50000	251	barang ada
2	KIB-00002	barang	OIL	NON	35000	40000	429	rdrr
3	KIB-00003	Karburator	ALTHD	BOX	30000	40000	225	3
4	KIB-00004	shockbreak	OIL	NON	30000	35000	198	333
5	KIB-00005	rem	ALTHD	BOX	30000	60000	194	33
6	KIB-00006	baut	OIL	BOX	30000	35000	200	333
7	KIB-00007	rantai	ALTHD	BOX	30000	35000	223	333
8	KIB-00008	velg	ALTHD	DUS	30000	35000	200	333
9	KIB-00009	helm	ALTHD	PAK	30000	35000	414	33

Gambar 4.38 Laporan Master Item

2. Laporan Master Pelanggan

Tampilan laporan Master Pelanggan merupakan *Output* yang dihasilkan dari *input* Master Pelanggan, Laporan ini berupa informasi tentang seluruh data Master Pelanggan yang ada didalam database.

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

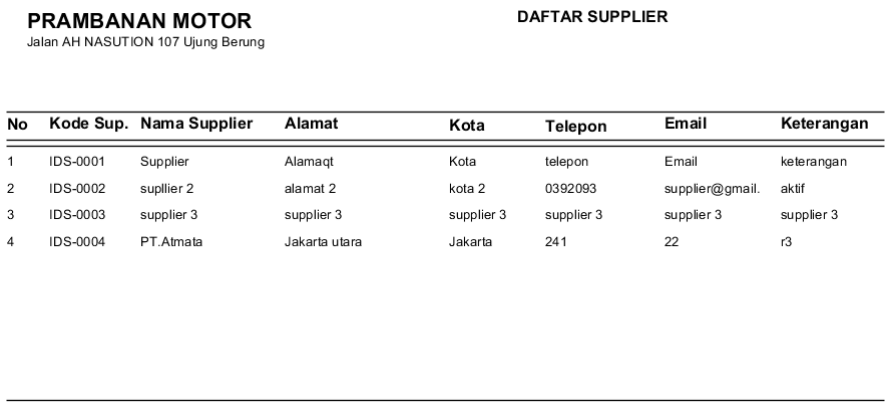
DAFTAR PELANGGAN

No	ID Pelanggan	Nama Supplier	Alamat	Kota	Telepon	Keterangan
1	IDP-0001	Dayat	sekeloa	Bandung	0123456789	BAgus
2	IDP-0002	marzan nurgraha	bandung	bandung	3242	

Gambar 4.39 Laporan Master Pelanggan

3. Laporan Master Supplier

Tampilan laporan Master Supplier merupakan *Output* yang dihasilkan dari *input* Master Supplier, Laporan ini berupa informasi tentang seluruh data Master Supplier yang ada didalam database.



PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

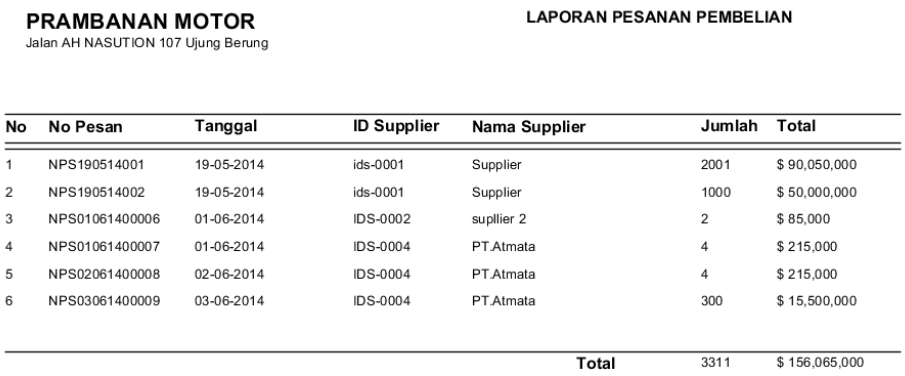
DAFTAR SUPPLIER

No	Kode Sup.	Nama Supplier	Alamat	Kota	Telepon	Email	Keterangan
1	IDS-0001	Supplier	Alamaqt	Kota	telepon	Email	keterangan
2	IDS-0002	supllier 2	alamat 2	kota 2	0392093	supplier@gmail.	aktif
3	IDS-0003	supplier 3	supplier 3	supplier 3	supplier 3	supplier 3	supplier 3
4	IDS-0004	PT.Atmata	Jakarta utara	Jakarta	241	22	r3

Gambar 4.40 Laporan Master Supplier

4. Laporan Transaksi Pesanan

Tampilan laporan Transaksi Pesanan Pembelian merupakan *Output* yang dihasilkan dari *input* Transaksi Pesanan Pembelian, Laporan ini berupa informasi tentang seluruh data Transaksi Pesanan Pembelian yang ada didalam database.



PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN PESANAN PEMBELIAN

No	No Pesan	Tanggal	ID Supplier	Nama Supplier	Jumlah	Total
1	NPS190514001	19-05-2014	ids-0001	Supplier	2001	$ 90,050,000
2	NPS190514002	19-05-2014	ids-0001	Supplier	1000	$ 50,000,000
3	NPS01061400006	01-06-2014	IDS-0002	supllier 2	2	$ 85,000
4	NPS01061400007	01-06-2014	IDS-0004	PT.Atmata	4	$ 215,000
5	NPS02061400008	02-06-2014	IDS-0004	PT.Atmata	4	$ 215,000
6	NPS03061400009	03-06-2014	IDS-0004	PT.Atmata	300	$ 15,500,000
				Total	3311	$ 156,065,000

Gambar 4.41 Laporan Transaksi Pesanan Pembelian

5. Laporan Transaksi Pembelian

Tampilan laporan Transaksi Pembelian merupakan *Output* yang dihasilkan dari *input* Transaksi Pembelian, Laporan ini berupa informasi tentang seluruh data Transaksi Pembelian yang ada didalam database.

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN PEMBELIAN

No	No Transaksi	Tanggal	ID Supplier	Nama Supplier		Jumlah	Total
1	TRB200500001	20-05-2014	IDS-0001	Supplier		10	$ 500,000
					Total	10	$ 500,000
					Bayar		$ 500,000
2	TRB240500011	24-05-2014	IDS-0002	supllier 2		900	$ 36,500,000
					Total	900	$ 36,500,000
					Bayar		$ 40,000,000
3	TRB290500012	29-05-2014	IDS-0002	supllier 2		39	$ 1,610,000
					Total	39	$ 1,610,000
					Bayar		$ 1,610,000
4	TRB010600018	01-06-2014	IDS-0002	supllier 2		1	$ 40,000
					Total	1	$ 40,000
					Bayar		$ 80,000
5	TRB290500013	29-05-2014	IDS-0003	supplier 3		36	$ 1,310,000
					Total	36	$ 1,310,000
					Bayar		$ 1,310,000

Gambar 4.42 Laporan Transaksi Pembelian

6. Laporan Transaksi Retur Pembelian

Tampilan laporan Transaksi Retur Pembelian merupakan *Output* yang dihasilkan dari *input* Transaksi Retur Pembelian, Laporan ini berupa informasi tentang seluruh data Transaksi Retur Pembelian yang ada didalam database.

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN RETUR PEMBELIAN

No	No Transaksi	Tanggal	ID Supplier	Nama Supplier	Jumlah	Total
1	TRB200500001	26-05-2014	IDS-0001	Supplier	1	$ 50,000
2	TRB240500011	26-05-2014	IDS-0002	supllier 2	2	$ 85,000
3	TRB240500011	03-06-2014	IDS-0002	supllier 2	2	$ 85,000
4	TRB030600020	03-06-2014	IDS-0004	PT.Atmata	2	$ 75,000

Total Keseluruhann

Jml Retur : $ 7
Total : $ 295,000

Gambar 4.43 Laporan Transaksi Retur Pembelian

7. Laporan Transaksi Pesanan Penjualan

Tampilan laporan Transaksi Pesanan Penjualan merupakan *Output* yang dihasilkan dari *input* Transaksi Pesanan Penjualan, Laporan ini berupa informasi tentang seluruh data Transaksi Pesanan Penjualan yang ada didalam database.

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN PESANAN PENJUALAN

No	No Pesan	Tanggal	Pelanggan	Jumlah	Total
1	PSJ010600003	01-06-2014	IDP-0002	1	$ 40,000
2	PSJ030600004	03-06-2014	IDP-0002	5	$ 315,000
3	PSJ260500001	26-05-2014	IDP-0001	1	$ 35,000

Jml Beli : 7
Total : $ 390,000

Gambar 4.44 Laporan Transaksi Pesanan Penjualan

8. Laporan Transaksi Penjualan

Tampilan laporan Transaksi Penjualan merupakan *Output* yang dihasilkan dari *input* Transaksi Penjualan, Laporan ini berupa informasi tentang seluruh data Transaksi Penjualan yang ada didalam database.

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

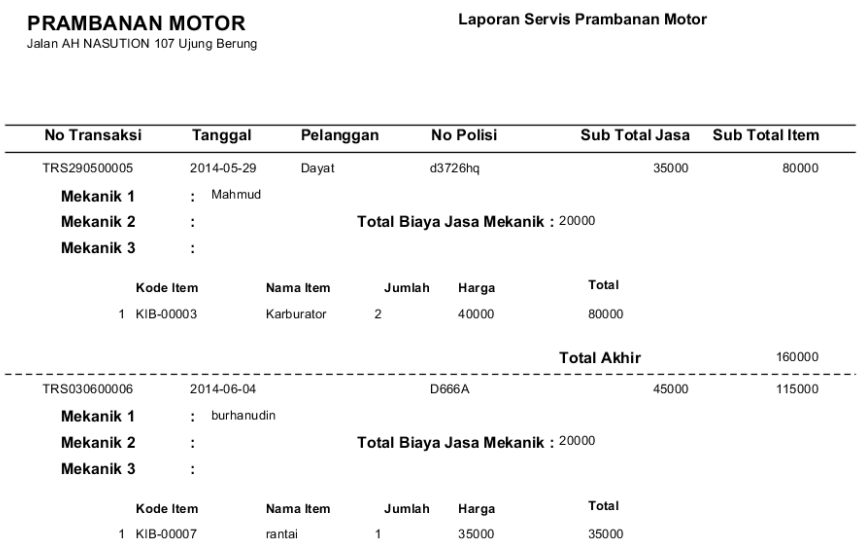
LAPORAN PENJUALAN

No Transaksi	Tanggal	No Pesanan	Tgl Pesanan	Kode Pelanggan	Nama
TRP270500002	27-05-2014		27-05-2014	IDP-0001	Dayat
No	**Kode**	**Nama Item**	**Jml Jual**	**Harga Jual**	**Total**
1	KIB-00006	baut	1	35000	35000
TRP270500003	27-05-2014		27-05-2014	IDP-0001	Dayat
No	**Kode**	**Nama Item**	**Jml Jual**	**Harga Jual**	**Total**
2	KIB-00001	nama item	10	50000	500000
TRP270500003	27-05-2014		27-05-2014	IDP-0001	Dayat
No	**Kode**	**Nama Item**	**Jml Jual**	**Harga Jual**	**Total**
3	KIB-00005	rem	6	60000	360000
TRP030600006	03-06-2014	PSJ030600004	03-06-2014	IDP-0002	marzan nurgraha
No	**Kode**	**Nama Item**	**Jml Jual**	**Harga Jual**	**Total**
4	KIB-00002	barang	1	40000	40000
TRP030600006	03-06-2014	PSJ030600004	03-06-2014	IDP-0002	marzan nurgraha

Gambar 4.45 Laporan Transaksi Penjualan

9. Laporan Transaksi Servis

Tampilan laporan Transaksi Servis merupakan *Output* yang dihasilkan dari *input* Transaksi Servis, Laporan ini berupa informasi tentang seluruh data Transaksi Servis yang ada didalam database.



PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

Laporan Servis Prambanan Motor

No Transaksi	Tanggal	Pelanggan	No Polisi	Sub Total Jasa	Sub Total Item
TRS290500005	2014-05-29	Dayat	d3726hq	35000	80000

Mekanik 1 : Mahmud
Mekanik 2 :
Mekanik 3 :
Total Biaya Jasa Mekanik : 20000

	Kode Item	Nama Item	Jumlah	Harga	Total
1	KIB-00003	Karburator	2	40000	80000

Total Akhir 160000

No Transaksi	Tanggal	Pelanggan	No Polisi	Sub Total Jasa	Sub Total Item
TRS030600006	2014-06-04		D666A	45000	115000

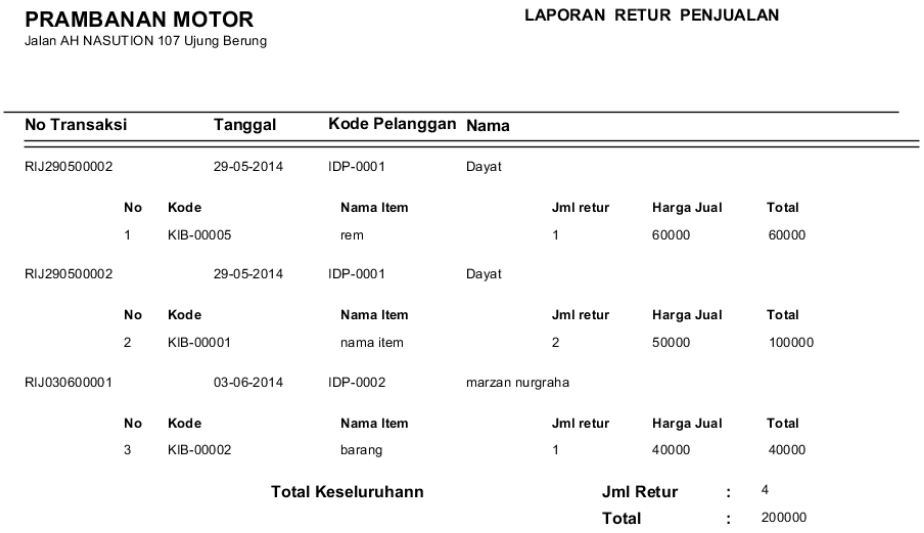
Mekanik 1 : burhanudin
Mekanik 2 :
Mekanik 3 :
Total Biaya Jasa Mekanik : 20000

	Kode Item	Nama Item	Jumlah	Harga	Total
1	KIB-00007	rantai	1	35000	35000

Gambar 4.46 Laporan Transaksi Servis

10. Laporan Transaksi Retur Penjualan

Tampilan laporan Transaksi Retur Penjualan merupakan *Output* yang dihasilkan dari *input* Transaksi Retur Penjualan, Laporan ini berupa informasi tentang seluruh data Retur Penjualan yang ada didalam database.



PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN RETUR PENJUALAN

No Transaksi	Tanggal	Kode Pelanggan	Nama
RIJ290500002	29-05-2014	IDP-0001	Dayat

No	Kode	Nama Item	Jml retur	Harga Jual	Total
1	KIB-00005	rem	1	60000	60000

No Transaksi	Tanggal	Kode Pelanggan	Nama
RIJ290500002	29-05-2014	IDP-0001	Dayat

No	Kode	Nama Item	Jml retur	Harga Jual	Total
2	KIB-00001	nama item	2	50000	100000

No Transaksi	Tanggal	Kode Pelanggan	Nama
RIJ030600001	03-06-2014	IDP-0002	marzan nurgraha

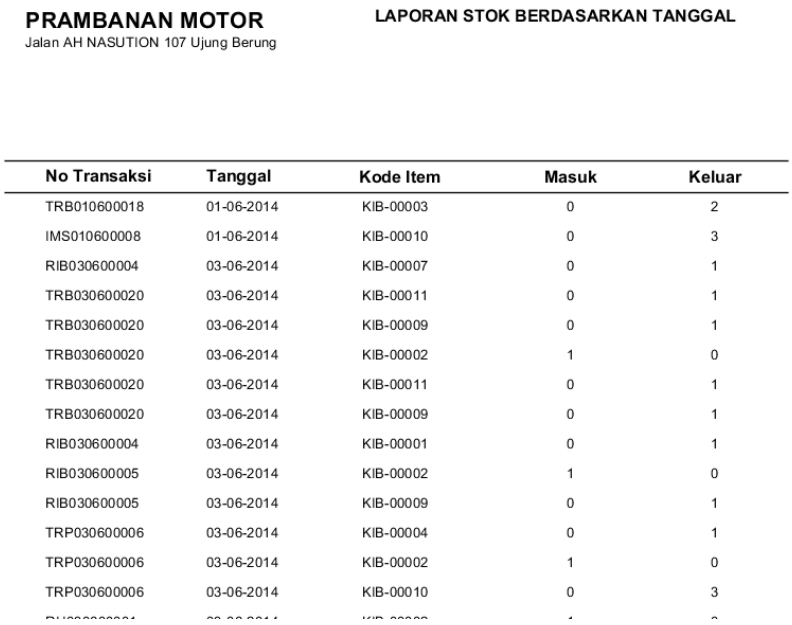
No	Kode	Nama Item	Jml retur	Harga Jual	Total
3	KIB-00002	barang	1	40000	40000

Total Keseluruhann
Jml Retur : 4
Total : 200000

Gambar 4.47 Laporan Transaksi Retur Penjualan

11. Laporan Stok Barang

Tampilan laporan Stok Barang merupakan *Output* yang dihasilkan dari *input* Transaksi Penjualan maupun Pembelian, Laporan ini berupa informasi tentang seluruh data Stok Barang yang ada didalam database.



PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN STOK BERDASARKAN TANGGAL

No Transaksi	Tanggal	Kode Item	Masuk	Keluar
TRB010600018	01-06-2014	KIB-00003	0	2
IMS010600008	01-06-2014	KIB-00010	0	3
RIB030600004	03-06-2014	KIB-00007	0	1
TRB030600020	03-06-2014	KIB-00011	0	1
TRB030600020	03-06-2014	KIB-00009	0	1
TRB030600020	03-06-2014	KIB-00002	1	0
TRB030600020	03-06-2014	KIB-00011	0	1
TRB030600020	03-06-2014	KIB-00009	0	1
RIB030600004	03-06-2014	KIB-00001	0	1
RIB030600005	03-06-2014	KIB-00002	1	0
RIB030600005	03-06-2014	KIB-00009	0	1
TRP030600006	03-06-2014	KIB-00004	0	1
TRP030600006	03-06-2014	KIB-00002	1	0
TRP030600006	03-06-2014	KIB-00010	0	3

Gambar 4.48 Laporan Stok Barang

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN STOK BERDASARKAN KODE

No Transaksi	Tanggal	Kode Item	Masuk	Keluar
RIB030600004	03-06-2014	KIB-00001	0	1
TRB240500011	24-05-2014	KIB-00001	0	1
RIB260500001	26-05-2014	KIB-00001	0	1
RIB260500002	26-05-2014	KIB-00001	0	1
TRP270500003	27-05-2014	KIB-00001	0	1
IMS290500001	29-05-2014	KIB-00001	0	1
TRB290500012	29-05-2014	KIB-00001	0	1
IKL290500001	29-05-2014	KIB-00001	0	1
RIJ290500002	29-05-2014	KIB-00001	0	1

Stok Masuk	0
Stok Keluar	9

Gambar 4.48 Laporan Stok Barang berdasarkan Kode

12. Laporan Komisi Mekanik

Tampilan laporan Komisi Mekanik merupakan *Output* yang dihasilkan dari *input* Transaksi Servis, Laporan ini berupa informasi tentang seluruh data Komisi Mekanik yang ada didalam database.

PRAMBANAN MOTOR
Jalan AH NASUTION 107 Ujung Berung

LAPORAN BIAYA SERVIS MEKANIK

Kode Mekanik IDM-0001
Nama Mekanik Mahmud

No Transaksi	Tanggal	Total Biaya Jasa
TRS290500005	29-05-2014 00:00:00	20000
	Sub Total	20000

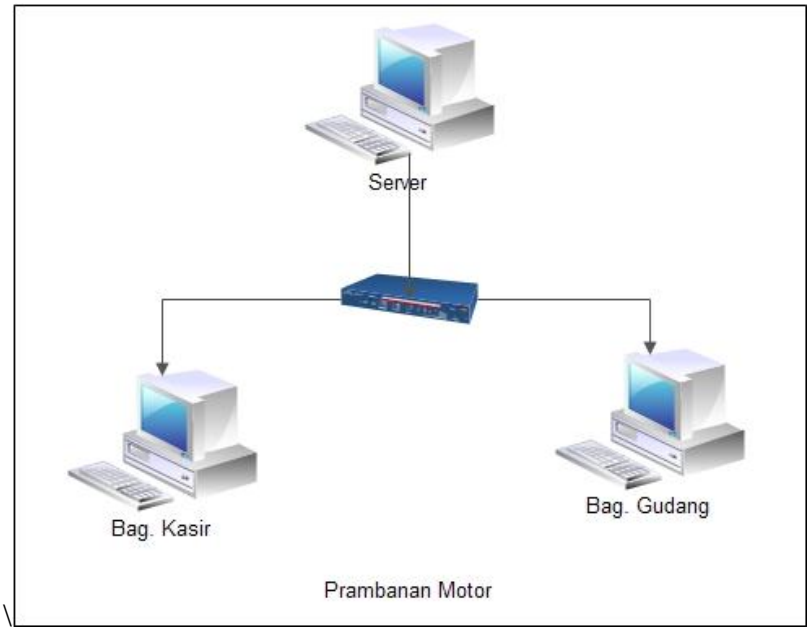
Kode Mekanik IDM-0004
Nama Mekanik burhanudin

No Transaksi	Tanggal	Total Biaya Jasa
TRS030600006	03-06-2014 00:00:00	20000
	Sub Total	20000
	Total Biaya	40000

Gambar 4.49 Laporan Komisi Mekanik

4.3. Perancangan Arsitektur Jaringan

Arsitektur Jaringan *yang* digunakan dalam perancangan sistem ini menggunakan sistem *client-server*, sistem *client-server* mempunyai dua komponen utama yaitu computer *server* dan computer *client. Sercver* merupakan computer induk yang melakukan pemrosesan terbanyak untuk memenuhi permintaan-permintaan dari computer client dan bertindak sebagai *server database* yang menyimpan data. *Client* yaitu computer yang melakukan pengiriman permintaan-permintaan data pada *server* kemudian menampilkan data tersebut pada *interface* aplikasi yang dimilikinya.



Gambar 4.50 Arsitektur jaringan yang diusulkan

Deskripsi :

Topologi yang penulis gunakan menggunakan topologi star dengan alasan jika salah satu computer *client* rusak, maka computer yang lain masih dapat digunakan.

4.4 Implementasi

Implementasi adalah suatu proses penerapan rancangan program yang telah dibuat kedalam sebuah aplikasi pemrograman sesuai dengan tujuan yang diharapkan dari program aplikasi tersebut. Kegiatan implementasi dilakukan dengan dasar yang telah direncanakan dalam tahap perancangan. Implementasi prototype aplikasi ini menggunakan bahasa pemprograman Java dengan JDK 7.4 (*Java Development Kit* 7.4) dan menggunakan *tools Netbeans IDE 7.4.* Basis data yang digunakan adalah *Mysql.* Implementasi dan pengujian sepenuhnya hanya dilakukan di perangakat PC dengan sistem Operasi *Microsoft Windows 7*.

4.4.1. Batasan Implementasi (optional)

Dalam implementasi perangkat lunak tentunya dibutuhkan dukungan perangkat lunak dan perangkat keras yang baik agar mendukung saat penulisan program atau *coding*. Batasan dalam implementasi perangkat lunak ini sebagai berikut :

1. Sistem informasi ini tidak membahas proses perbaikan (*maintenance).*
2. Sistem informasi ini hanya membahas transaksi jasa servis, transaksi pembelian, transaksi penjualan serta laporan transaksinya.
3. Sistem informasi ini tidak membahas laporan keuangan, seperti laporan buku besar atau laporan laba-rugi.

4.4.2. Implementasi Perangkat Lunak

Pada implemenasi perangkat lunak ini, penulis menggunakan perangkat lunak sebagai berikut :

1. XAMPP versi 1.8.1 (database *Mysql)*
2. IDE *Netbeans* 7.4
3. JDK 7.4 (*Java Development kit 7.4*)
4. *Microsoft Windows 7.*

4.4.3. Implementasi Perangkat Keras

Perangkat Keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus dipenuhi adalah sebagai berikut :

1. Processor AMD x240 dual core atau diatasnya
2. Memory DDR 1 Gb
3. Harddisk 40Gb
4. Monitor, Mouse dan Keyboard.

4.4.4. Implementasi Basis Data (Sintaks SQL)

Pada implementasi Basis Data menggunakan bahasa SQL, dimana aplikasi pemrograman yang dipakai adalah MySQL, implementasi basis data dalam MySQL adalah sebagai berikut :

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
```

```
--
-- Host: 127.0.0.1
-- Generation Time: Jun 05, 2014 at 03:48 PM
-- Server version: 5.5.27
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET
@OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_C
LIENT */;
/*!40101 SET
@OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET
_RESULTS */;
/*!40101 SET
@OLD_COLLATION_CONNECTION=@@COLLATION_CONN
ECTION */;
/*!40101 SET NAMES utf8 */;
```

```
--
-- Database: `dbprambanan`
--

-- --------------------------------------------------------

--
-- Table structure for table `jenisitem`
--

CREATE TABLE IF NOT EXISTS `jenisitem` (
  `kode` varchar(32) NOT NULL DEFAULT '',
  `keterangan` varchar(32) NOT NULL,
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `satuanitem`
--
```

```
CREATE TABLE IF NOT EXISTS `satuanitem` (
  `kode` varchar(32) NOT NULL DEFAULT '',
  `keterangan` varchar(32) NOT NULL,
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `temptitem`
--

CREATE TABLE IF NOT EXISTS `temptitem` (
  `Kode_Item` varchar(32) NOT NULL,
  `Tipe_Item` varchar(32) NOT NULL,
  `Jenis` varchar(32) NOT NULL,
  `Stok` varchar(32) NOT NULL,
  `Satuan` varchar(32) NOT NULL,
  `Harga_Pokok` varchar(32) NOT NULL,
  `Harga_Jual` varchar(32) NOT NULL,
```

```
  `komisi` varchar(32) DEFAULT NULL,
  `nominal` varchar(32) DEFAULT NULL,
  `Keterangan` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`Kode_Item`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `temptitemjasa`
--

CREATE TABLE IF NOT EXISTS `temptitemjasa` (
  `Kode_Item` varchar(32) NOT NULL,
  `Tipe_Item` varchar(32) NOT NULL,
  `Jenis` varchar(32) NOT NULL,
  `Stok` varchar(32) NOT NULL,
  `Satuan` varchar(32) NOT NULL,
  `Harga_Pokok` varchar(32) NOT NULL,
  `Harga_Jual` varchar(32) NOT NULL,
  `komisi` varchar(32) DEFAULT NULL,
```

```
  `nominal` varchar(32) DEFAULT NULL,
  `Keterangan` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`Kode_Item`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `titem`
--

CREATE TABLE IF NOT EXISTS `titem` (
  `Kode_Item` varchar(32) NOT NULL,
  `Tipe_Item` varchar(32) NOT NULL,
  `Nama_Item` varchar(32) NOT NULL,
  `Jenis` varchar(32) NOT NULL,
  `Stok` varchar(32) DEFAULT NULL,
  `Satuan` varchar(32) NOT NULL,
  `Harga_Pokok` varchar(32) DEFAULT NULL,
  `Harga_Jual` varchar(32) NOT NULL,
  `komisi` varchar(1) DEFAULT NULL,
```

```
  `nominal` varchar(32) DEFAULT NULL,
  `Keterangan` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`Kode_Item`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `titemkeluar`
--

CREATE TABLE IF NOT EXISTS `titemkeluar` (
  `No_Transaksi` varchar(32) NOT NULL,
  `Tanggal` varchar(32) NOT NULL,
  `Jumlah_Item` varchar(32) NOT NULL,
  `Total_Harga` varchar(32) NOT NULL,
  `Keterangan` varchar(32) NOT NULL,
  `Status` varchar(32) NOT NULL,
  PRIMARY KEY (`No_Transaksi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
-- --------------------------------------------------------

--
-- Table structure for table `titemmasuk`
--

CREATE TABLE IF NOT EXISTS `titemmasuk` (
  `No_Transaksi` varchar(32) NOT NULL,
  `Tanggal` varchar(32) NOT NULL,
  `Jumlah_Item` varchar(32) NOT NULL,
  `Total_Harga` varchar(32) NOT NULL,
  `Keterangan` varchar(32) NOT NULL,
  `Status` varchar(32) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `tkendaraan`
--
```

```
CREATE TABLE IF NOT EXISTS `tkendaraan` (
  `No_Polisi` varchar(32) NOT NULL DEFAULT '',
  `Pemilik` varchar(32) NOT NULL,
  `Alamat` varchar(32) NOT NULL,
  `Merk` varchar(32) NOT NULL,
  `Tipe` varchar(32) DEFAULT NULL,
  `Jenis` varchar(32) DEFAULT NULL,
  `Tahun` varchar(32) DEFAULT NULL,
  `Silinder` varchar(32) DEFAULT NULL,
  `Warna` varchar(32) DEFAULT NULL,
  `No_Rangka` varchar(32) DEFAULT NULL,
  `No_Mesin` varchar(32) DEFAULT NULL,
  `Keterangan` varchar(32) DEFAULT NULL,
  PRIMARY KEY (`No_Polisi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `tkomisimekanik`
--
```

```
CREATE TABLE IF NOT EXISTS `tkomisimekanik` (
  `No_Transaksi` varchar(32) NOT NULL,
  `Tanggal` varchar(32) NOT NULL,
  `Mekanik` varchar(32) NOT NULL,
  `ID_Pelanggan` varchar(32) NOT NULL,
  `No_Polisi` varchar(32) NOT NULL,
  `Total_Komisi` varchar(32) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `ttransaksibeli`
--

CREATE TABLE IF NOT EXISTS `ttransaksibeli` (
  `No_Transaksi` varchar(32) NOT NULL,
  `No_Pesanan` varchar(32) NOT NULL,
  `Tanggal` varchar(32) NOT NULL,
  `ID_Supplier` varchar(32) NOT NULL,
```

```
  `Jml_Item` varchar(32) NOT NULL,
  `Total_Transaksi` varchar(32) NOT NULL,
  `Jml_Bayar` varchar(32) NOT NULL,
  `Sisa_Bayar` varchar(32) NOT NULL,
  `Keterangan` varchar(32) NOT NULL,
  `Status` varchar(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `ttransaksijual`
--

CREATE TABLE IF NOT EXISTS `ttransaksijual` (
  `No_Transaksi` varchar(32) NOT NULL,
  `No_Pesanan` varchar(32) NOT NULL,
  `Tanggal` varchar(32) NOT NULL,
  `ID_Pelanggan` varchar(32) NOT NULL,
  `Jml_Item` varchar(32) NOT NULL,
  `Total_Transaksi` varchar(32) NOT NULL,
```

```
  `Jml_Bayar` varchar(32) NOT NULL,
  `Sisa_Bayar` varchar(32) NOT NULL,
  `Keterangan` varchar(32) NOT NULL,
  `Status` varchar(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `ttransaksiservis`
--

CREATE TABLE IF NOT EXISTS `ttransaksiservis` (
  `No_Transaksi` varchar(32) NOT NULL,
  `Tanggal` varchar(32) NOT NULL,
  `ID_Pelanggan` varchar(32) NOT NULL,
  `User` varchar(32) NOT NULL,
  `No_Polisi` varchar(32) NOT NULL,
  `Total_Jasa` varchar(32) NOT NULL,
  `Jml_Item` varchar(32) NOT NULL,
  `Total_Item` varchar(32) NOT NULL,
```

```
  `Km_Sekarang` varchar(32) NOT NULL,
  `Km_Berikutnya` varchar(32) NOT NULL,
  `Status_Servis` varchar(32) NOT NULL,
  `Keluhan_Servis` varchar(32) NOT NULL,
  `Mekanik1` varchar(32) NOT NULL,
  `Mekanik2` varchar(32) NOT NULL,
  `Mekanik3` varchar(32) NOT NULL,
  `Subtotal_Servis` varchar(32) NOT NULL,
  `Jml_Bayar` varchar(32) NOT NULL,
  `Sisa_Bayar` varchar(32) NOT NULL,
  `Status` varchar(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `tuser`
--

CREATE TABLE IF NOT EXISTS `tuser` (
  `ID_User` varchar(8) NOT NULL,
```

```
  `Nama` varchar(32) NOT NULL,
  `Bagian` varchar(32) NOT NULL,
  `Password` varchar(150) NOT NULL,
  `Konfirmasi` varchar(150) NOT NULL,
  PRIMARY KEY (`ID_User`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*!40101 SET
CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIE
NT */;
/*!40101 SET
CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RE
SULTS */;
/*!40101 SET
COLLATION_CONNECTION=@OLD_COLLATION_CONNECT
ION */;
```

4.4.5. Implementasi Antar Muka

Implementasi dilakukan dengan sebuah file project yang berektensi [dot].java yang ada dalam *Tools* netbeans yang didalamnya terdapat form-form untuk pembuatan suatu aplikasi.

4.4.5.1. Implementasi Login

Merupakan halaman yang tampil pertama kali saat program dijalankan , halaman login ini merupakan kunci utama untuk masuk ke program, apabila *user* gagal melakukan login , Menu utama dari aplikasi yang berisi sistem informasi bengkel tidak akan muncul.

Table 4.9 Implementasi Halaman Login

Sub Menu	Deskripsi	Nama File
Menu Login	Tampilan Menu Login berisi inputan *username* dan *password* login	MenuLogin.Java
Masuk	Tombol Masuk Sistem	jBmasuk
Batal	Tombol Keluar Sistem	jBbatal

4.4.5.2. Implementasi Halaman Utama

Form halaman utama merupakan pusat dari semua halaman form. Form ini berisi menu-menu utama yang dapat mengakses dan menjadi penghubung antara pengguna dengan sistem. Berikut implemtasi struktur menu pada halaman utama Sistem bengkel :

Table 4.10 Implementasi Halaman Utama

Menu	Deskripsi	Nama File
Menu Utama	Menu Utama hanya berisi informasi nama bengkel dan alamat	jPMenuUtama
Master Data	Menu Master Data berisi menu user, menu master item,menu master pelanggan,menu master kendarran,menu master supplier,menu master mekanik	JPMasterData
Pembelian	Menu Pembelian berisi menu pesanan pembelian, menu transaksi pembelian, menu retur pembelian, item masuk	jPpembelian
Penjualan	Menu Penjualan berisi menu pesanan	jPpenjualan

	penjualan,menu transaksi penjualan,menu transaksi servis,menu retur penjualan,menu keluar	
Laporan	Menu ini berisi laporan setiap transaksi dan daftar item yang sudah ada	jPlaporan

4.4.5.3. Implementasi Halaman Berdasarkan Sub Menu Master Data

Form sub menu Master Data ini merupakan halaman yang berisi informasi data item barang seperti data suku cadang, data kendaraan, data supplier, data user , data , data pelanggan

Table 4.11 Implementasi Halaman Berdasarkan Sub Menu Master Data

Menu	Deskripsi	Nama File
Master Item	Berisi informasi data barang dan jasa , menambah , mengurangi, mengedit	MasterItem.java
Master Kendaraan	Berisi informasi data kendaraan, menambah , mengurangi, mengedit	MasterKendaraan.java
Master Pelanggan	Berisi informasi data pelanggan , menambah , mengurangi, mengedit	MasterPelanggan.java
Master Supplier	Berisi informasi data supplier , menambah , mengurangi, mengedit	MasterSupplier.java
Master User	Berisi informasi data user berfungsi untuk login sistem , menambah , mengurangi, mengedit	MasterUser.java
Master Mekanik	Berisi informasi data mekanik , menambah , mengurangi, mengedit	MasterMekanik.java

4.4.5.4. Implementasi Halaman Berdasarkan Sub Menu Pembelian

Sub Menu pembelian berfungsi untuk menambah, mengedit , mencari transaksi pembelian.

Table 4.12 Implementasi Halaman Berdasarkan Sub Menu Pembelian

Menu	Deskripsi	Nama File
Menu Pesanan Pembelian	Form untuk mengisi transaksi pesanan pembelian	PesananPembelian.java
Menu Transaksi Pembelian	Form untuk mengisi transaksi pembelian	TransaksiPembelian.java
Menu Retur pembelian	Form untuk mengisi retur pembelian	ReturPembelian.java
Menu Item Masuk	Form untuk mengisi item masuk	ItemMasuk.java

4.4.5.5. Implementasi Halaman Berdasarkan Sub Menu Penjualan

Sub Menu Penjualan berfungsi untuk menambah, mengedit atau mencari transaksi penjualan, data pesanan penjualan sebelumnya bisa diambil langsung otomatis.

Table 4.13 Implementasi Halaman Berdasarkan Sub Menu Penjualan

Menu	Deskripsi	Nama File
Menu Pesanan Penjualan	Form untuk mengisi data pesanan penjualan	PesananPenjualan.java
Menu Transaksi Penjualan	Form untuk mengisi transaksi penjualan	TransaksiPenjualan.java
Menu Transaksi Servis	Form untuk mengisi transaksi servis	TransaksiServis.java
Menu Retur Penjualan	Form untuk mengisi retur penjualan	ReturPenjualan.java
Menu Item Keluar	Form untuk mengisi item keluar	ItemKeluar.java

4.4.5.6. Implementasi Halaman Berdasarkan Sub Menu Laporan

Sub Menu Laporan berisi laporan berbagai transaksi ataupun laporan data item , data kendaraan, data supplier.

Table 4.14 Implementasi Halaman Berdasarkan Sub Menu Laporan

Menu	Deskripsi	Nama File
Laporan Master Barang	Berisi Laporan Daftar Item, Laporan Daftar Pelanggan, Laporan Daftar Supplier	laporanmasterbarang.java
Laporan Pembelian	Berisi Laporan Pesanan , Laporan transaksi Pembelian, Transaksi Retur	laporanpembelian.java
Laporan Penjualan	Berisi Laporan Transaksi Pesanan, Transaksi Penjualan, Transaksi Servis, Transaksi Retur	Laporanpenjualan.java
Laporan Lain lain	Berisi Laporan Stok Barang, Laporan Stok Barang By Kode, Laporan komisi Mekanik	Laporanlainlain.java

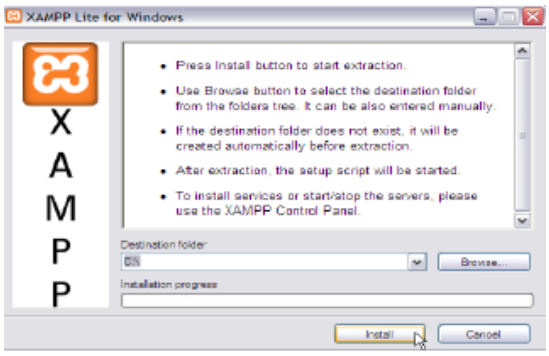
4.4.6. Implementasi Instalasi Program

Implementasi instalasi program yang dilakukan penulis untuk menjalankan sistem yang telah dibuat sebagai berikut :

4.4.6.1. Xampp 1.8.1

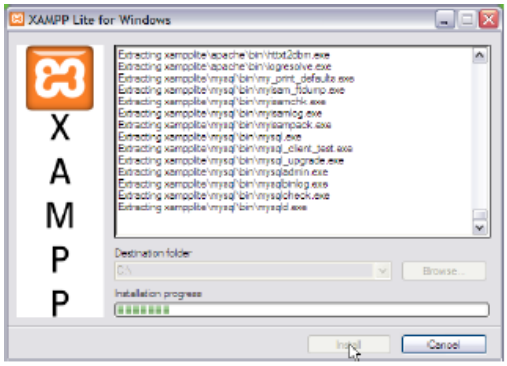
Merupakan perangkat lunak yang mendukung banyak sistem operasi yang fungsinya sebagai server yang berdiri sendiri (*localhost*), yang terdiri atas program *Apache HTTP server* , *MySQL database*, dan penerjemah bahasa pemrograman *PHP* dan *Perl*. Instalasi programnya sebagai berikut :

1. Klik 2 kali file (dot).exe xampp kemudian muncul tampilan seperti dibawah ini. Pilih folder unutk tempat install, disini penulis menggunakan drive C. Kemudian install.



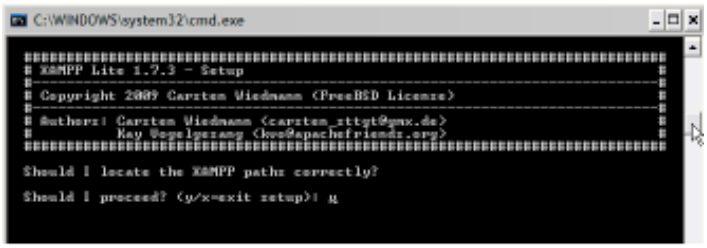
Gambar 4.51 Proses Instalasi Xampp pertama

2. Proses instalasi sedang berjalan



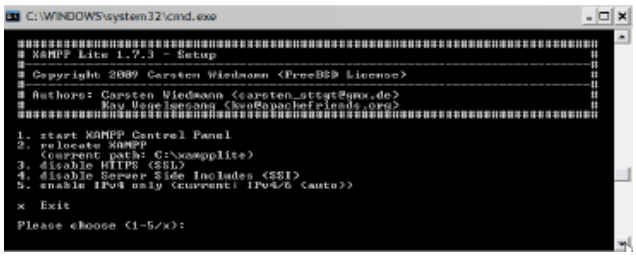
Gambar 4.52 Proses Instalasi Xampp kedua

3. Proses instalasi selesai kemudian muncul dialog seperti gambar dibawah ini, pilih y dan enter.



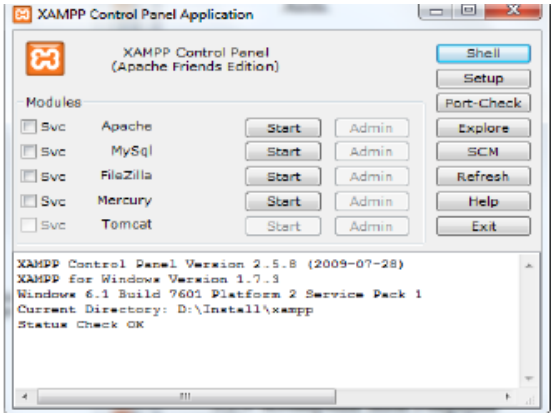
Gambar 4.53 Proses Instalasi Xampp ketiga

4. Untuk keluar dari menu instalasi xampp gunakan huruf X



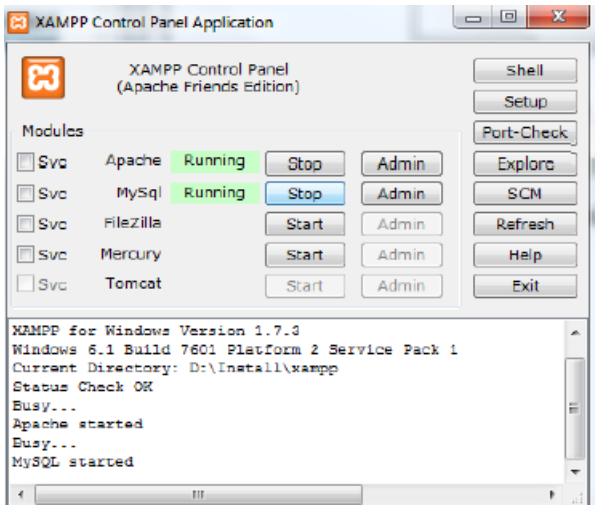
Gambar 4.54 Proses Instalasi Xampp keempat

5. Xampp sudah terinstal dikomputer dan control panel xampp muncul seperti gambar dibawah ini.



Gambar 4.55 Kontrol Panel Xampp

6. Kemudian untuk menjalankan program yang dibutuhkan untuk menjalankan sistem informasi bengkel prambanan motor jalankan *Apache server* dan *MySql* dengan meng-klik tombol star.



Gambar 4.56 Kontrol Panel Xampp Aktif

4.4.6.2. Instalasi JRE

Pertama yang harus dilakukan adalah membuka file instalasi JRE yang berekstensi.exe lalu double click untuk menginstalnya.



Gambar 4.57 Icons Instalasi JRE

Maka akan terlihat tampilan awal proses instalasi seperti pada gambar berikut :



Gambar 4.58 Tampilan Awal Instalasi JRE

Klik tombol **Instal**, maka akan terlihat tampilan ketika proses instalasi sedang berlangsung, seperti gambar 4.32



Gambar 4.59 Tampilan Proses Instalasi JRE

Setelah proses instalasi selesai maka akan muncul tampilan seperti berikut, lalu klik tombol **Close.**



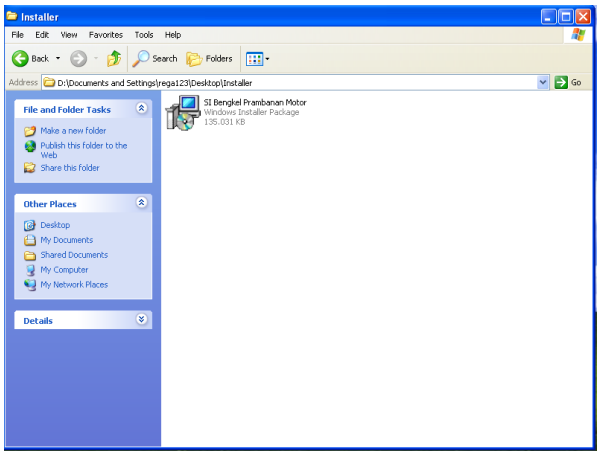
Gambar 4.60 Tampilan Akhir Instalasi JRE

Sekarang, dalam komputer kita telah terinstall jre. Maka bila kita sudah bisa menjalankan file-file hasil kompilasi dari bahasa pemrograman java.

4.4.6.2. Instalasi Program Prambanan Motor

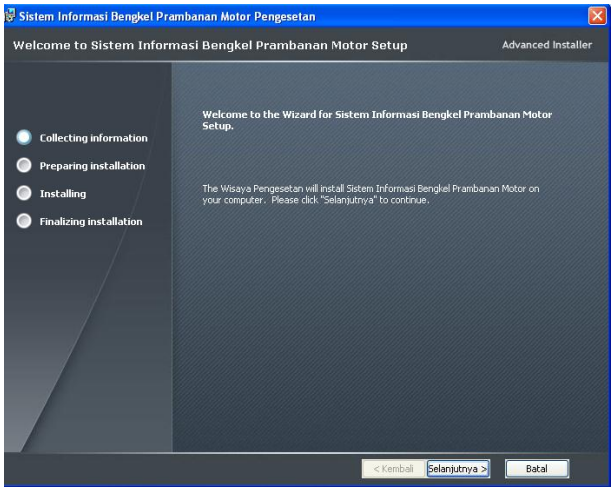
Berikut ini adalah tahapan – tahapan melakukan instalasi program aplikasi sistem informasi pelayanan kesehatan bayi dan balita di posyandu delima bandung :

1. Buka folder instalasi program



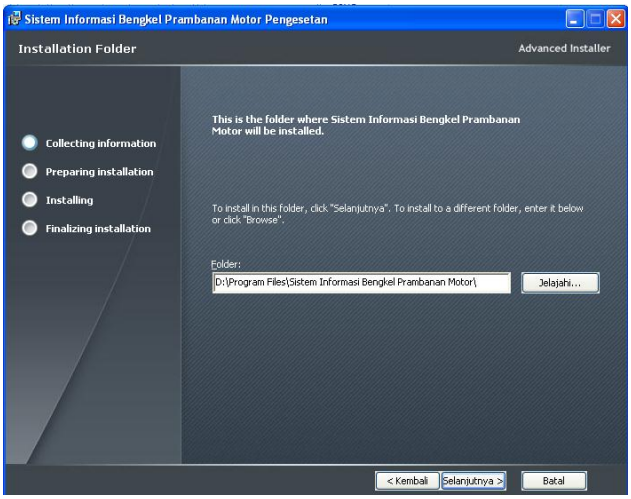
Gambar 4.61 Tampilan program instalasi

2. Tampilan proses intalasi pilih selanjutnya



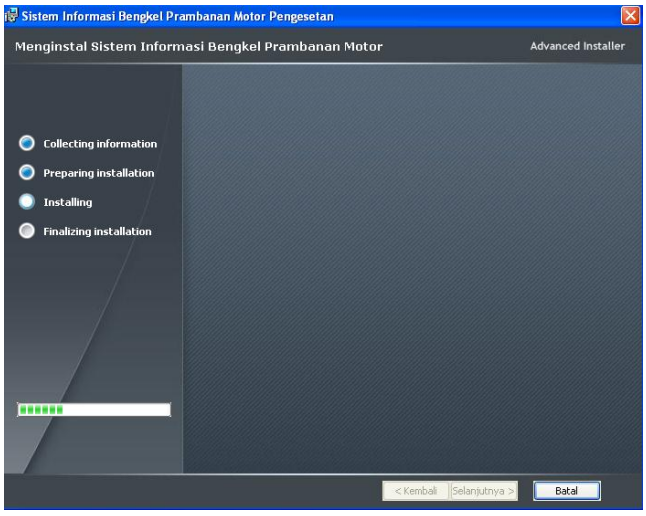
Gambar 4.62 Tampilan proses instalasi program

3. Pemilihan folder tempat program di simpan, pilih selanjutnya



Gambar 4.63 Tampilan penyimpanan program

4. Tunggu Proses instalasi hinga selesai



Gambar 4.64 Tampilan proses instalasi berjalan

4.4.7. Penggunaan Program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program Sistem Informasi yang telah penulis buat. Adapun cara pengoperasiannya adalah sebagai berikut :

1. Form Login

Form login ini berfungsi sebagai penentu bagi pengguna program sistem informasi, sehingga tidak sembarangan orang bisa mengajses program ini.

Hak akses utuk login dibagi menjadi tiga bagian diantaranya :

1. Admin bengkel

   Admin disini sebagai pemilik toko sekaligus *superuser* yang dapat mengakses seluruh fungsi program yang ada

2. Bagian Kasir

   Bagian Kasir bertugas hanya pada transaksi penjualan saja meliputi pemesanan barang, transaksi penjualan , transaksi servis, transaksi retur penjualan, dan transaksi item keluar.

3. Bagian Gudang

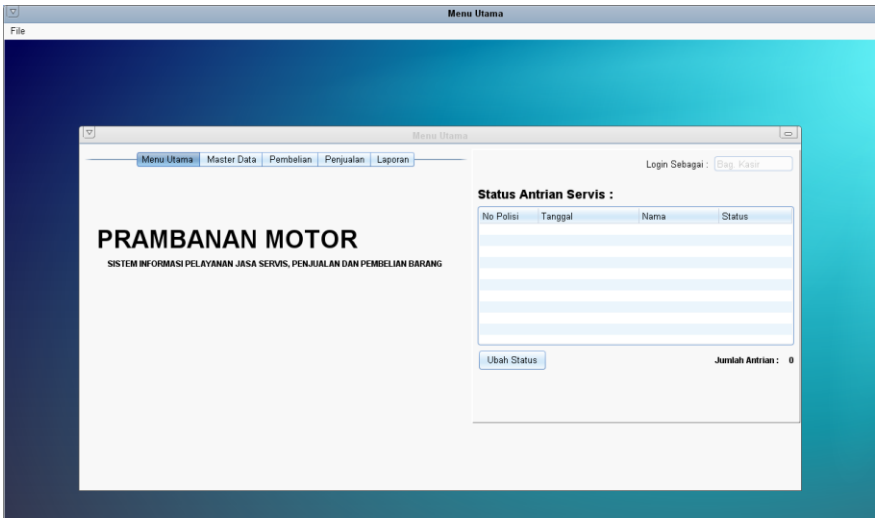
   Bagian gudang hanya bertugas pada transaksi pembelian saja meliputi pemesanan barang, transaksi pembelian, transaksi retur pembelian , transaksi item masuk.



Gambar 4.65 Tampilan Form Login

2. Form Menu Utama

Form Menu Utama adalah form yang muncul pertama kali ketika user melakukan login ke sistem, menu ini hanya berisi informasi nama bengkel , Status antrian kendaraan yang sudah terdaptar dan informasi login.

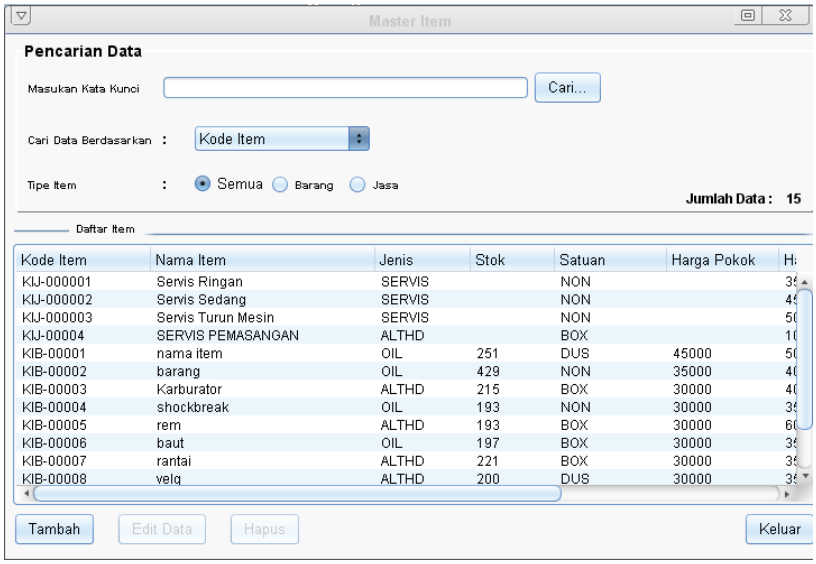


Gambar 4.66 Tampilan Menu Utama

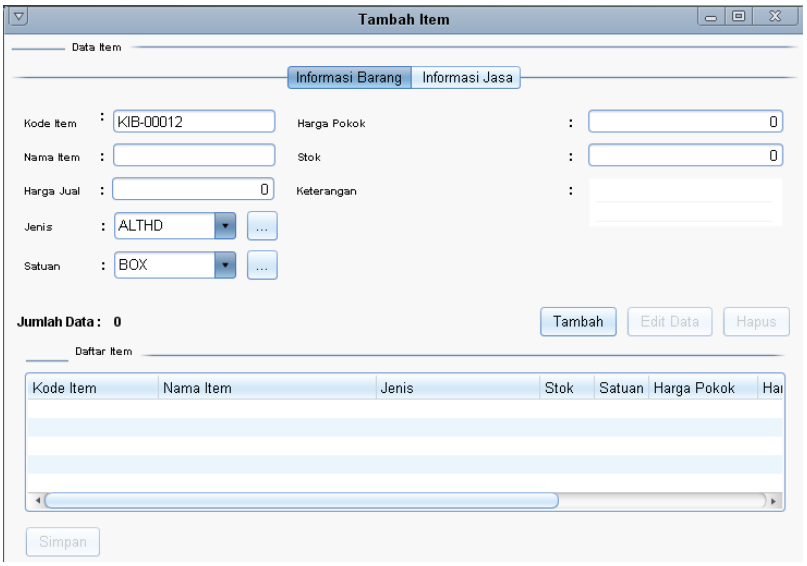
3. Form Master Data

Form Master Data berfungsi sebagai penyimpanan, penambahan, pengeditan, penghapusan data item barang atau jasa, data kendaraan , data pelanggan , data supplier, data mekanik dan data user.

a. Master Item

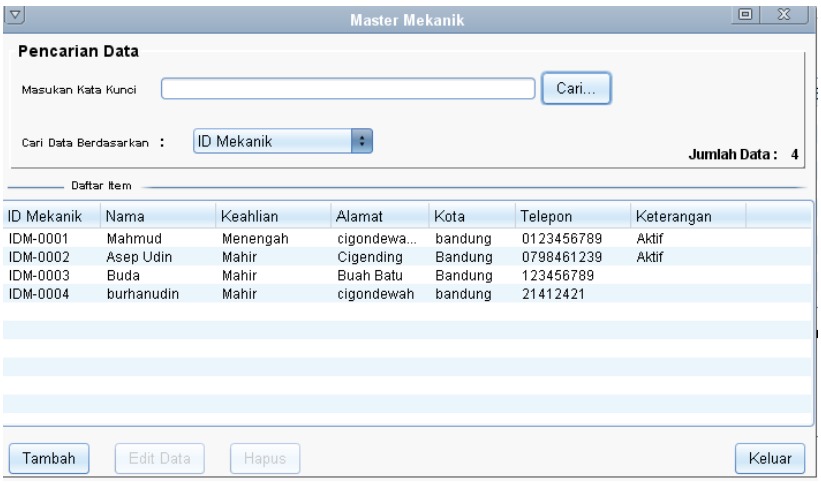


Gambar 4.67 Tampilan Master Data Item



Gambar 4.68 Tampilan Tambah Data Item

b. Master Mekanik

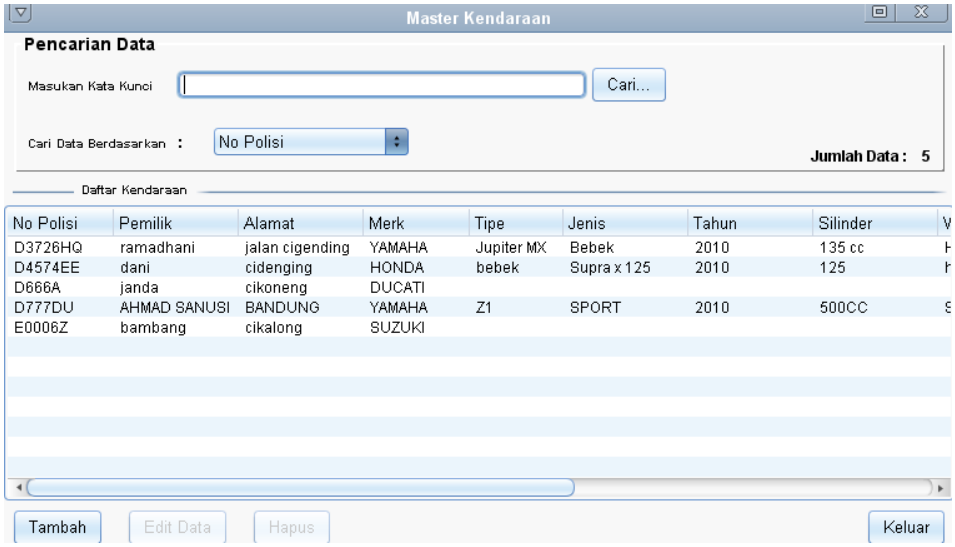


Gambar 4.69 Tampilan Master Mekanik



Gambar 4.70 Tampilan Tambah Mekanik

c. Master Kendaraan

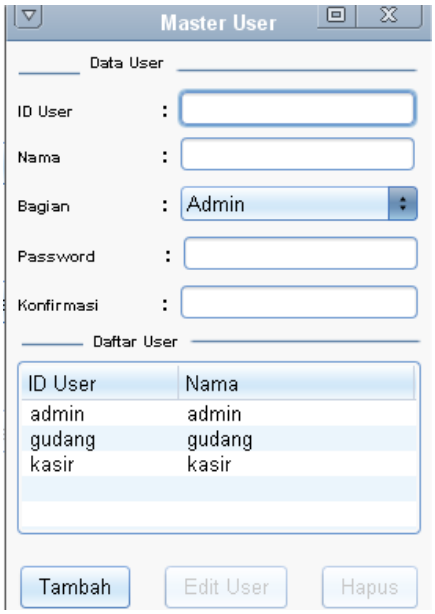


Gambar 4.71 Tampilan Master Kendaraan



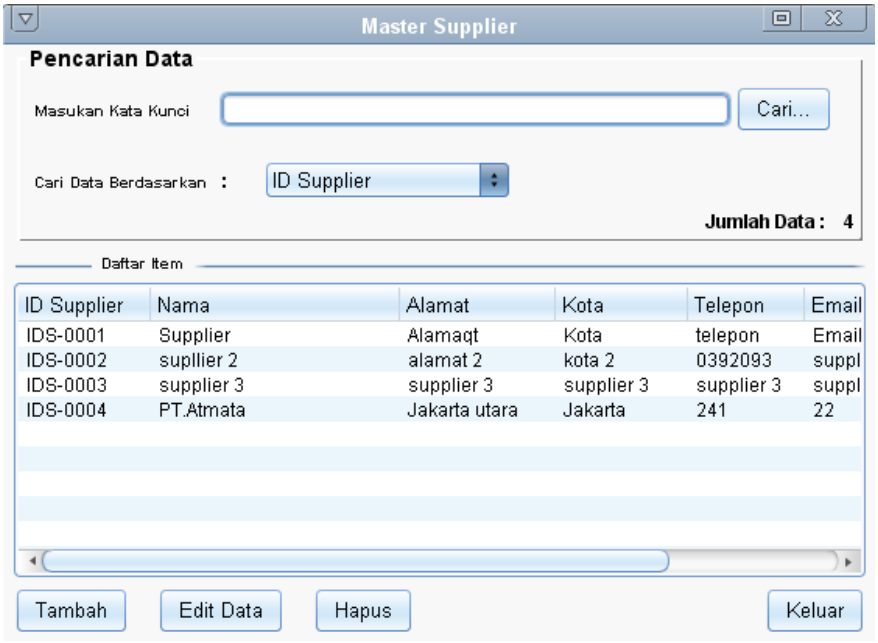
Gambar 4.72 Tampilan Tambah Kendaraan

d. Master User



Gambar 4.73 Tampilan Master User

e. Master Supplier

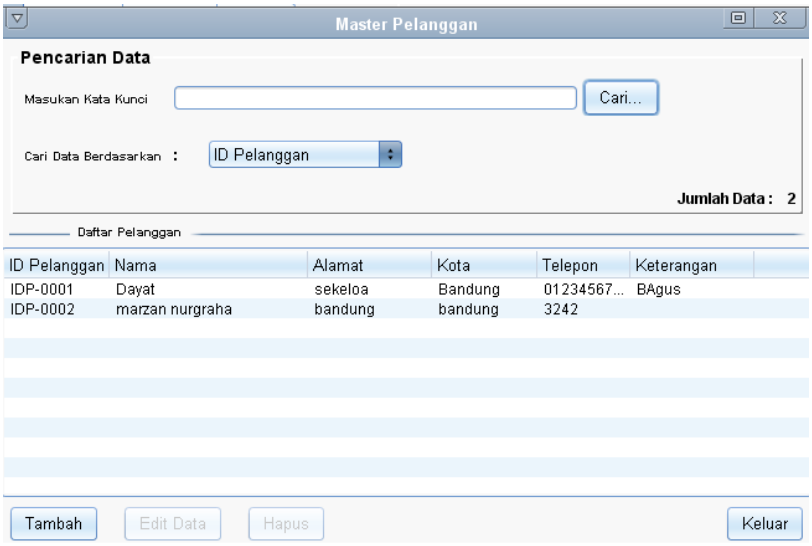


Gambar 4.74 Tampilan Master Supplier



Gambar 4.75 Tampilan Tambah Supplier

f. Master Pelanggan



Gambar 4.76 Tampilan Master Pelanggan

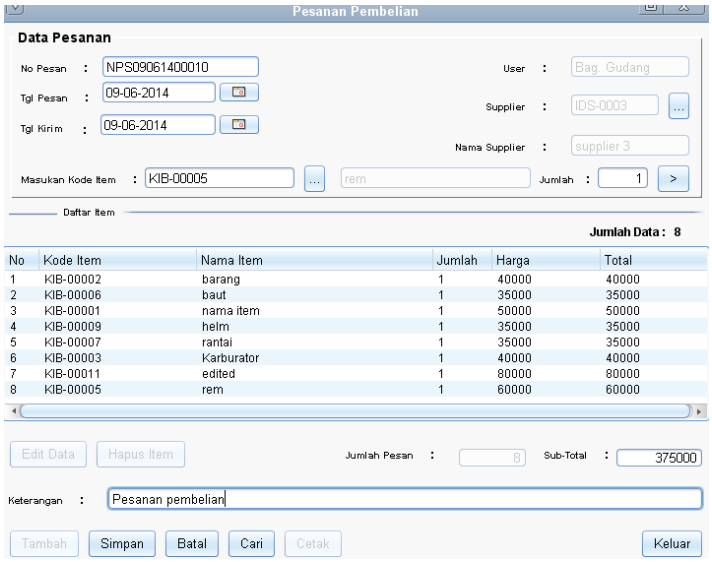


Gambar 4.77 Tampilan Tambah Pelanggan

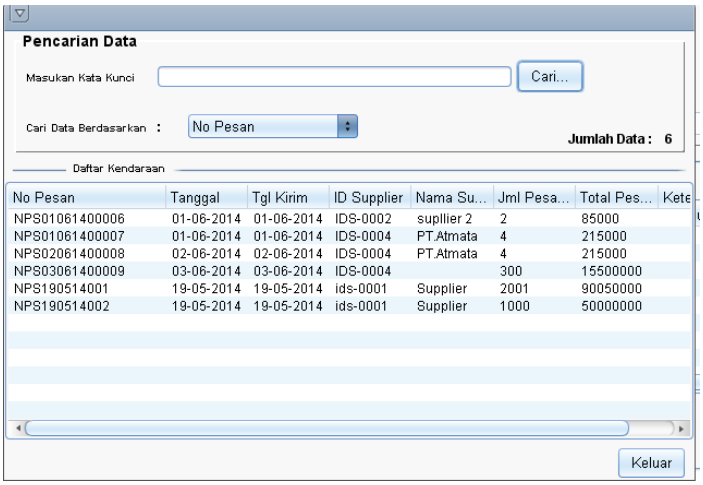
4. Form Pembelian

Form Pembelian berfungsi untuk mengolah seluruh transaksi pembelian yang ada dibengkel prambanan motor, terdiri dari pesanan pembelian, transaksi pembelian, retur pembelian dan item masuk.

a. Pesanan Pembelian

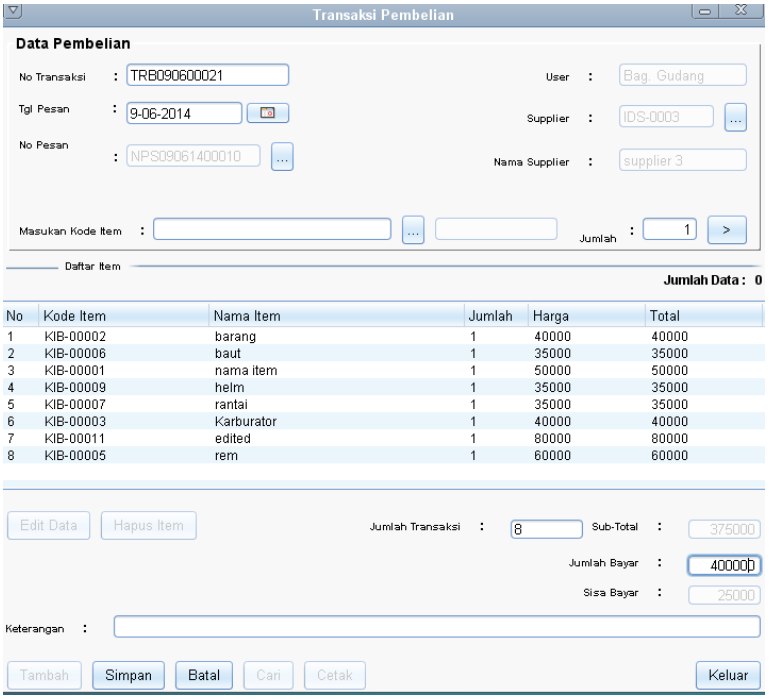


Gambar 4.78 Tampilan Pesanan Pembelian

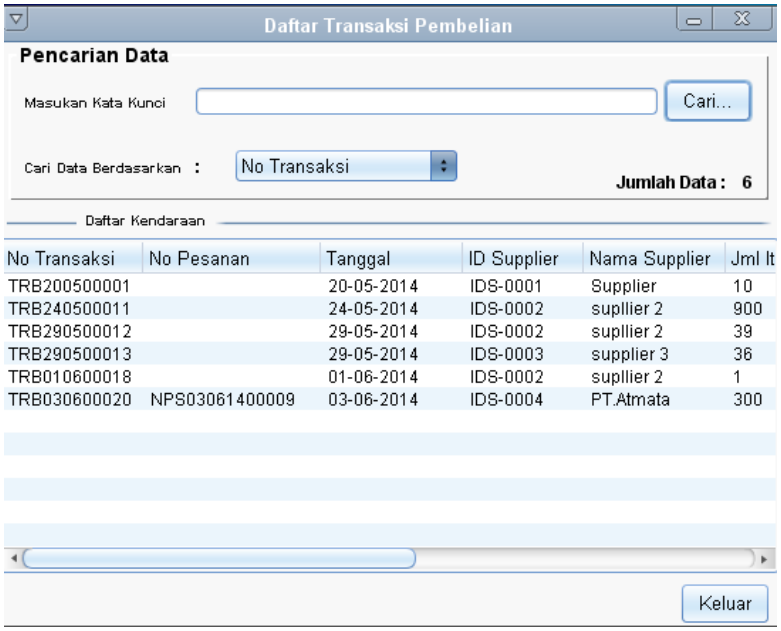


Gambar 4.79 Tampilan Pencarian Pesanan Pembelian

b. Transaksi Pembelian

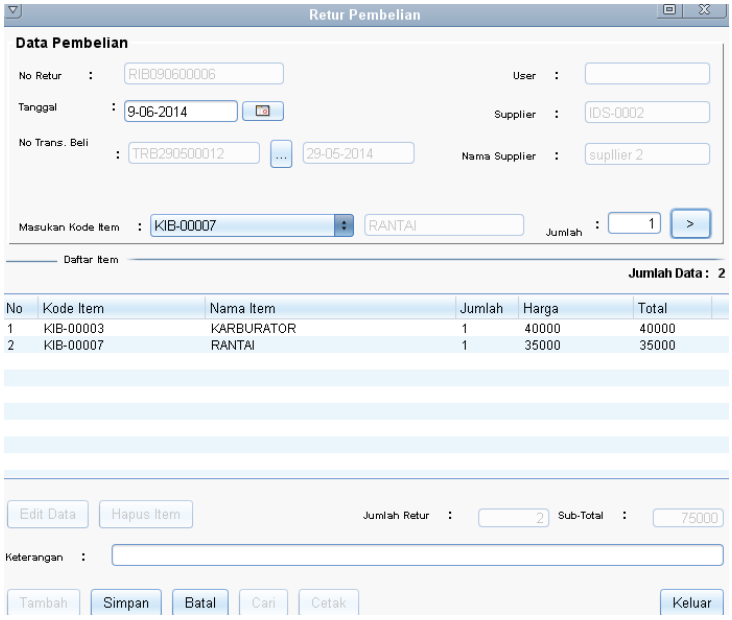


Gambar 4.80 Tampilan Transaksi Pembelian

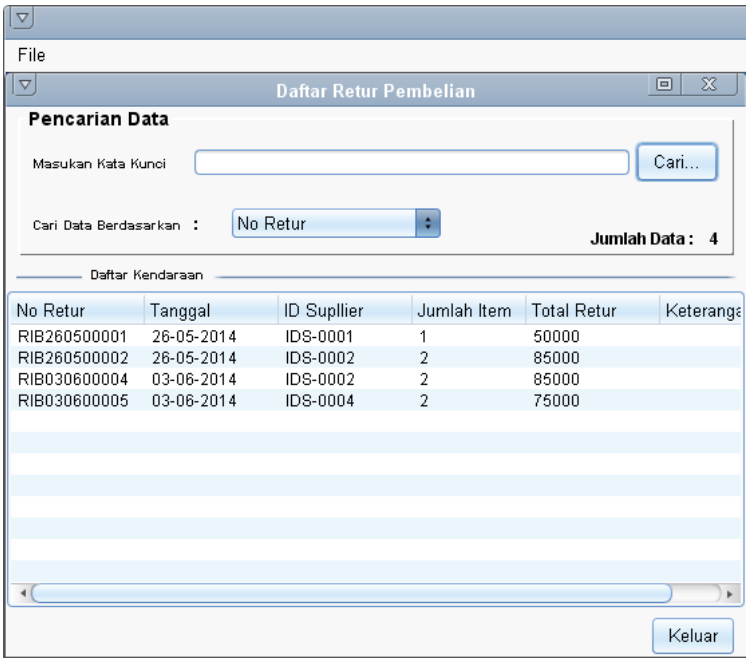


Gambar 4.81 Tampilan Pencarian Transaksi Pembelian

c. Retur Pembelian

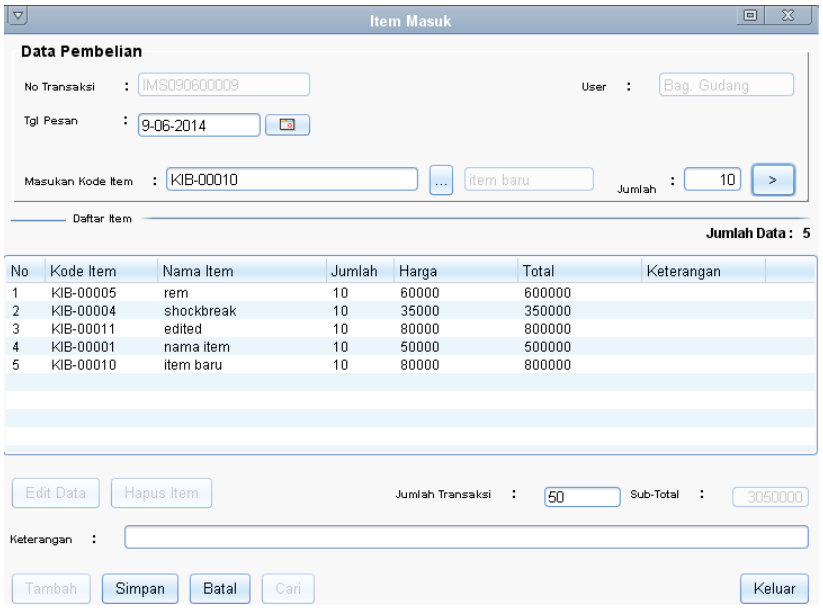


Gambar 4.82 Tampilan Retur Pembelian

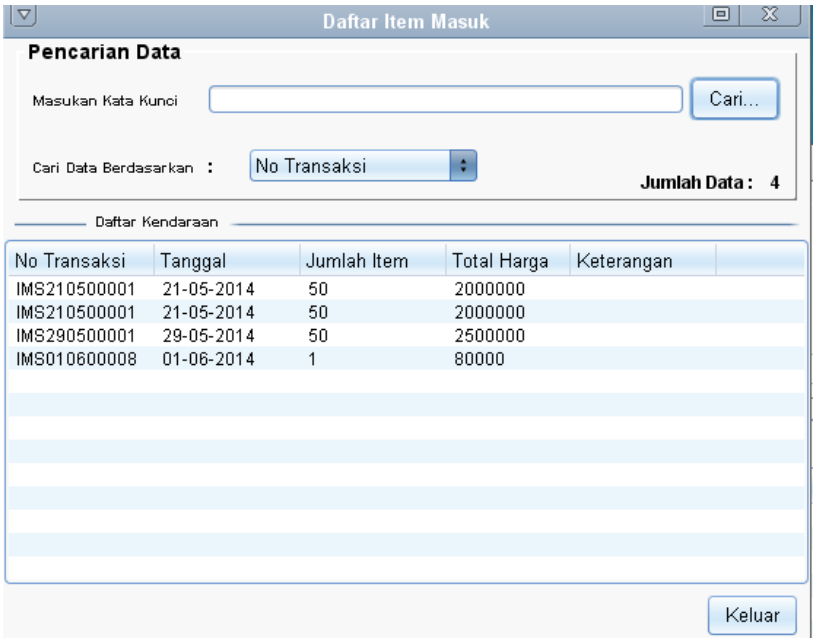


Gambar 4.83 Tampilan Pencarian Retur Pembelian

d. Item Masuk



Gambar 4.84 Tampilan Item Masuk

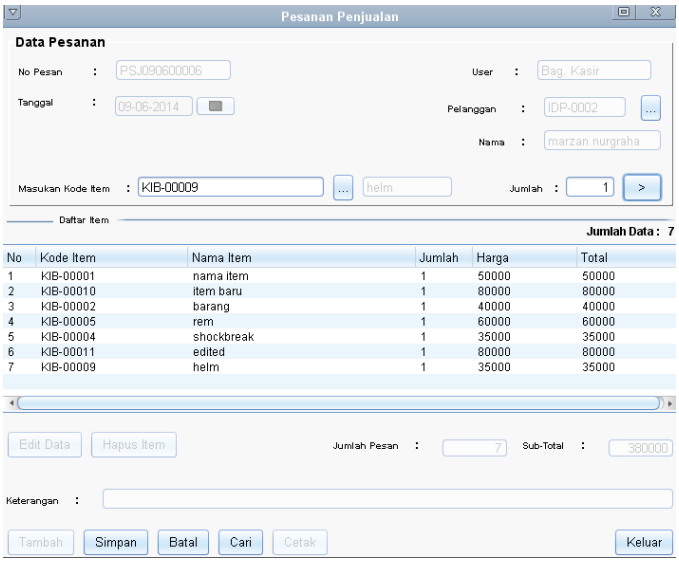


Gambar 4.85 Tampilan Pencarian Item Masuk

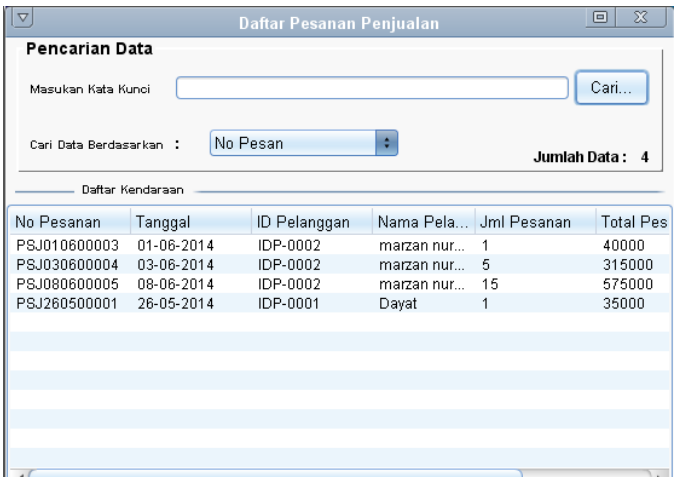
5. Form Penjualan

Form Penjualan berfungsi mengolah seluruh transaksi yang ada di penjualan, meliputi pesanan penjualan , transaksi penjulan, transaksi servis ,transaksi retur dan item keluar.

a. Pesanan Penjualan

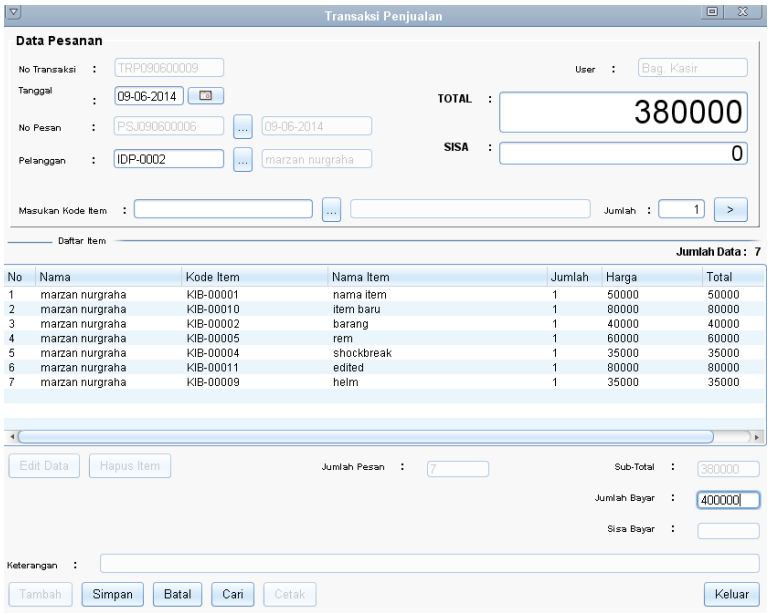


Gambar 4.86 Tampilan Form Penjualan

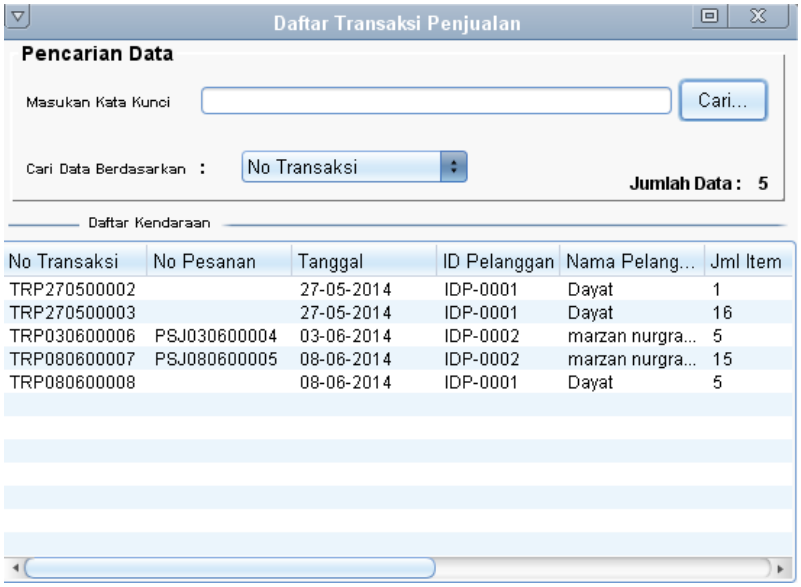


Gambar 4.87 Tampilan Pencarian Form Penjualan

b. Transaksi Penjualan

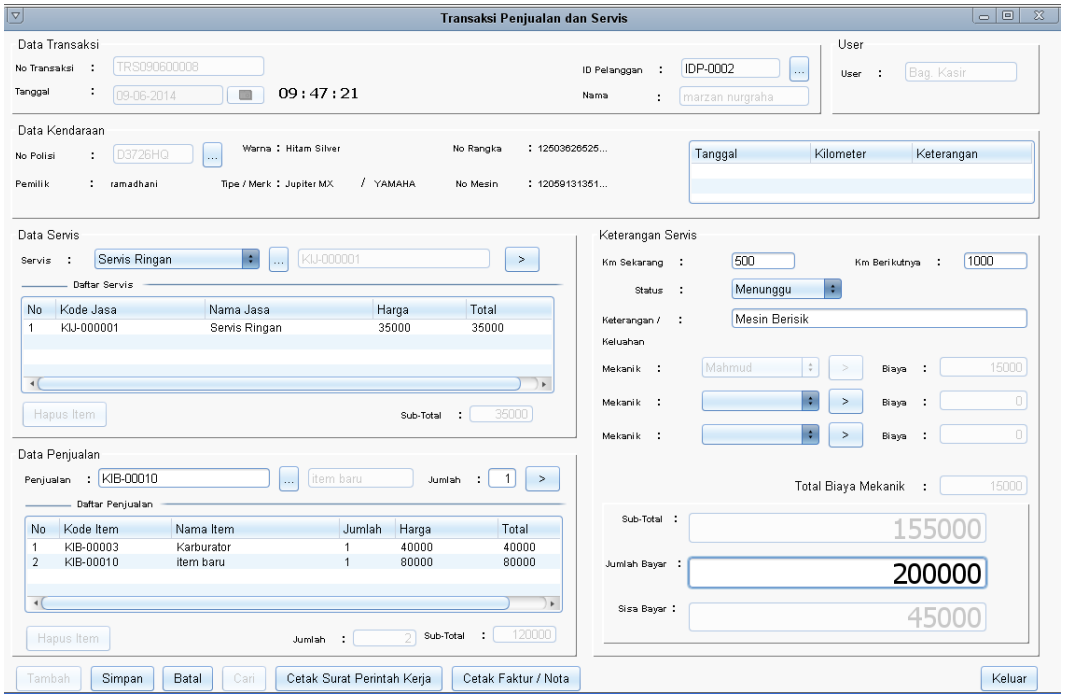


Gambar 4.88 Tampilan Transaksi Penjualan

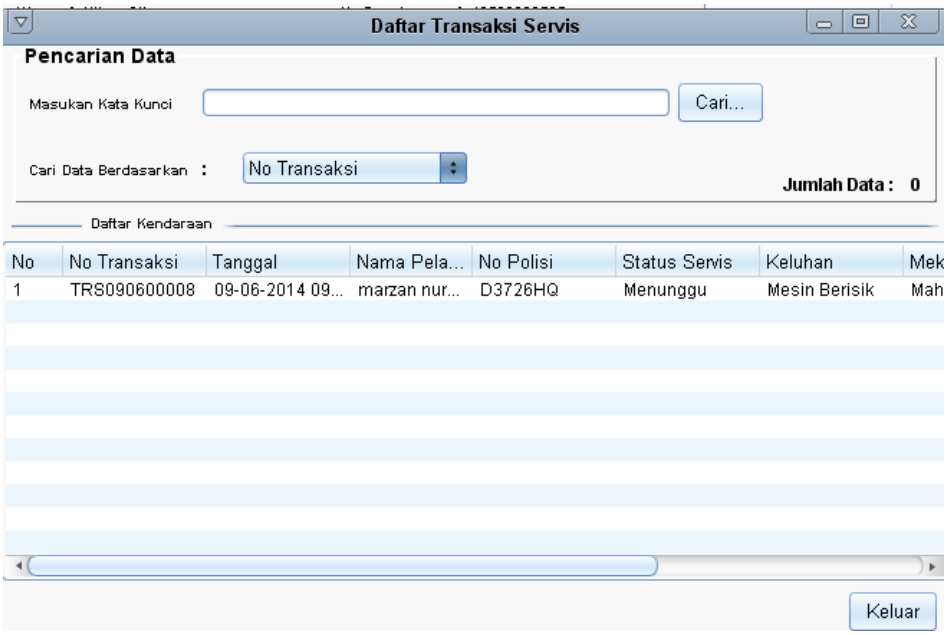


Gambar 4.89 Tampilan Pencarian Transaksi Penjualan

c. Transaksi Servis

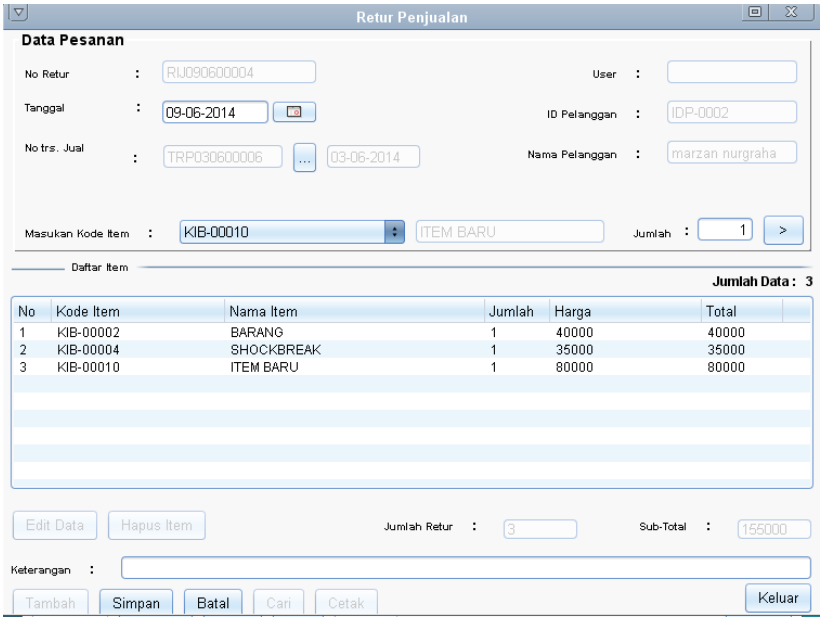


Gambar 4.90 Tampilan Transaksi Servis

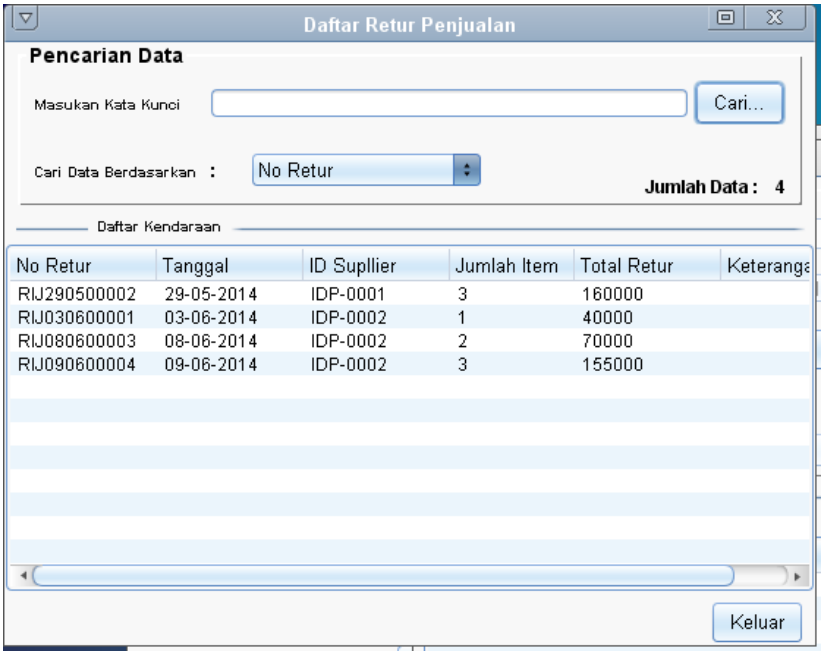


Gambar 4.91 Tampilan Pencarian Transaksi Servis

d. Retur Penjualan

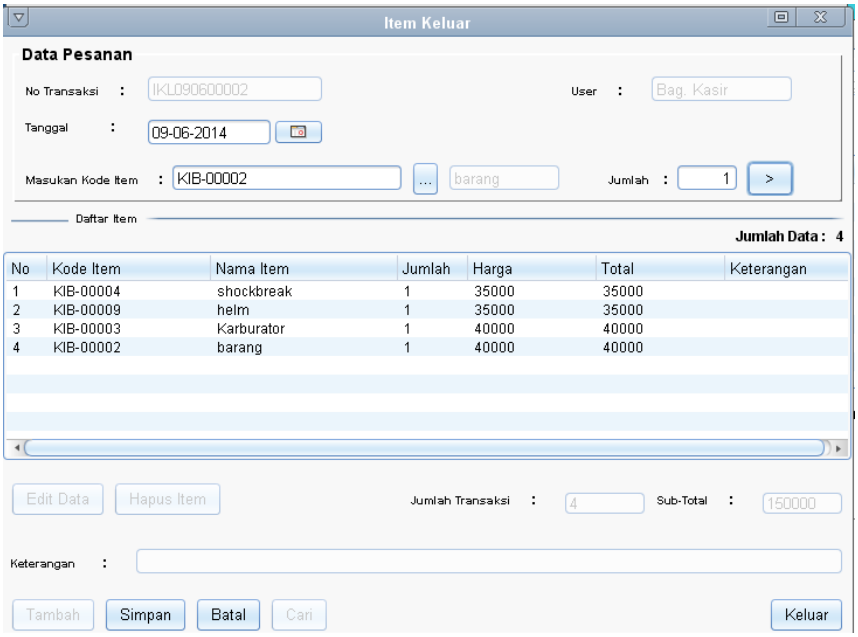


Gambar 4.92 Tampilan Transaksi Retur

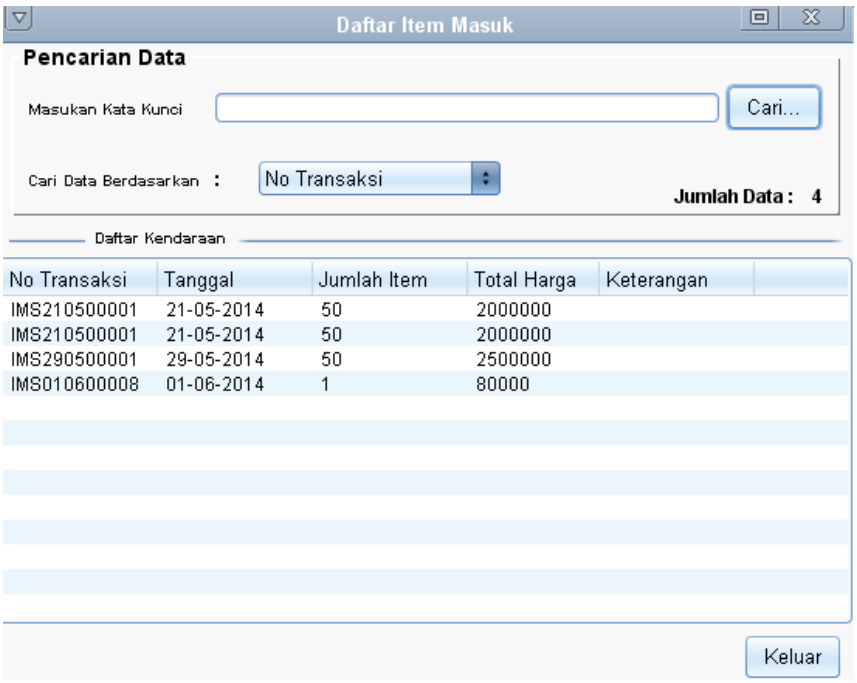


Gambar 4.93 Tampilan Pencarian Transaksi Retur

e. Item Keluar



Gambar 4.94 Tampilan Item Keluar



Gambar 4.95 Tampilan Pencarian Item Keluar

6. Form Laporan

Form laporan berfungsi untuk mencatat setiap transaksi yang sudah dilakukan baik itu pembelian ,penjualan , penambahan master item maupun jasa



Gambar 4.96 Tampilan Laporan Transaksi

4.5. Pengujian

Pengujian perangkat lunak ini menggunakan metode pengujian *Black Box.* Pengujian *Black Box* berfokus pasa persyaratan atau kebutuhan fungsional perangkat lunak yang dibuat.

4.5.1 Rencana Pengujian

Adapun hal-hal yang akan diuji melauli teknik pengujian *Black box* adalah sebagai berikut :

Tabel 4.15 Rencana Pengujian

Hal yang diuji	Pola Pengujian
Login User	Melakukan *Login* ke sistem
Master Data	Menambah, mengedit, mencari dan menghapus
Transaksi Pembelian	Menambah, mengedit, mencari dan menghapus
Transaksi Penjualan	Menambah, mengedit, mencari dan menghapus
Laporan	Menginput data laporan berdasarkan periode atau kode tertentu

4.5.2 Kasus dan Hasil Pengujian

Berikut ini uraian pengujian dengan teknik *Black box* berdasarkan hal yang diuji pada rencana pengujian :

Tabel 4.16 Rencana Pengujian

Hal yang diuji	Skenario Pengujian		Hasil yang diharapkan	Hasil pengujian
Login *user*	1	Input Login (Data Benar)	User masuk ke sistem (Menu Utama)	[✓]Diterima [ ]Ditolak
	2	Input Login (Data Salah)	Muncul peringatan error gagal	[✓]Diterima [ ]Ditolak
Master Data	1	Input data, edit data, hapus data, cari data	Data tersimpan , data berhasil diubah, data berhasil dihapus, data yang dicari sesuai dengan inputan	[✓]Diterima [ ]Ditolak
	2	Input data, edit data, hapus data, cari data tidak sesuai dengan aturan	Muncul peringatan error atau data tidah berhasil di simpan atau diubah ke database	[✓]Diterima [ ]Ditolak
Transaksi Pembelian	1	Input data, edit data, hapus data, cari data	Data tersimpan , data berhasil diubah, data berhasil dihapus, data yang dicari sesuai dengan inputan	[✓]Diterima [ ]Ditolak
	2	Input data, edit data,	Muncul peringatan error	[✓]Diterima

		hapus data, cari data tidak sesuai dengan aturan	atau data tidah berhasil di simpan atau diubah ke database	[ ]Ditolak
Transaksi Penjualan	1	Input data, edit data, hapus data, cari data	Data tersimpan , data berhasil diubah, data berhasil dihapus, data yang dicari sesuai dengan inputan	[✓]Diterima [ ]Ditolak
	2	Input data, edit data, hapus data, cari data tidak sesuai dengan aturan	Muncul peringatan error atau data tidah berhasil di simpan atau diubah ke database	[✓]Diterima [ ]Ditolak
Laporan	1	Input data seuai kategori dari laporan	Data tersimpan , data berhasil diubah, data berhasil dihapus, data yang dicari sesuai dengan inputan	[✓]Diterima [ ]Ditolak
	2	Input kategori tertentu tidak sesuai aturan	Laporan muncul tanpa data	[✓]Diterima [ ]Ditolak

4.6. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian diatas dapat disimpulkan bahwa perangkat lunak Sistem Informasi Pelayanan jasa servis dan jual-beli suku cadang pada bengkel prambanan motor berjalan dengan baik dan mengeluarkan hasil sesuai dengan yang diharapkan.