

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

#### **4.1 Perancangan Sistem**

Pada tahap perancangan ini akan dijelaskan mengenai perancangan sistem informasi pada objek yang digunakan, perancangan arsitektur program yang akan dibuat, perancangan tampilan dan perancangan menu. Analisis dan perancangan adalah serangkaian kegiatan yang selalu beriringan dalam setiap pengembangan *software*, sebagai sebuah hubungan sebab dan akibat yang memunculkan sebuah siklus hidup sistem. Walaupun dalam kenyataannya pengembangan sistem yang sederhana, aktivitas ini tidak tampak.

##### **4.1.1 Tujuan Perancangan Sistem**

Tujuan dari perancangan ini merupakan suatu kegiatan pengembangan prosedur dan proses yang sedang berjalan untuk menghasilkan sesuatu yang baru atau memperbaharui yang ada untuk meningkatkan kinerja sistem itu sendiri sehingga dapat memenuhi hasil yang diinginkan. Rancangan aplikasi yang baru, akan diterapkan suatu kegiatan untuk menemukan dan mengembangkan metode, prosedur dan proses suatu data agar tujuan dari suatu organisasi dapat tercapai. Adapun tujuan dari tahap perancangan ini adalah untuk menghasilkan

perancangan aplikasi berupa pemodelan menggunakan pendekatan berorientasi objek sehingga dapat memperbaiki atau meningkatkan efisiensi kerja sistem yang sedang berjalan.

#### **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 master pada bengkel *2Q-Speed Concept*, data transaksi penjualan *spare part*, data transaksi perbaikan (*service*) hingga laporan laporan – laporan yang dibutuhkan oleh pemilik bengkel seperti laporan penjualan, laporan *service* kendaraan, dan laporan stok barang. 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**

Pada tahap ini, perancangan prosedur yang diusulkan dilakukan untuk memperbaiki atau meningkatkan efisiensi kerja. Tahap perancangan yang digambarkan merupakan tahap perancangan untuk membangun suatu sistem dan mengkonfigurasi komponen – komponen perangkat lunak dan perangkat keras sehingga menghasilkan sistem yang baik. Tahapan perancangan prosedur ini akan

dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan *UML*.

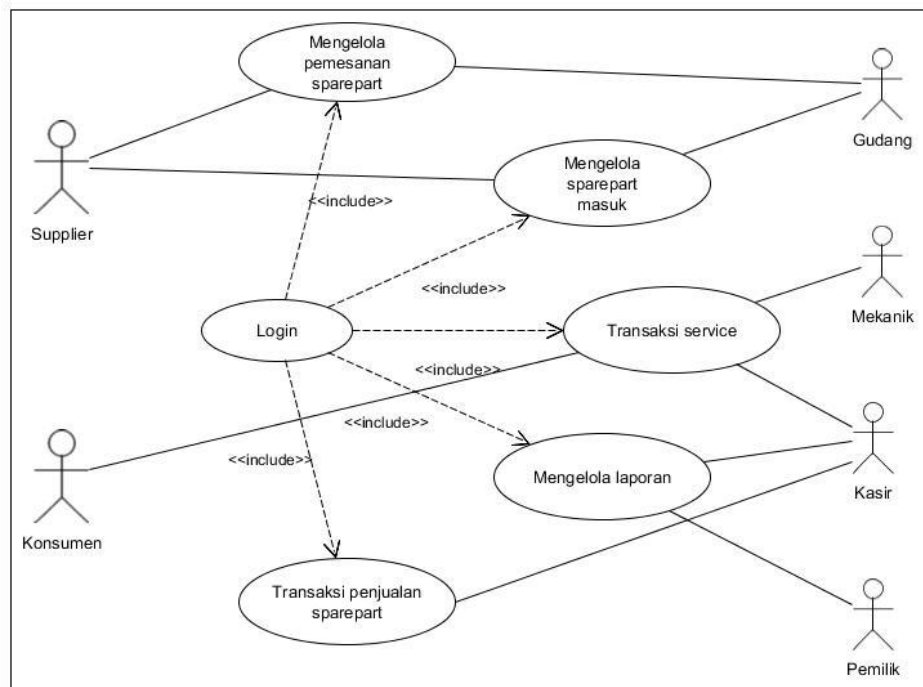
1. Proses pemesanan *spare part*
  - a. Bag. Gudang mengecek *spare part* yang stoknya sudah minim.
  - b. Bag. Gudang *login* ke sistem.
  - c. Bag. Gudang memilih *supplier* untuk menerima pesanan *spare part*.
  - d. Bag. Gudang mengisi data *spare part* yang habis atau stok minim dan menyertakan jumlahnya.
  - e. Sistem melakukan penyimpanan pemesanan kedalam *database*.
  - f. Bag. Gudang mencetak daftar pemesanan *spare part*.
  - g. Bag. Gudang menyerahkan daftar pemesanan kepada pemilik bengkel.
  - h. Pemilik bengkel menandatangani pemesanan *spare part*.
2. Proses *spare part* masuk gudang
  - a. Barang (*sparepart*) dikirim ke gudang.
  - b. Bag. Gudang mengecek kesesuaian jumlah barang dengan pesanan.
  - c. Bag. Gudang melakukan *login* ke sistem.
  - d. Bag. Gudang mencatat penerimaan *spare part*, jika jumlah yang dikirim dengan jumlah yang dipesan tidak sesuai bag. Gudang dapat mengubah jumlah *spare part* yang diterima.
  - e. Sistem menyimpan ke *database*.
3. Proses penjualan *spare part*
  - a. Konsumen meminta barang yang akan dibeli dengan jumlah tertentu.

- b. Kasir melakukan *login* ke dalam sistem.
  - c. Kasir menginputkan data – data barang yang dibeli konsumen beserta jumlah pembelian.
  - d. Sistem menampilkan total jumlah belanja konsumen.
  - e. Kasir menginputkan jumlah uang yang dibayarkan oleh konsumen.
  - f. Sistem menyimpan transaksi penjualan ke *database*.
  - g. Kasir mencetak nota penjualan untuk konsumen.
4. Proses *service* motor
- a. Konsumen membawa kendaraannya ke bengkel *2Q-Speed Concept*.
  - b. Kasir *login* ke sistem.
  - c. Kasir mencatat data – data fisik kendaraan beserta dengan keluhan yang akan diperbaiki.
  - d. Kasir menginput biaya *service* dan mengisi mekanik yang menangani.
  - e. Jika ada *spare part* yang harus ditambahkan maka kasir akan menambahkan dalam catatan *service* ini.
  - f. Sistem menampilkan jumlah total sekaligus menyimpan data kedalam *database*.
  - g. Kasir mencetak nota *service* motor.
5. Proses laporan stok *spare part*, transaksi *service*, penjualan dan pembelian *sparepart*.
- a. Pemilik melakukan *login* ke sistem.
  - b. Pemilik menginputkan periode laporan yang akan dicetak.

- c. Sistem menampilkan laporan stok *spare part*, transaksi *service*, penjualan dan pembelian *spare part* sesuai dengan periode tanggal yang diinputkan.

#### 4.1.3.1 Use Case Diagram yang Diusulkan

*Usecase diagram* yang diusulkan merupakan gambaran diagram yang menggambarkan semua kasus yang akan ditangani oleh perangkat lunak beserta aktor atau pelakunya. Berikut *use case diagram* yang diusulkan pada perancangan sistem informasi bengkel *2Q-Speed Concept* :



**Gambar 4.1 Use Case Diagram yang Diusulkan**

**a. Definisi Aktor dan Deskripsinya**

Berikut adalah definisi aktor dan deskripsi yang diusulkan:

**Tabel 4.1 Definisi Aktor dan Deskripsi Yang Diusulkan**

No	Aktor	Deskripsi
P-01	Konsumen	Pihak yang melakukan transaksi <i>service</i> dan pembelian barang
P-02	Supplier	Pihak yang menyediakan <i>spare part</i> yang dipesan
P-03	Mekanik	Pihak yang melakukan perbaikan motor
P-04	Gudang	Pihak yang mendapatkan akses untuk melakukan pemesanan <i>spare part</i> yang habis atau minim stok dan melakukan pencatatan barang masuk
P-05	Kasir	Pihak yang mendapatkan akses untuk melakukan penjualan <i>sparepart</i> , pencatatan pada saat konsumen melakukan <i>service</i> motor, menerima transaksi pembayaran dari konsumen dan membuat laporan per periode.
P-06	Pemilik	Pihak yang mendapatkan akses penuh untuk meninjau transaksi yang terjadi di bengkel

		melalui laporan yang sudah bersifat informasi dari bengkel.
--	--	---

**b. Definisi *Use Case* dan Deskripsinya**

Berikut adalah definisi *use case* dan deskripsi yang diusulkan:

**Tabel 4.2 Definisi *Use Case* Dan Deskripsi Yang Diusulkan**

No	<i>Use Case</i>	Deskripsi
UC-01	<i>Login</i>	Menentukan hak akses <i>user</i> pada saat masuk ke sistem.
UC-02	Mengelola pemesanan <i>spare part</i>	Merupakan proses mencatat pemesanan <i>spare part</i> yang habis atau stok minim. Untuk diberikan pada supplier.
UC-03	Mengelola <i>spare part</i> masuk	Merupakan proses mencatat <i>spare part</i> yang masuk ke gudang sesuai dengan catatan pemesanan yang sebelumnya dilakukan
UC-04	Transaksi penjualan <i>spare part</i>	Merupakan proses mencatat transaksi penjualan <i>spare part</i> setiap harinya.
UC-05	Transaksi <i>service</i>	Merupakan proses mencatat transaksi <i>service</i> yang terjadi dibengkel setiap

		harinya. Sekaligus untuk mencatat upah (biaya <i>service</i> ) yang dikerjakan oleh mekanik
UC-06	Mengelola laporan	Merupakan proses untuk melihat proses – proses transaksi yang terjadi di bengkel yang sudah berupa informasi penjualan pada bengkel.

#### 4.1.3.2 Skenario *Use Case*

Adapun skenario *use case* yang diusulkan sebagai langkah dari awal hingga proses *output* sebagai berikut:

1. Skenario *use case login* yang diusulkan

Berikut ini adalah skenario *use case login* yang diusulkan

**Tabel 4.3 Skenario *Use Case Login* yang Diusulkan**

<b>Identifikasi</b>	
<b>Judul</b>	Login
<b>No. Use case</b>	UC-01
<b>Deskripsi use case</b>	Menentukan hak akses user pada saat masuk ke sistem
<b>aktor</b>	Kasir, gudang dan pemilik
<b>Trigger</b>	Aktor menekan tombol Login
<b>Kondisi awal</b>	Aplikasi menampilkan sebuah form untuk login
<b>Skenario utama</b>	



Aksi aktor	Reaksi sistem
1. <i>User</i> mengisi <i>username</i> dan <i>password</i> pada <i>field</i> .	
2. <i>User</i> menekan tombol <i>login</i> .	
	3. Sistem memeriksa inputan berdasarkan <i>username</i> dan <i>password</i> pada <i>database</i> . Jika <i>username</i> atau <i>password</i> salah maka melakukan <i>login</i> lagi.
<b>Kondisi akhir skenario utama</b>	4. Sistem menampilkan halaman utama berdasarkan hak akses dari <i>user</i> .

2. Skenario *use case* mengelola pemesanan *spare part* yang diusulkan

Berikut ini adalah skenario *use case* pemesanan *spare part* yang diusulkan:

**Tabel 4.4 Skenario Use Case Mengelola Pemesanan Sparepart yang  
Diusulkan**

Identifikasi	
<b>Judul</b>	Mengelola pemesanan sparepart
<b>No. Use case</b>	UC-02
<b>Deskripsi use case</b>	Melakukan pemesanan barang habis atau stok minim
<b>Aktor</b>	Gudang
<b>Trigger</b>	Menekan tombol tambah / simpan / simpan pesanan

<b>Kondisi awal</b>	Aplikasi menampilkan form
<b>Skenario utama</b>	
<b>Aksi-aktor</b>	<b>Reaksi sistem</b>
1. Gudang melakukan pengecekan stok <i>spare part</i> sesuai stok minimal.	
2. Gudang mengklik menu pemesanan <i>spare part</i> .	
	3. Sistem menampilkan <i>form</i> pemesanan <i>spare part</i> .
4. Gudang memilih <i>spare part</i> yang akan dipesan dan mengisi jumlah barang yang akan dipesan.	
5. Gudang menekan tombol tambah.	
	6. Menampilkan data <i>spare part</i> yang akan dipesan.
7. Gudang menekan tombol simpan pesanan.	
<b>Kondisi akhir skenario utama</b>	8. Sistem menyimpan pesanan ke <i>database</i> dan menampilkan nota pemesanan yang harus ditandatangani oleh pemilik bengkel.

<b>Skenario Alternatif – 1</b>	
<b>Aksi aktor</b>	<b>Reaksi sistem</b>
1. Jika melakukan kesalahan dalam mencatat pesanan sebelum menekan tombol simpan pesanan. Gudang mengklik data yang tidak sesuai.	
	2. Sistem menampilkan data yang dipilih.
3. Klik tombol hapus untuk melakukan penambahan pesanan ulang.	
<b>Kondisi akhir skenario alternatif – 1</b>	4. Sistem menyimpan perubahan data pesanan dan menampilkan pada tabel.

3. Skenario *use case* mengelola *spare part* masuk yang diusulkan

**Tabel 4.5 Skenario *Use Case* Mengelola *Spare part* Masuk yang Diusulkan**

<b>Identifikasi</b>	
<b>Judul</b>	Mengelola Spare part Masuk
<b>No. Use case</b>	UC-03
<b>Deskripsi use case</b>	Mencatat barang masuk ke gudang dan menyesuaikan dengan pesanan yang telah dipesan sebelumnya.

<b>Aktor</b>	Gudang	
<b>Trigger</b>	Aktor menekan tombol Rubah/Reset/Simpan Barang Masuk	
<b>Kondisi awal</b>	Aplikasi menampilkan form	
<b>Skenario utama</b>		
<b>Aksi aktor</b>	<b>Reaksi sistem</b>	
1. Gudang mengecek <i>spare part</i> masuk		
2. Gudang menekan menu <i>spare part</i> masuk.		
	3. Sistem menampilkan <i>form spare part</i> masuk.	
4. Menekan tombol pilih dan memilih data pemesanan.		
	5. Menampilkan data pada form.	
6. Menekan tombol simpan barang masuk.		
<b>Kondisi akhir skenario utama</b>	7. Menyimpan data barang masuk ke database.	
<b>Skenario Alternatif – 1</b>		
<b>Aksi aktor</b>	<b>Reaksi sistem</b>	
1. Jika jumlah yang diterima		

lebih/kurang dari yang dipesan. Maka klik data barang yang lebih/kurang.	
	2. Menampilkan data yang dimaksud.
3. Mengisi kolom jumlah dengan jumlah barang yang diterima dan klik tombol simpan barang masuk.	
<b>Kondisi akhir skenario alternatif – 1</b>	4. Menyimpan data barang masuk ke <i>database</i> .

4. Skenario *use case* mengelola penjualan *spare part* yang diusulkan

Berikut ini adalah skenario *use case* penjualan *spare part* yang diusulkan:

**Tabel 4.6 Skenario Use Case Mengelola Penjualan Spare part yang Diusulkan**

<b>Identifikasi</b>	
<b>Judul</b>	Mengelola Penjualan Sparepart
<b>No. Use case</b>	UC-04
<b>Deskripsi use case</b>	Melakukan pencatatan transaksi penjualan yang ada dibengkel
<b>Aktor</b>	Kasir
<b>Trigger</b>	Aktor menekan tombol Tambah/Hapus/Cetak Nota Penjualan
<b>Kondisi awal</b>	Aplikasi menampilkan form

<b>Skenario utama</b>	
<b>Aksi aktor</b>	<b>Reaksi sistem</b>
1. Menekan tombol menu penjualan <i>spare part</i> .	
	2. Menampilkan form penjualan <i>spare part</i> .
3. Klik data barang yang akan dibeli konsumen dan input jumlah barang.	
	4. Menampilkan data yang dipilih ke tabel dan menampilkan jumlah belanja.
5. Mengisi jumlah uang yang diberikan konsumen lalu tekan <i>enter</i> .	
	6. Menampilkan jumlah kembalian dari pembelian tersebut.
7. Menekan tombol cetak nota penjualan untuk diberikan kepada konsumen.	
<b>Kondisi akhir skenario utama</b>	8. Menyimpan data penjualan dan menampilkan nota penjualan yang akan dicetak.

5. Skenario *use case* mengelola *service* motor yang diusulkan

Berikut ini adalah skenario *use case* mengelola *service* motor yang diusulkan:

**Tabel 4.7 Skenario Use Case Mengelola Service Motor yang Diusulkan**

<b>Identifikasi</b>	
<b>Judul</b>	Mengelola Service Motor
<b>No. Use case</b>	UC-05
<b>Deskripsi use case</b>	Melakukan pencatatan transaksi <i>service</i> motor
<b>Aktor</b>	Kasir
<b>Trigger</b>	Aktor menekan tombol Tambah/Hapus/Reset/Cetak Nota Service
<b>Kondisi awal</b>	Aplikasi menampilkan form
<b>Skenario utama</b>	
<b>Aksi-aktor</b>	<b>Reaksi sistem</b>
1. Mengklik menu <i>service</i> motor.	
	2. Menampilkan <i>form service</i> motor.
3. Mengisi data motor, keluhan, nama pemilik, biaya <i>service</i> dan mekanik yang mengerjakan <i>service</i> .	
	4. Menampilkan total biaya <i>service</i> yang harus dibayar.
5. Menginput jumlah uang yang	

diberikan konsumen lalu klik <i>enter</i> .	
	6. Menampilkan jumlah kembalian.
7. Menekan tombol cetak nota <i>service</i> .	
<b>Kondisi akhir skenario utama</b>	8. Menyimpan data pada <i>database</i> dan menampilkan nota yang akan dicetak.
<b>Skenario Alternatif – 1</b>	
<b>Aksi-aktor</b>	<b>Reaksi sistem</b>
1. Jika <i>service</i> memerlukan <i>spare part</i> tambahan klik data <i>spare part</i> yang ditambahkan saat <i>service</i> .	
	2. Menampilkan data <i>spare part</i> yang ditambahkan dan menampilkan jumlah yang harus dibayar dan yang telah ditambahkan dengan harga <i>spare part</i> yang ditambahkan.
3. Menginput jumlah uang yang diberikan oleh konsumen dan menekan tombol cetak nota <i>service</i> .	
<b>Kondisi akhir skenario alternative -1</b>	4. Menyimpan data transaksi <i>service</i>



	dan menampilkan nota yang akan dicetak.
--	---

6. Skenario *use case* mengelola laporan yang diusulkan

Berikut ini adalah skenario *use case* mengelola laporan yang diusulkan:

**Tabel 4.8 Skenario Use Case Mengelola Laporan yang Diusulkan**

Identifikasi	
<b>Judul</b>	Mengelola Laporan
<b>No. Use case</b>	UC-06
<b>Deskripsi use case</b>	Menampilkan pencatatan transaksi yang terjadi dibengkel
<b>Aktor</b>	Kasir dan pemilik
<b>Trigger</b>	Aktor menekan tombol Cetak Laporan
<b>Kondisi awal</b>	Aplikasi menampilkan form
Skenario utama	
Aksi aktor	Reaksi sistem
1. Aktor menekan menu bar laporan.	
	2. Menampilkan sub menu laporan.
3. Memilih laporan yang akan dicetak.	
	4. Menampilkan form laporan.
5. Mengisi periode laporan yang akan ditampilkan dan pilih cetak laporan.	

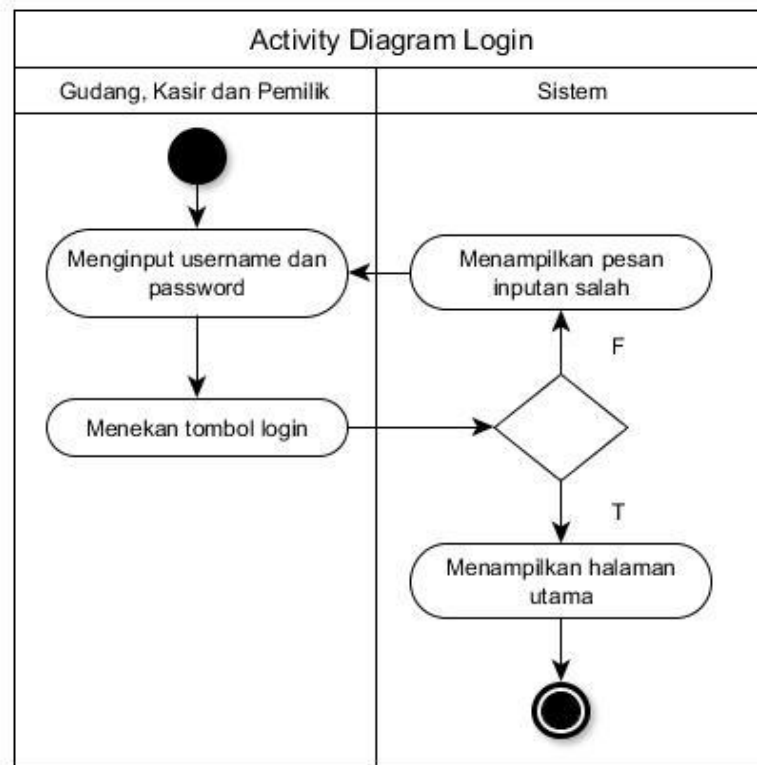
<b>Kondisi akhir skenario utama</b>	6. Menampilkan laporan sesuai dengan periode yang diminta.
-------------------------------------	--

#### 4.1.3.3 Activity Diagram

*Activity diagram* yang diusulkan ini menggambarkan ilustrasi aliran sebuah sistem yang akan berlangsung pada Bengkel 2Q-Speed Concept ini :

##### 1. Activity Diagram Login yang Diusulkan

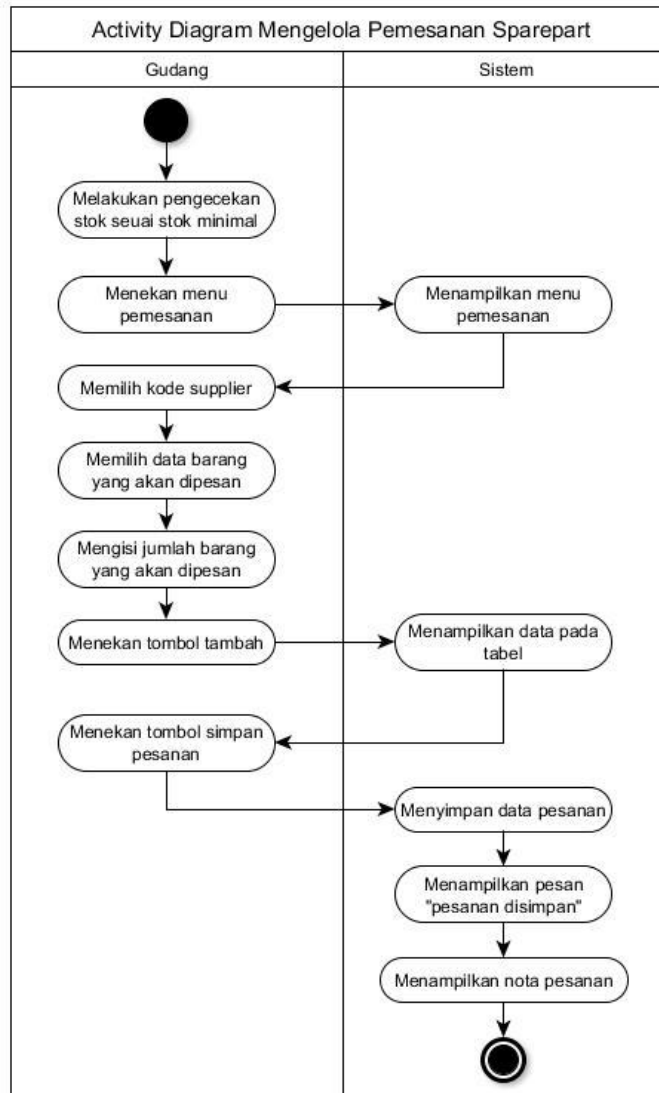
Berikut ini adalah *activity diagram login* yang diusulkan:



**Gambar 4.2 Activity Diagram Login yang Diusulkan**

## 2. Activity Diagram Mengelola Pemesanan Spare part yang Diusulkan

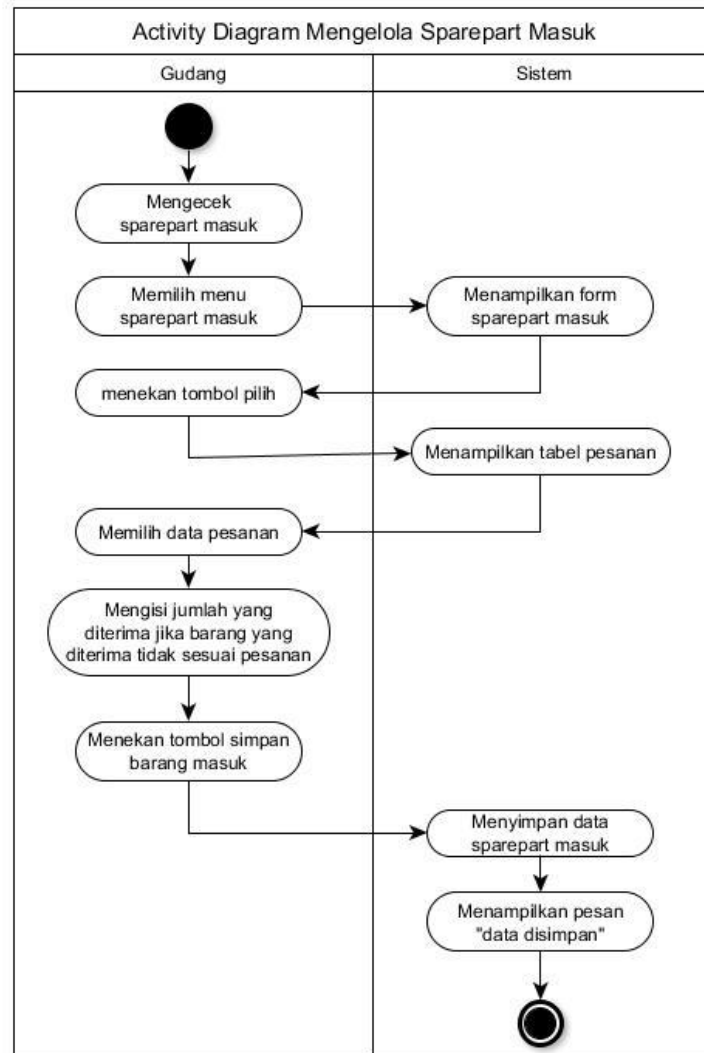
Berikut ini adalah *activity diagram* mengelola pemesanan *sparepart* yang diusulkan:



**Gambar 4.3 Activity Diagram Mengelola Pemesanan Spare part yang Diusulkan**

### 3. Activity Diagram Mengelola Spare part Masuk yang Diusulkan

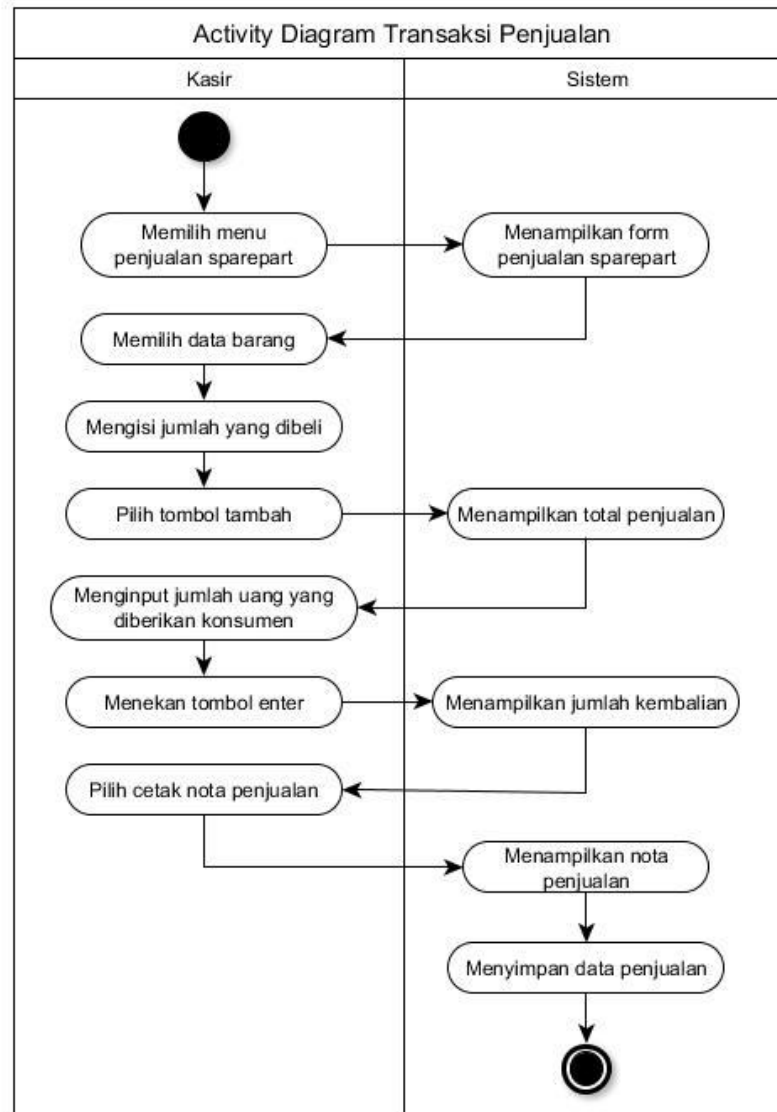
Berikut ini adalah *activity diagram* mengelola *sparepart* masuk yang diusulkan:



Gambar 4.4 Activity Diagram Mengelola Spare part Masuk yang Diusulkan

#### 4. Activity Diagram Transaksi Penjualan yang Diusulkan

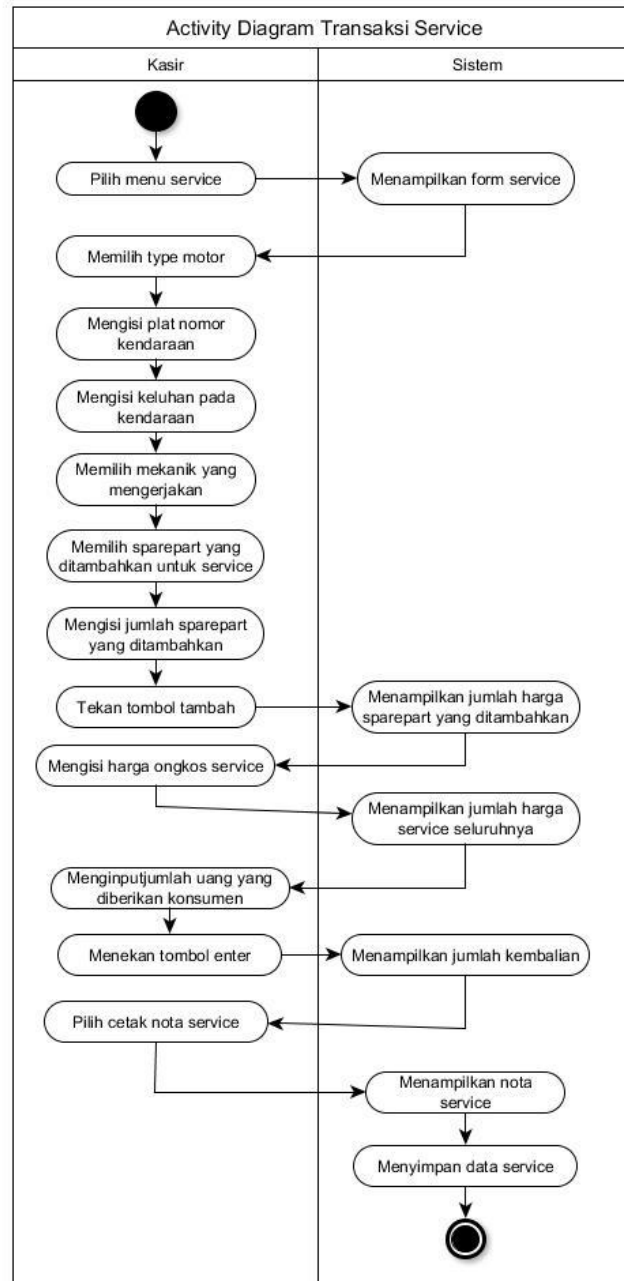
Berikut ini adalah *activity diagram* transaksi penjualan yang diusulkan:



**Gambar 4.5 Activity Diagram Transaksi Penjualan yang Diusulkan**

### 5. Activity Diagram Transaksi service

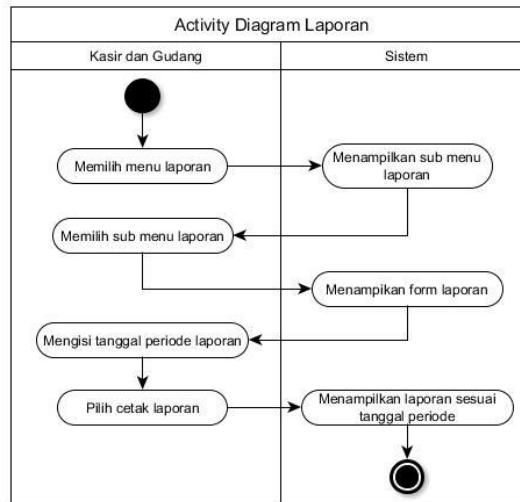
Berikut ini adalah *activity diagram* transaksi *service* yang diusulkan:



**Gambar 4.6 Activity Diagram Transaksi Service yang Diusulkan**

## 6. Activity Diagram Laporan yang Diusulkan

Berikut ini adalah *activity diagram* laporan yang diusulkan:



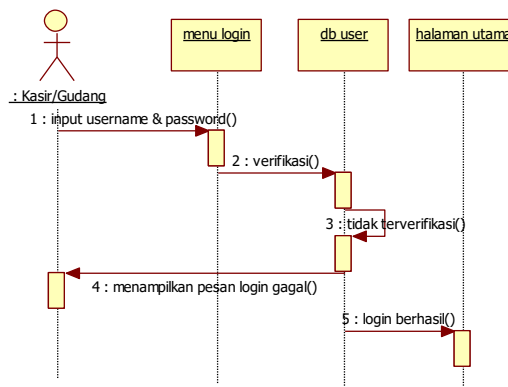
Gambar 4.7 Activity Diagram Laporan yang Diusulkan

### 4.1.3.4 Sequence diagram

Berikut ini merupakan *sequence diagram* yang diusulkan:

#### 1. Sequence Diagram Login yang Diusulkan

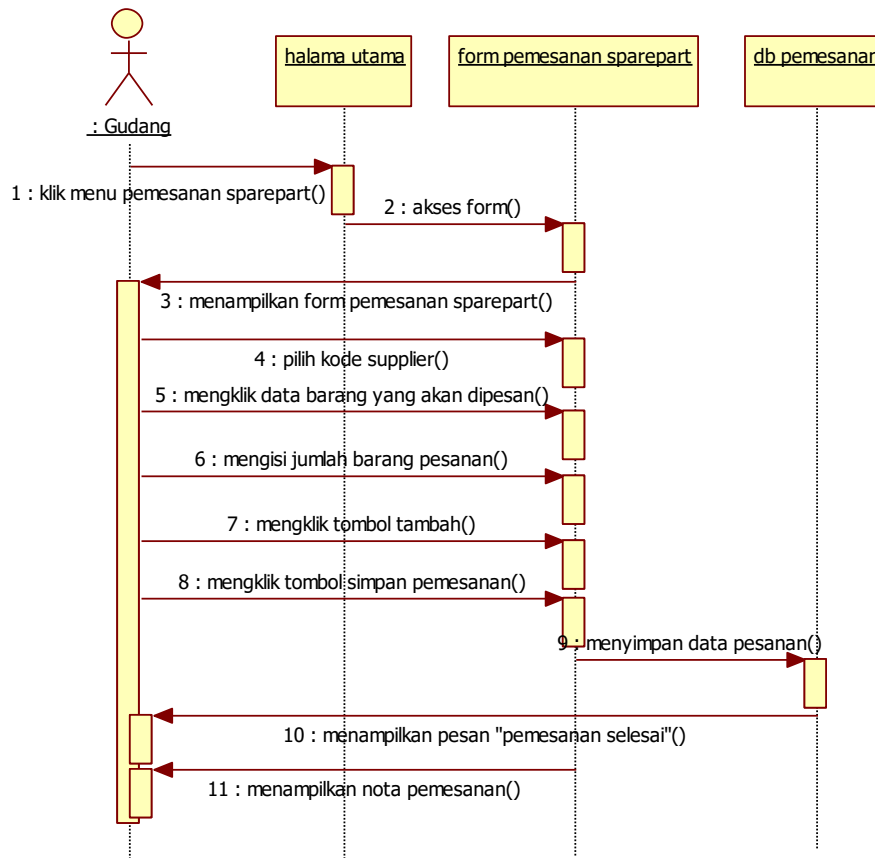
Berikut ini adalah *sequence diagram login* yang diusulkan:



Gambar 4.8 Sequence Diagram Login yang Diusulkan

## 2. Sequence Diagram Mengelola Pemesanan Spare part yang Diusulkan

Berikut ini adalah *sequence diagram* mengelola pemesanan *sparepart* yang diusulkan:

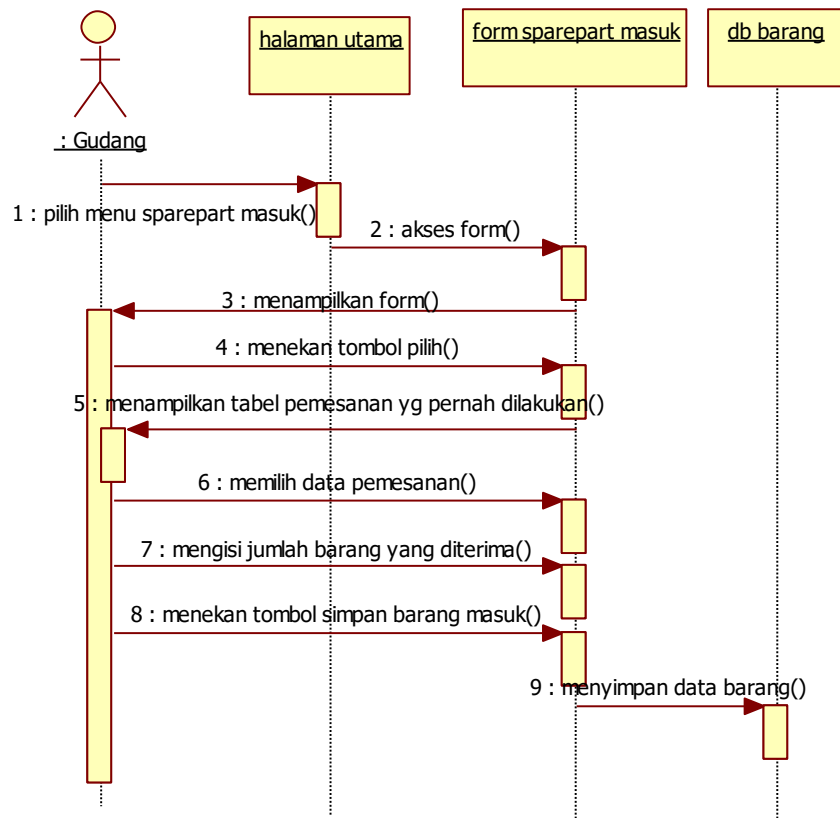


**Gambar 4.9** *Sequence Diagram* Mengelola Pemesanan *Spare part* yang Diusulkan



### 3. *Sequence Diagram* Mengelola *Spare part* Masuk yang Diusulkan

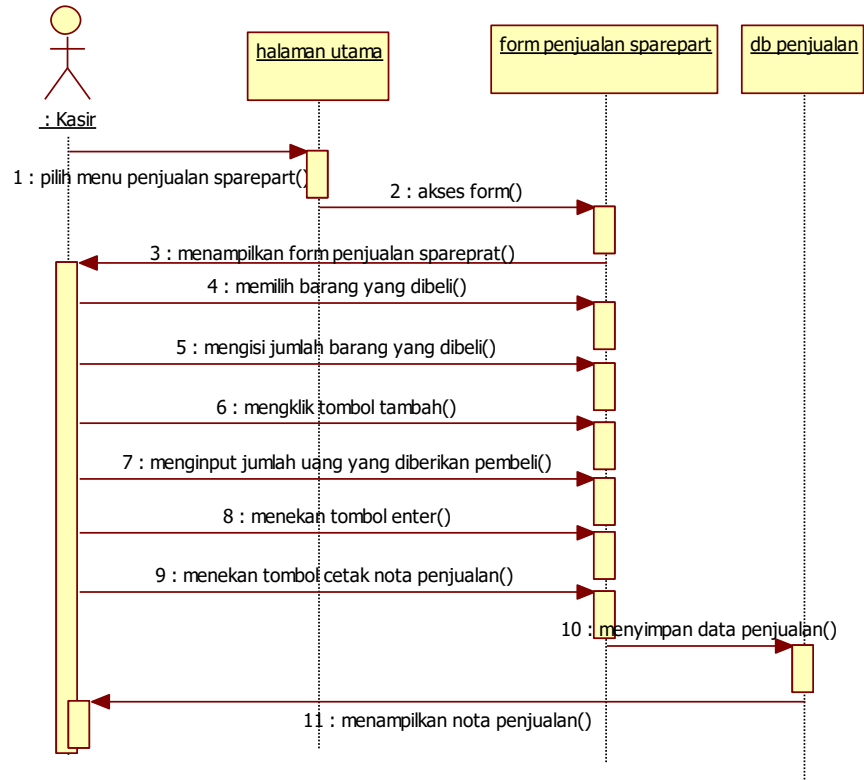
Berikut ini adalah *sequence diagram* mengelola *sparepart* masuk yang diusulkan:



**Gambar 4.10** *Sequence Diagram* Mengelola *Spare part* Masuk yang Diusulkan

#### 4. *Sequence Diagram* Transaksi Penjualan yang Diusulkan

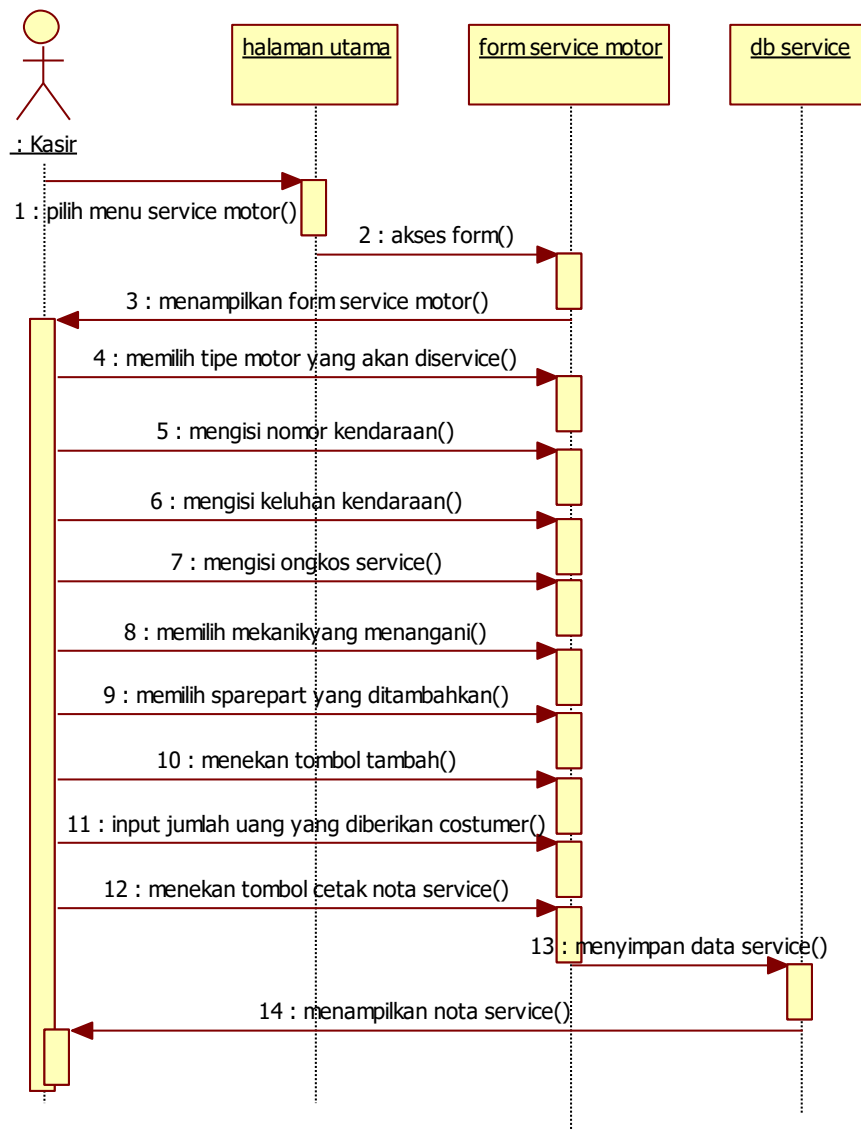
Berikut ini adalah *sequence diagram* transaksi penjualan yang diusulkan:



Gambar 4.11 *Sequence Diagram* Transaksi Penjualan yang Diusulkan

## 5. *Sequence Diagram* Transaksi *Service* yang Diusulkan

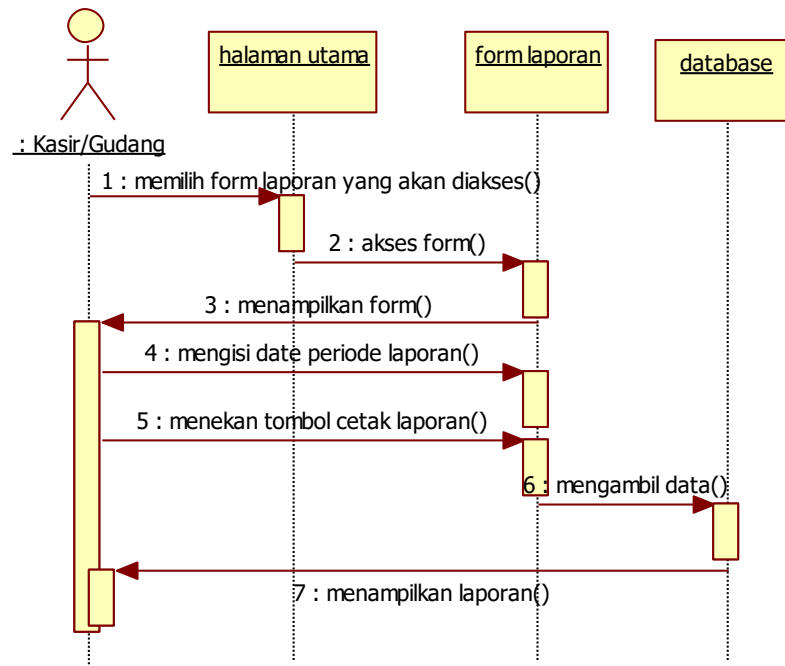
Berikut ini adalah *sequence diagram* transaksi *service* yang diusulkan:



**Gambar 4.12** *Sequence Diagram* Transaksi *Service* yang Diusulkan

## 6. Sequence Diagram Laporan yang Diusulkan

Berikut ini adalah *sequence diagram* laporan yang diusulkan:

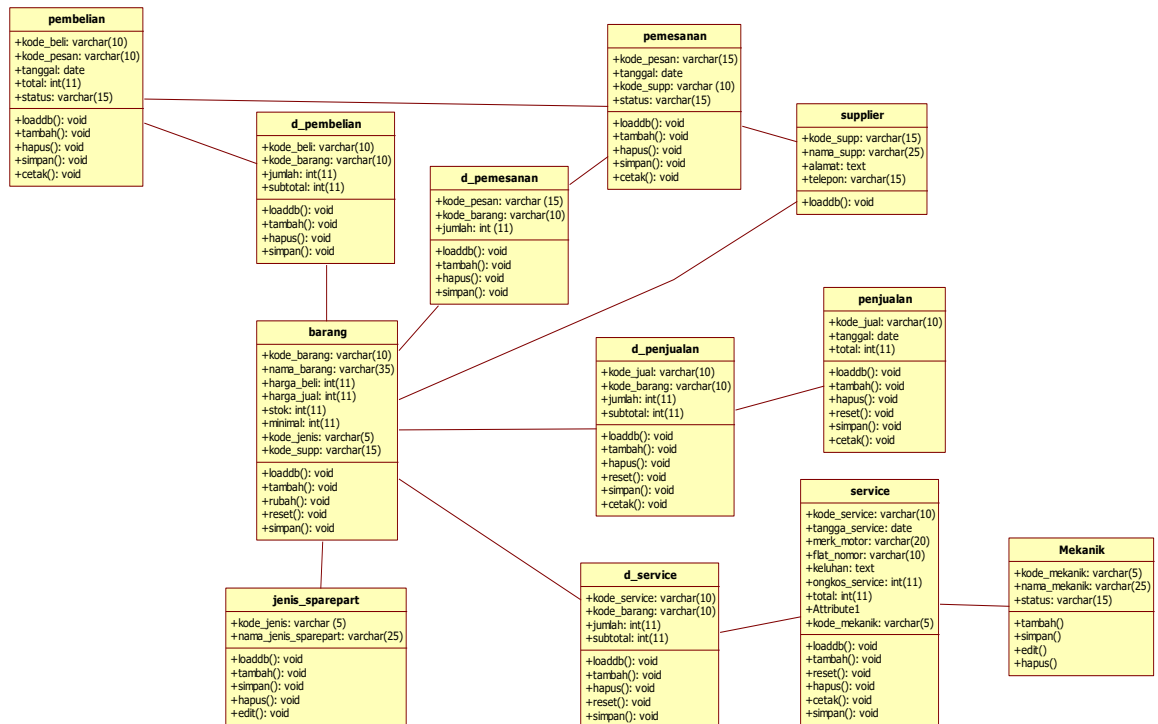


Gambar 4.13 *Sequence Diagram* Laporan yang Diusulkan

## 4.1.4 Perancangan Data

### 4.1.4.1 Class Diagram

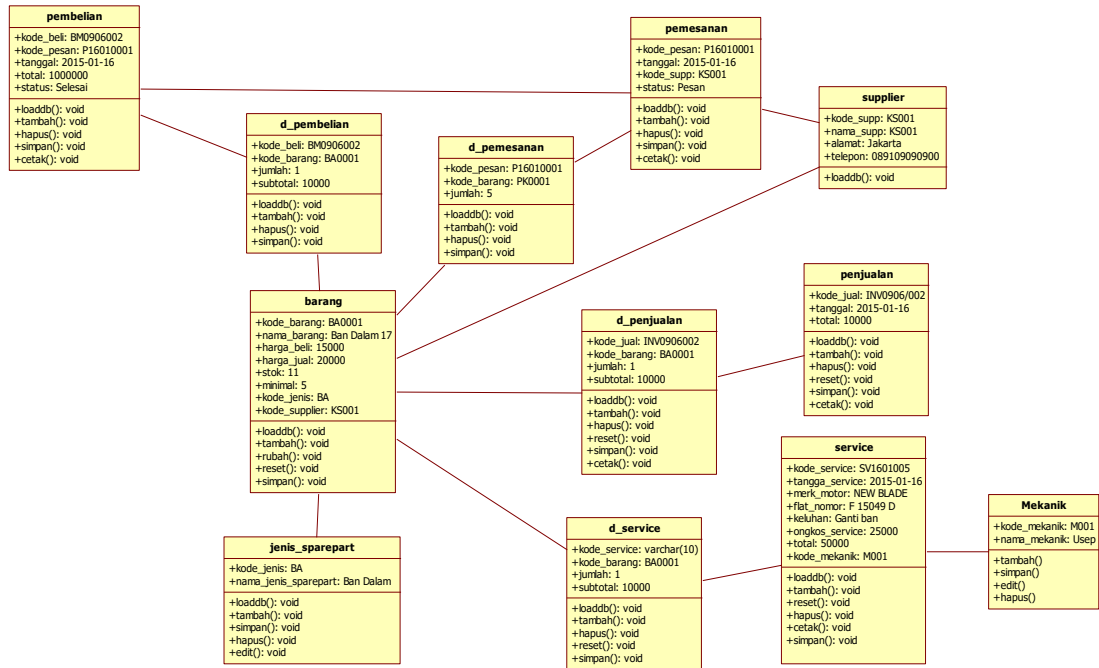
Berikut ini merupakan gambar dari *class diagram* yang diusulkan :



Gambar 4.14 Class Diagram yang Diusulkan

### 4.1.4.2 Object Diagram

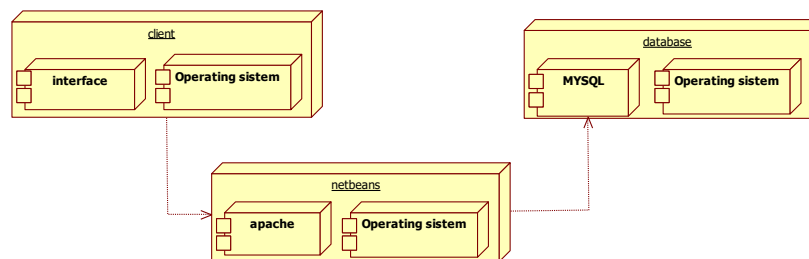
Berikut ini merupakan gambar dari *object diagram* yang diusulkan :



Gambar 4.15 Object Diagram yang Diusulkan

### 4.1.4.3 Deployment Diagram

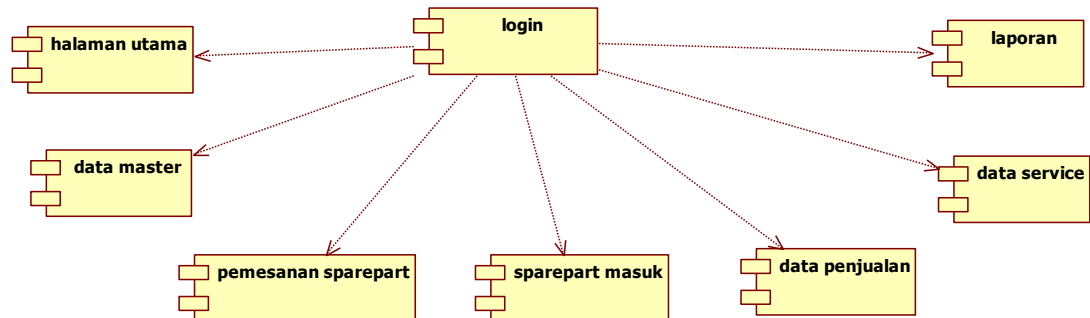
Berikut adalah gambar dari *deployment diagram* yang diusulkan sebagai berikut:



Gambar 4.16 Deployment Diagram yang Diusulkan

#### 4.1.4.4 Component Diagram

Berikut adalah gambar *component diagram* yang diusulkan adalah sebagai berikut:



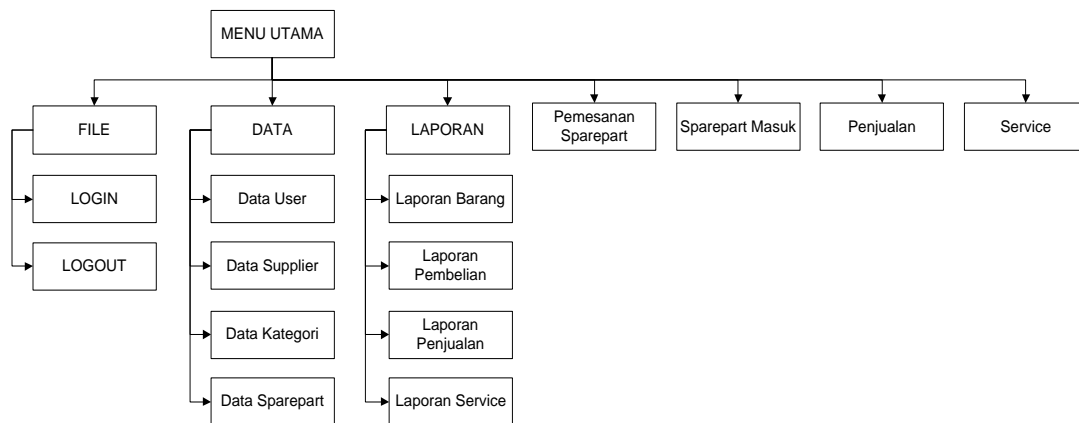
**Gambar 4.17 Component Diagram yang Diusulkan**

## 4.2 Perancangan Antar Muka

Perancangan antarmuka atau *interface* ini akan menggambarkan letak-letak tampilan yang dirancang pada sistem yang diusulkan. Adapun yang akan dibahas adalah struktur menu, perancangan *input* dan *output*.

### 4.2.1 Struktur Menu

Struktur menu di bawah ini merupakan bagian dari alur aplikasi bengkel *2Q-Speed Concept* yang mempermudah *user* menelusur komponen – komponen sistem yang ada di dalamnya.



**Gambar 4.18 Struktur Menu**

#### 4.2.2 Perancangan *Input*

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

##### 1. Rancangan Menu *Login*

Tampilan yang dirancang berfungsi untuk dapat *login* ke sistem.

FORM LOGIN

LOGO

Username:

Password:

**Gambar 4.19 Rancangan Menu *Login***



## 2. Rancangan Data Master

Tampilan ini berfungsi untuk me-*manage* data master seperti data *user*, data *supplier*, data *sparepart* dan data lainnya.

### a. *Manage Data User*

*Manage data user* berfungsi untuk melakukan siapa saja yang dapat mengakses sistem ini.

DATA USER

Username

Password

Nama User

Hak Akses

No	Username	Password	Nama Pengguna	Hak Akses

**Gambar 4.20 Rancangan *Manage Data Master User***

### b. *Manage Data Supplier Sparepart*

*Manage data supplier sparepart* ini berfungsi untuk mencatat *supplier – supplier* yang menjadi pemasok *sparepart* tetap pada bengkel ini.

**DATA SUPPLIER SPAREPART**

Kode Supplier

Nama Supplier

Alamat

No Telepon

No	Kode Supplier	Nama Supplier	Alamat Supplier	No Telepon

**Gambar 4.21 Rancangan Mengelola Data Master Supplier**

**c. Manage Data Jenis Sparepart**

*Manage data jenis sparepart* ini berfungsi untuk mengkategorikan *sparepart* yang masuk ke bengkel ini. Untuk memudahkan gudang saat melakukan cek barang dan cek barang dilakukan berdasarkan jenisnya.

**DATA JENIS SPAREPART**

Kode Jenis

Jenis Sparepart

No	Kode Jenis Sparepart	Nama Jenis Sparepart

**Gambar 4.22 Rancangan Mengelola Data Master Jenis Sparepart**

**d. Manage Data Sparepart**

*Manage data sparepart* ini berfungsi untuk mencatat *sparepart* yang sudah dan harus selalu ada di bengkel.

DATA SPAREPART

Kode Jenis

Jenis Sparepart  Harga Jual

Kode Barang  Harga Beli

Nama Barang  Minimal  Stok

No	Kode Barang	Nama Barang	Harga Jual	Harga Beli	Minimal	Stok

**Gambar 4.23 Rancangan Mengelola Data Master *Sparepart***

### 3. Rancangan Pemesanan Sparepart

Tampilan ini berfungsi untuk gudang melakukan pemesanan *sparepart* yang sudah habis atau stok yang sudah menipis.

Kode Supplier	<input type="text"/>	Data Sparepart	<input type="checkbox"/>	Barang Minim			
Nama Supplier	<input type="text"/>	<input type="text"/>					
Alamat	<input type="text"/>	No	Kode	Nama	Harga Jual	Harga Beli	Minimal
Telepon	<input type="text"/>	<input type="text"/>					
Kode Pemesanan	<input type="text"/>	<input type="text"/>					
Tanggal	<input type="text"/>	<input type="text"/>					
Data Sparepart yang Dipesan :							
Kode Barang	<input type="text"/>	Kode	Nama	Jumlah			
Nama Barang	<input type="text"/>	<input type="text"/>					
Jumlah	<input type="text"/>	<input type="text"/>					
	<input type="button" value="Tambah"/>	<input type="button" value="Hapus"/>					
<input type="button" value="SIMPAN PEMESANAN"/>							

**Gambar 4.24 Rancangan Pemesanan Sparepart**

### 4. Rancangan Sparepart Masuk

Tampilan ini berfungsi untuk mencatat *sparepart* yang masuk ke gudang. Untuk menyelaraskan apakah barang sesuai dengan pesanan.

<b>DATA PEMESANAN</b>		Kode Pembelian	<input type="text"/>				
Kode Pemesanan	<input type="text"/>	Tanggal Pembelian	<input type="text"/>				
	<input type="button" value="Pilih"/>						
Tanggal	<input type="text"/>						
Supplier	<input type="text"/>						
Kode Barang	<input type="text"/>	Kode	Nama	Jumlah Pesan	Harga	Jumlah Terima	Sub Total
Jumlah terima	<input type="text"/>	<input type="text"/>					
	<input type="button" value="Rubah"/>						
<input type="button" value="Simpan penerimaan sparepart"/>		<input type="button" value="Reset"/>					

**Gambar 4.25 Rancangan Sparepart Masuk**

## 5. Rancangan Penjualan

Tampilan ini berfungsi untuk melakukan pencatatan penjualan *sparepart* yang terjadi setiap harinya. Pencatatan penjualan ini wajib dilakukan untuk menelaraskan barang dan penjualan dan untuk menghasilkan laporan.

**Form Pilih Barang :**

Kode Barang       Harga

Nama Barang       Jumlah

Kode Barang	Nama Barang	Jumlah	Harga	Sub Total

HAPUS

BAYAR

Reset

**Gambar 4.26 Rancangan Penjualan *Sparepart***

## 6. Rancangan *Service Motor*

Tampilan ini berfungsi untuk mencatat transaksi *service* dan jika ada penambahan *sparepart* yang digunakan saat melakukan *service*.

<b>Data Service Motor</b> Kode Service <input type="text"/> Tanggal Service <input type="text"/> Type Motor <input type="text"/> <input type="button" value="v"/> Plat Motor <input type="text"/> Keluhan Motor <input type="text"/>		<table border="1"> <thead> <tr> <th>Kode Barang</th> <th>Nama Barang</th> <th>Jumlah</th> <th>Harga</th> <th>Sub Total</th> </tr> </thead> <tbody> <tr> <td colspan="5" style="height: 100px;"> </td> </tr> </tbody> </table>	Kode Barang	Nama Barang	Jumlah	Harga	Sub Total					
Kode Barang	Nama Barang	Jumlah	Harga	Sub Total								
<b>DATA SPAREPART YANG DITAMBAHKAN :</b> Kode Barang <input type="text"/> Nama Barang <input type="text"/> Jumlah <input type="text"/> <input type="button" value="Tambah"/> <input type="button" value="Hapus"/>	<b>DATA BARANG :</b> <table border="1"> <thead> <tr> <th>No</th> <th>Kode barang</th> <th>Nama Barang</th> <th>Stok</th> </tr> </thead> <tbody> <tr> <td colspan="4" style="height: 100px;"> </td> </tr> </tbody> </table>	No	Kode barang	Nama Barang	Stok					Ongkos Service <input type="text"/> Mekanik <input type="text"/> <input type="button" value="v"/> Total <input type="text"/> Bayar <input type="text"/> Kembalian <input type="text"/> <input type="button" value="Cetak Nota"/> <input type="button" value="Reset"/>		
No	Kode barang	Nama Barang	Stok									

**Gambar 4.27 Rancangan *Service Motor***

## 7. Rancangan Laporan

Tampilan ini berfungsi untuk mencetak laporan sesuai periode yang diinginkan.

<b>PERIODE LAPORAN</b>  <input type="text"/> <input type="text"/>  <input type="button" value="Cetak Laporan"/>
---

**Gambar 4.28 Rancangan Laporan**

### 4.2.3 Perancangan *Output*

Rancangan *output* adalah hasil dari proses yang dilakukan oleh sistem, hasil dari inputan yang nantinya akan di tampilkan pada keluaran dalam bentuk informasi. *Output* yang penulis maksud adalah laporan dari semua data yang telah diinputkan dan disimpan kedalam *database* lalu dipanggil kembali dalam bentuk laporan.

#### 1. Laporan Data Barang

Tampilan laporan data barang merupakan *output* yang dihasilkan dari *input* barang yang masuk, laporan ini berupa informasi seluruh data barang yang masih tersedia yang ada dalam *database*.

2Q-Speed Concept			
Jln. Ciguruwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung			
LAPORAN STOK SPAREPART			
No	Kode Barang	Nama Sparepart	Jumlah Stok
\$V	\$(kode_barang)	\$(nama_barang)	\$(stok)
Mengetahui Pemilik,		Summary	Mengetahui Gudang,
<input type="text"/>			<input type="text"/>

**Gambar 4.29 Rancangan Laporan Barang**

#### 2. Laporan Pembelian *Sparepart*

Tampilan laporan pembelian *sparepart* merupakan *output* yang dihasilkan dari *input* pemesanan yang dilakukan, laporan ini berupa informasi atau pencatatan tentang seluruh data pemesanan yang pernah dilakukan yang ada didalam *database*.

2Q-Speed Concept		
Jln. Ciguruwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung		
Laporan Pembelian		
Periode :	\$P{a}	\$P{b}
Kode Pembelian : \$F{pembelian_kode_beli}		
Nama Sparepart	Jumlah	Sub Total
\$F{barang_nama_barang}	\$F	\$F
Total Pembelian :		\$F
Total Pembelian Periode ini :		\$V{sum}
Mengetahui Pemilik,	Summary	Mengetahui Gudang,
<input type="text"/>		<input type="text"/>

**Gambar 4.30 Rancangan Laporan Pembelian**

### 3. Laporan Penjualan *Sparepart*

Tampilan laporan penjualan *sparepart* merupakan *output* yang dihasilkan dari *input* setiap penjualan yang terjadi dibengkel, laporan ini berupa informasi tentang seluruh data penjualan yang ada didalam *database*.

2Q-Speed Concept		
Jln. Ciguruwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung		
Laporan Penjualan		
Periode :	\$P{a}	\$P{b}
Kode Penjualan : \$F{penjualan_kode_jual}		
Nama Sparepart	Jumlah	Sub Total
\$F{barang_nama_barang}	\$F	\$F
Total Penjualan :		\$F
Total Penjualan Periode ini :		\$V{sum}
Mengetahui Pemilik,	Summary	Mengetahui Gudang,
<input type="text"/>		<input type="text"/>

**Gambar 4.31 Rancangan Laporan Penjualan**



#### 4. Laporan Service Motor

Tampilan laporan *service* motor merupakan *output* yang dihasilkan dari *input* data *service* motor yang terjadi di bengkel, laporan ini berupa informasi tentang seluruh data *service* motor yang ada didalam *database*.

2Q-Speed Concept				
Jln. Ciguruwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung				
Laporan Service				
Periode : \$P(a) - \$P(b)				
No service	Merk Motor	Plat Motor	Mekanik	Total Service
\$F	\$F	\$F	\$F	\$F(service_total)
Total Pendapatan :				\$V(sum)
Mengetahui Pemilik,		Summary	Mengetahui Gudang,	
<input type="text"/>			<input type="text"/>	

Gambar 4.32 Rancangan Laporan *Service* Motor

#### 5. Nota Pemesanan Sparepart

Berikut ini adalah tampilan nota pemesanan *sparepart*.

2Q-Speed Concept		
Jln. Ciguruwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung		
Nota Pemesanan Sparepart :		
Tanggal	: \$F(pemesanan_tanggal)	
Nama Supplier	: \$F(supplier_nama_supp)	
Alamat	: \$F(supplier_alamat)	
Telepon	: \$F(supplier_telepon)	
No	Nama Barang	Jumlah
\$V	\$F(barang_nama_barang)	\$F
Mengetahui Pemilik,		Mengetahui Gudang,
<input type="text"/>		<input type="text"/>

Gambar 4.33 Nota Pemesanan *Spare part*

## 6. Nota Penjualan Spare part

Berikut ini adalah tampilan nota penjualan *sparepart*.

2Q-Speed Concept			
Jln. Ciguruwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung			
Nota Penjualan :			Tanggal Transaksi : \$F
No	Barang	Qty	Subtotal
\$V	\$F{barang_nama_barang}	\$F	\$F
<b>PERHATIAN !!!</b> barang yang sudah dibeli tidak dapat dikembalikan aau ditukar		<b>TOTAL</b>	\$F{penjualan_total}
		<b>BAYAR</b>	\$F
		<b>KEMBALI</b>	\$F
Tanda terima,		kasir	

Gambar 4.34 Nota Penjualan *Spare part*

## 7. Nota Service

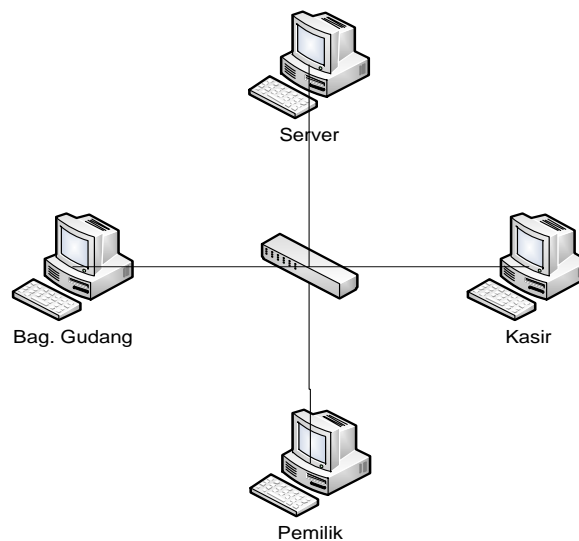
Berikut adalah tampilan nota *service*.

2Q-Speed Concept			
Jln. Ciguruwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung			
Tanggal Transaksi : \$F		Title	
Tanggal Ambil : \$P{tg}			
Konsumen : \$F{service_pemilik}			
TYPE : \$F{service_merk_motor}		MEKANIK : \$F	
1	\$F{service_keluhan}		\$F
		<b>JUMLAH</b>	\$F{service_total}
Tanda terima,		kasir	

Gambar 4.35 Nota *Service*

### 4.3 Perancangan Arsitektur Jaringan

Perancangan arsitektur jaringan yang digunakan pada bengkel ini adalah topologi *star*, karena dengan menggunakan topologi *star* masing – masing *work station* akan dihubungkan secara langsung ke *server*. Keunggulan dari topologi *star* adalah bahwa dengan adanya kabel tersendiri untuk setiap *work station* ke *server*. Dan juga bila terdapat gangguan di suatu jalur kabel maka gangguan hanya akan terjadi dalam komunikasi antara *work station* yang bersangkutan dengan *server* itu saja dan jaringan secara keseluruhan tidak mengalami gangguan. Berikut ini adalah gambar jaringan topologi star:



**Gambar 4.36 Perancangan Arsitektur Jaringan**

### 4.4 Pengujian

Proses pengujian berfokus pada logika internal *software*, memastikan bahwa semua pernyataan sudah diuji dan pada eksternal fungsional mengarahkan

pengujian untuk menemukan kesalahan – kesalahan dan memastikan bahwa masukan yang dibatasi akan memberikan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

#### 4.4.1 Rencana Pengujian

Rencana pengujian yang akan penulis uraikan dilakukan dengan pengujian sistem. Pengujian sistem informasi bengkel *2Q-Speed Concept* berikut menggunakan data uji berupa masukan dari pengguna di bengkel *2Q-Speed Concept*. Rencana pengujian selengkapnya terlihat pada tabel dibawah ini:

**Tabel 4.9 Rencana Pengujian**

<b>Kelas Uji</b>	<b>Tingkat Uji</b>	<b>Butir Uji</b>	<b>Jenis Pengujian</b>
<i>Login</i>	<i>Username</i>	Modul	<i>BlackBox</i>
	<i>Password</i>	Modul	<i>BlackBox</i>
Data Master	<i>Data User</i>	Modul	<i>BlackBox</i>
	Data Mekanik	Modul	<i>BlackBox</i>
	<i>Data Jenis Sparepart</i>	Modul	<i>BlackBox</i>
	<i>Data Sparepart</i>	Modul	<i>BlackBox</i>
Transaksi	<i>Penjualan Sparepart</i>	Modul	<i>BlackBox</i>
	<i>Service Motor</i>	Modul	<i>BlackBox</i>
Laporan	Laporan Barang	Modul	<i>BlackBox</i>

	Laporan Pembelian	Modul	<i>Blackbox</i>
	Laporan Penjualan	Modul	<i>BlackBox</i>
	Laporan <i>Service Motor</i>	Modul	<i>BlackBox</i>

#### 4.4.2 Kasus dan Hasil Pengujian

Kasus dan hasil pengujian dibuat untuk mengetahui apakah Sistem Informasi pada bengkel *2Q-Speed Concept* ini dapat berjalan dengan baik atau tidak.

##### 1. Pengujian Data *Login*

Berikut ini adalah tabel pengujian data *login*:

**Tabel 4.10 Pengujian Data *Login***

<b>Kasus dan Hasil Uji (Data Normal)</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<i>Username : admin</i>	Mendapatkan all akses kedalam sistem	Mendapatkan all akses kedalam sistem	[✓]Diterima
<i>Password : admin</i>			[ ]Ditolak
<i>Username : pm</i>	<i>Login sukses dengan hak ases mengelola data user dan laporan</i>	<i>Login sukses dengan hak ases mengelola data user dan laporan</i>	[✓]Diterima
<i>Password : pm</i>			[ ]Ditolak

<i>Username :</i> <i>kasir</i>	<i>Login sukses</i> dengan hak akses	<i>Login sukses</i> dengan hak akses	<input checked="" type="checkbox"/> Diterima
<i>Password :</i> <i>kasir</i>	<i>form penjualan dan</i> <i>form service</i>	<i>form penjualan dan</i> <i>form service</i>	<input type="checkbox"/> Ditolak
<i>Username : gd</i>	<i>Login sukses</i> dengan hak akses	<i>Login sukses</i> dengan hak akses	<input checked="" type="checkbox"/> Diterima
<i>Password: gd</i>	<i>form pemesanan</i> <i>spare part dan form</i> <i>spare part masuk</i>	<i>form pemesanan</i> <i>spare part dan form</i> <i>spare part masuk</i>	<input type="checkbox"/> Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
<i>Username: gd</i>	<i>Login gagal dan</i>	<i>Login gagal dan</i>	<input checked="" type="checkbox"/> Diterima
<i>Password :</i> <i>admin</i>	<i>muncul pesan error</i>	<i>muncul pesan error</i>	<input type="checkbox"/> Ditolak

## 2. Pengujian Data Master

Berikut ini adalah tabel pengujian data master:

**Tabel 4.11 Pengujian Data Master**

<b>Kasus dan Hasil Uji (Data Normal</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mengisi semua <i>field</i> untuk menambah data <i>sparepart</i>	Muncul pesan data tersimpan dan data <i>sparepart</i> bertambah di <i>database</i>	Muncul pesan data tersimpan dan data <i>sparepart</i> bertambah di <i>database</i>	[✓]Diterima  [ ]Ditolak
<i>Input data supplier</i> baru	Muncul pesan data berhasil disimpan	Muncul pesan data berhasil disimpan	[✓]Diterima [ ]Ditolak
Mengisi semua <i>field</i> untuk menambah jenis <i>sparepart</i>	Muncul pesan data tersimpan dan data jenis <i>sparepart</i> bertambah <i>didatabase</i>	Muncul pesan data tersimpan dan data jenis <i>sparepart</i> bertambah <i>didatabase</i>	[✓]Diterima  [ ]Ditolak
Mengisi semua <i>field data user</i>	Muncul pesan data tersimpan dan data <i>user</i> bertambah di <i>database</i>	Muncul pesan data tersimpan dan data <i>user</i> bertambah di <i>database</i>	[✓]Diterima  [ ]Ditolak

<b>Kasus dan Hasil Uji (Data Salah)</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mengosongkan salah satu <i>field</i> pada proses tambah data	Muncul pesan <i>error</i> “isi semua kolom yang ada”	Muncul pesan <i>error</i> “isi semua kolom yang ada”	[✓]Diterima  [ ]Ditolak
Mengisi no telepon pada data <i>user</i> dengan huruf	Muncul pesan <i>error</i> “tolong <i>input</i> -kan angka”	Muncul pesan <i>error</i> “tolong <i>input</i> -kan angka”	[✓]Diterima  [ ]Ditolak

### 3. Pengujian Data Transaksi

Berikut ini adalah tabel pengujian data transaksi:

**Tabel 4.12 Pengujian Data Transaksi**

<b>Kasus dan Hasil Uji (Data Normal)</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mengisi seluruh <i>field</i> menambah pemesanan <i>sparepart</i>	Muncul pesan pesanan tersimpan dan menampilkan nota pemesanan untuk dicetak	Muncul pesan pesanan tersimpan dan menampilkan nota pemesanan untuk dicetak	[✓]Diterima  [ ]Ditolak
Mengisi	Muncul pesan data	Muncul pesan data	[✓]Diterima



seluruh <i>field</i> menambah data <i>sparepart</i> masuk	barang masuk tersimpan	barang masuk tersimpan	<input type="checkbox"/> Ditolak
Mengisi semua <i>field</i> penjualan <i>sparepart</i>	<i>Database</i> penjualan bertambah dan menampilkan nota penjualan untuk dicetak	<i>Database</i> penjualan bertambah dan menampilkan nota penjualan untuk dicetak	<input checked="" type="checkbox"/> Diterima  <input type="checkbox"/> Ditolak
Mengisi semua <i>filed service</i> motor	<i>Database service</i> motor bertambah dan menampilkan nota <i>service</i> motor untuk dicetak	<i>Database service</i> motor bertambah dan menampilkan nota <i>service</i> motor untuk dicetak	<input checked="" type="checkbox"/> Diterima  <input type="checkbox"/> Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mengosongkan salah satu <i>filed</i> pada <i>form</i> penjualan	Muncul pesan <i>error</i> “jumlah belum diisi”	Muncul pesan <i>error</i> “jumlah belum diisi”	<input checked="" type="checkbox"/> Diterima  <input type="checkbox"/> Ditolak

#### 4. Pengujian Data Laporan

Berikut ini adalah tabel pengujian data laporan:

**Tabel 4.13 Pengujian Data Laporan**

<b>Kasus dan Hasil Uji (Data Normal)</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mengisi periode laporan selama 1 bulan pada laporan penjualan	menampilkan laporan penjualan sesuai dengan periode.	menampilkan laporan penjualan sesuai dengan periode.	[ <input checked="" type="checkbox"/> ]Diterima  [ <input type="checkbox"/> ]Ditolak
Mengisi periode laporan selama 1 bulan pada laporan service	menampilkan laporan <i>service</i> sesuai dengan periode.	menampilkan laporan <i>service</i> sesuai dengan periode.	[ <input checked="" type="checkbox"/> ]Diterima  [ <input type="checkbox"/> ]Ditolak
<b>Kasus dan Hasil Uji (Data Salah)</b>			
<b>Data Masukan</b>	<b>Yang Diharapkan</b>	<b>Pengamatan</b>	<b>Kesimpulan</b>
Mengisi tanggal periode dengan tahun yang akan datang	Muncul pesan " <i>the document has no pages</i> "	Muncul pesan " <i>the document has no pages</i> "	[ <input checked="" type="checkbox"/> ]Diterima  [ <input type="checkbox"/> ]Ditolak

#### **4.4.3 Kesimpulan Hasil Pengujian**

Pengujian yang telah dilakukan merupakan beberapa proses yang terdapat dalam aplikasi. Berdasarkan hasil pengujian diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang diharapkan. Namun demikian pengujian tersebut dapat dikatakan belum sempurna, dikarenakan hanya dilakukan pada satu sisi pengujian. Dari semua yang telah dilakukan dalam pengujian ini diharapkan dapat mewakili pengujian fungsi yang lainnya. Secara garis besar dari pengujian dan hasil implementasi perangkat lunak yang dibangun sudah sesuai dengan spesifikasi kebutuhan dan hasil perancangan perangkat lunak.

#### **4.5 Implementasi**

Implementasi merupakan pengaplikasian perangkat lunak, perangkat keras serta *database* yang digunakan. Pada tahapan ini akan dilakukan implementasi meliputi lingkungan implementasi, batasan implementasi dan konfigurasi dari kebutuhan sistem untuk dapat menjalankan aplikasi perangkat lunak yang telah dibangun.

##### **4.5.1 Implementasi Perangkat Lunak**

Perangkat lunak yang digunakan dalam pembuatan program penjualan dan pembelian sparepart pada bengkel *2Q-Speed Concept* ini adalah sebagai berikut:

1. Sistem operasi yang digunakan dalam pembangunan dan pengujian aplikasi ini adalah *Microsoft Windows 7*.

2. Bahasa pemrograman yang digunakan untuk membangun aplikasi ini adalah bahasa pemrograman java dengan *Java Development Kit 6.9* (JDK6.9).
3. Perangkat Pemograman Netbeans 7.0.1.
4. Xampp 1.7.7
5. Basis data dengan MySQL5.0.67.
6. Laporan menggunakan iReport-4.0.2 *for Windows*.
7. *Java Libraries*.

#### **4.5.2 Implementasi Perangkat Keras**

Berikut adalah perangkat keras yang digunakan dalam implementasi aplikasi transaksi penjualan dan service di bengkel *2Q-Speed Concept* ini dengan minimal spesifikasi adalah :

1. CPU Intel Dual Core 2.0 GHz.
2. Memory DDR RAM 1 GB.
3. Harddisk 120 GB.
4. Perangkat keluaran berupa *Monitor 14"*.
5. Perangkat masukan berupa *mouse* dan *keyboard*
6. dan perangkat printer

#### **4.5.3 Implementasi Basis Data**

Pada pembangunan aplikasi ini dibangun *database* agar tercipta sistem informasi yang terintegrasi dan dinamis. Selain itu pembuatan dan implementasi

*database* merupakan langkah yang memperlihatkan kesesuaian perancangan pada tahapan analisis dan perancangan terhadap pembangunan aplikasi. Berikut adalah basis data yang dibuat menggunakan bahasa SQL (MySQL) dalam aplikasi transaksi penjualan dan service di bengkel *2Q-Speed Concept*:

#### 1. Tabel Barang

```
CREATE TABLE IF NOT EXISTS `barang` (
  `kode_barang` varchar(10) NOT NULL,
  `nama_barang` varchar(35) NOT NULL,
  `harga_jual` int(11) NOT NULL,
  `harga_beli` int(11) NOT NULL,
  `stok` int(11) NOT NULL,
  `minimal` int(11) NOT NULL,
  `kode_jenis` varchar(5) NOT NULL,
  PRIMARY KEY (`kode_barang`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

#### 2. Tabel Detail Pembelian

```
CREATE TABLE IF NOT EXISTS `d_pembelian` (
  `kode_beli` varchar(10) NOT NULL,
  `kode_barang` varchar(10) NOT NULL,
  `jumlah` int(11) NOT NULL,
  `subtotal` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

## 3. Tabel Detail Pembelian

```
CREATE TABLE IF NOT EXISTS `d_pemesanan` (  
  `kode_pesanan` varchar(15) NOT NULL,  
  `kode_barang` varchar(10) NOT NULL,  
  `jumlah` int(11) NOT NULL  
  ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

## 4. Tabel Detail Penjualan

```
CREATE TABLE IF NOT EXISTS `d_penjualan` (  
  `kode_jual` varchar(10) NOT NULL,  
  `kode_barang` varchar(10) NOT NULL,  
  `jumlah` int(11) NOT NULL,  
  `subtotal` int(11) NOT NULL  
  ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

## 5. Tabel Detail Service

```
CREATE TABLE IF NOT EXISTS `d_service` (  
  `kode_service` varchar(10) NOT NULL,  
  `kode_barang` varchar(10) NOT NULL,  
  `jumlah` int(11) NOT NULL,  
  `subtotal` int(11) NOT NULL  
  ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

## 6. Tabel Jenis Sparepart

```
CREATE TABLE IF NOT EXISTS `jenis_sparepart` (  
  `kode_barang` varchar(10) NOT NULL,  
  `nama_barang` varchar(50) NOT NULL,  
  `jumlah` int(11) NOT NULL,  
  `subtotal` int(11) NOT NULL  
  ) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```

`kode_jenis` varchar(5) NOT NULL,
`nama_jenis_sparepart` varchar(25) NOT NULL,
PRIMARY KEY (`kode_jenis`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

#### 7. Tabel Pembelian

```

CREATE TABLE IF NOT EXISTS `pembelian` (
`kode_beli` varchar(10) NOT NULL,
`kode_pesanan` varchar(10) NOT NULL,
`tanggal` date NOT NULL,
`total` int(11) NOT NULL,
`status` varchar(15) NOT NULL,
PRIMARY KEY (`kode_beli`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

#### 8. Tabel Pemesanan

```

CREATE TABLE IF NOT EXISTS `pemesanan` (
`kode_pesanan` varchar(15) NOT NULL,
`tanggal` date NOT NULL,
`kode_supp` varchar(10) NOT NULL,
`status` varchar(15) NOT NULL,
PRIMARY KEY (`kode_pesanan`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

#### 9. Tabel Penjualan

```
CREATE TABLE IF NOT EXISTS `penjualan` (  
  `kode_jual` varchar(10) NOT NULL,  
  `tanggal` date NOT NULL,  
  `total` int(11) NOT NULL,  
  PRIMARY KEY (`kode_jual`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

#### 10. Tabel Service

```
CREATE TABLE IF NOT EXISTS `service` (  
  `kode_service` varchar(10) NOT NULL,  
  `tanggal_service` date NOT NULL,  
  `merk_motor` varchar(20) NOT NULL,  
  `flat_nomor` varchar(10) NOT NULL,  
  `keluhan` text NOT NULL,  
  `ongkos_service` int(11) NOT NULL,  
  `total` int(11) NOT NULL,  
  `mekanik` varchar(25) NOT NULL,  
  PRIMARY KEY (`kode_service`)  
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

#### 11. Tabel Supplier

```
CREATE TABLE IF NOT EXISTS `supplier` (  
  `kode_supp` varchar(15) NOT NULL,  
  `nama_supp` varchar(25) NOT NULL,
```



```

`alamat` text NOT NULL,
`telepon` varchar(15) NOT NULL,
PRIMARY KEY (`kode_supp`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

## 12. Tabel User

```

CREATE TABLE IF NOT EXISTS `user` (
`username` varchar(25) NOT NULL,
`password` varchar(25) NOT NULL,
`nama` varchar(25) NOT NULL,
`hak_akses` varchar(15) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

```

### 4.5.4 Implementasi Antar Muka

Antarmuka dalam sebuah aplikasi adalah hal yang sangat penting, aplikasi *desktop* yang menggunakan bahasa JAVA ini, implementasi antarmuka dilakukan dengan sebuah halaman yang berekstensi java. Implementasi antarmuka dari halaman yang dibuat dapat dilihat pada tabel di bawah ini.

#### 1. Implementasi *Login*

Berikut ini adalah tabel implementasi *login*:

**Tabel 4.14 Implementasi *Login***

<b>Menu</b>	<b>Deskripsi</b>	<b>File</b>
Menu <i>Login</i>	Tampilan Menu <i>Login</i> berisi inputan	MenuLogin.java

	<i>username, password dan hak akses login</i>	
<i>Login</i>	Tombol Masuk Sistem	jLogin.java

## 2. Implementasi Halaman Utama

Berikut ini adalah tabel implementasi halaman utama

**Tabel 4.15 Implementasi Halaman Utama**

<b>Menu</b>	<b>Deskripsi</b>	<b>File</b>
<i>File</i>	Menu <i>file</i> berisi menu <i>login</i> dan <i>logout</i>	<i>File.java</i>
Data Master	Menu data berisi sub menu data <i>user</i> , data mekanik, data supplier, data kategori dan data <i>sparepart</i>	FDataMaster.java
Laporan	Menu Laporan berisi sub menu Laporan barang, laporan pembelian, laporan penjualan dan laporan <i>service</i>	FDataLaporan.java
Transaksi	Menu transaksi berisi sub menu <i>form</i> pemesanan <i>sparepart</i> , <i>form sparepart</i> masuk, <i>form</i> penjualan dan <i>form</i> service	FDataTransaksi.java

## 3. Implementasi Halaman Sub Menu Data

Berikut ini adalah tabel implementasi halaman sub menu data

**Tabel 4.16 Implementasi Halaman Sub Menu Data**

<b>Menu</b>	<b>Deskripsi</b>	<b>Nama File</b>
Data <i>User</i>	Berisi informasi data <i>user</i> , menambah,	Fuser.java

	menghapus, mengedit data <i>user</i> yang dapat mengakses sistem	
Data Mekanik	Berisi informasi kode mekanik dan nama mekanik	Fmekanik.java
Data Supplier	Berisi informasi data supplier yang menjadi pemasok <i>sparepart</i> pada bengkel	FSupplier.java
Data Kategori	Berisi informasi kode jenis <i>sparepart</i> nama jenis <i>spare part</i> yang diperjual belikan dibengkel.	FJenisSparepart.java
Data <i>Sparepart</i>	Berisi informasi kode barang, nama barang, harga jual, harga beli, stok minimal dan stok tersedia.	FSparepart.java

#### 4. Implementasi Halaman Sub Menu Laporan

Berikut ini adalah tabel implementasi halaman sub menu laporan

**Tabel 4.17 Implementasi Halaman Sub Menu Laporan**

<b>Menu</b>	<b>Deskripsi</b>	<b>Nama <i>File</i></b>
Laporan Barang	Berisi informasi barang yang masih ada digudang	FBarang.java
Laporan Pembelian	<i>Form</i> untuk mencetak laporan pembelian yang dilakukan selama periode bulanan	FLapPembelian.java

Laporan Penjualan	<i>Form</i> untuk mencetak laporan transaksi penjualan <i>sparepart</i> yang dilakukan selama periode bulanan	FLapPenjualan.java
Laporan <i>Service</i>	<i>Form</i> untuk mencetak laporan transaksi <i>service</i> motor yang dilakukan selama periode bulanan	FLapService.java

#### 5. Implementasi Halaman Sub Menu Transaksi

Berikut ini adalah tabel halaman sub menu transaksi

**Tabel 4.18 Implementasi Halaman Sub Menu Transaksi**

<b>Menu</b>	<b>Deskripsi</b>	<b>Nama <i>File</i></b>
Menu Pemesanan <i>sparepart</i>	<i>Form</i> untuk menambah, atau melakukan pemesanan <i>sparepart</i> yang ditujukan untuk supplier	FPemesanan.java
Menu <i>Sparepart</i> Masuk	<i>Form</i> untuk mencatat <i>sparepart</i> yang masuk ke gudang sesuai dengan pesanan yang dilakukan sebelumnya	FSparepartMasuk.java
Menu penjualan <i>sparepart</i>	<i>Form</i> untuk mencatat, menambah transaksi penjualan yang terjadi setiap harinya	FPenjualan.java
Menu <i>Service</i> Motor	<i>Form</i> untuk mencatat transaksi <i>service</i> yang terjadi setiap harinya	FService.java

### 4.5.5 Implementasi Instalasi Program

Dalam mengimplementasikan program ke dalam komputer, maka sebelumnya harus dipersiapkan segala kebutuhan yang mendukung agar perangkat lunak dapat berjalan dengan baik pada *platform* teknologi yang kita pakai. Berikut ini adalah proses implementasi instalasi program:

#### 1. Klik Installer Aplikasi

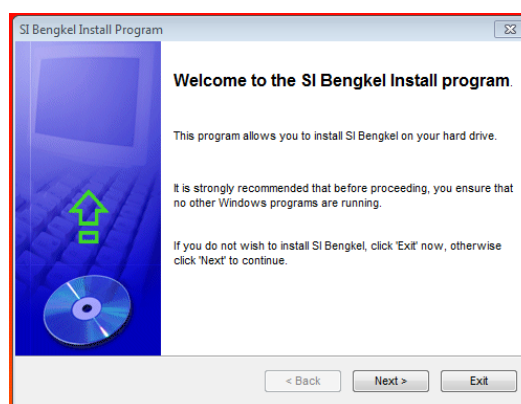
Double klik pada ikon installer seperti dibawah. Ini berfungsi untuk melakukan penginstalan sistem informasi bengkel motor ke dalam komputer.



**Gambar 4.37 Tampilan Proses Instalasi Program 1**

#### 2. Klik Next

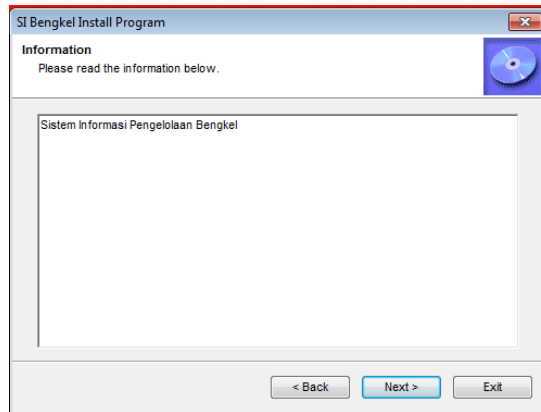
Setelah melakukan *double* Klik, akan muncul menu bar seperti dibawah. Klik *next* untuk melanjutkan penginstalan sistem informasi bengkel ini.



**Gambar 4.38 Tampilan Proses Instalasi Program 2**

### 3. Klik *Next*

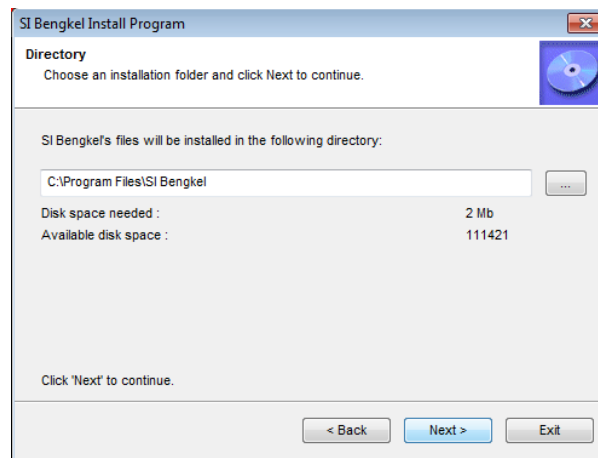
Setelah mengklik *next* pada halaman sebelumnya, akan muncul menu bar seperti dibawah. Klik *next* untuk melanjutkan instalasi sistem informasi ini.



**Gambar 4.39 Tampilan Proses Instalasi Program 3**

### 4. Klik *Next*

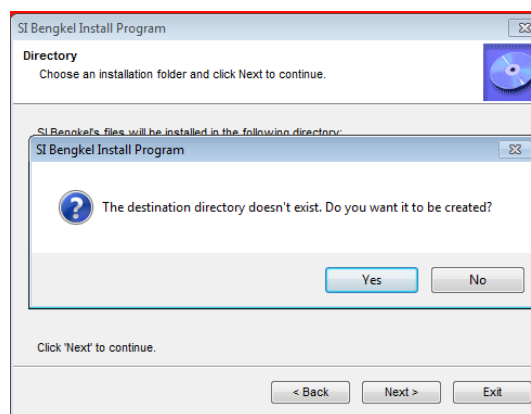
Setelah mengklik *next* dihalaman sebelumnya. Komputer akan menampilkan menu bar seperti dibawah. Ini untuk meletakkan file penginstallan akan diletakan di *drive* mana. Klik *next* pada halaman ini



**Gambar 4.40 Proses Instalasi Program 4**

## 5. Klik *Yes*

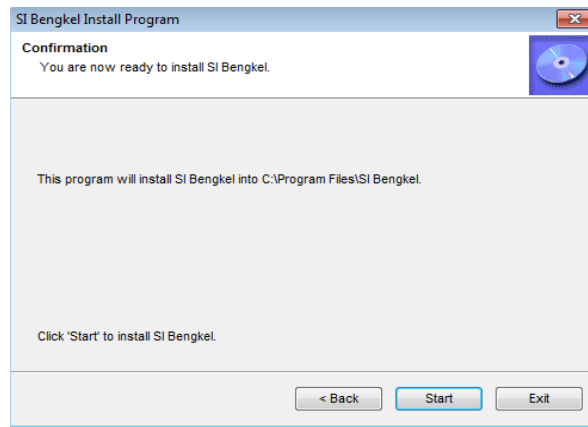
Setelah mengklik *next* untuk ke 3 kali pada halaman sebelumnya. Setelah itu akan muncul pesan seperti dibawah. Pesan ini memastikan apakah akan membuat directory destinasi baru untuk sistem informasi ini. Lalu klik *yes* seperti dibawah.



**Gambar 4.41 Tampilan Proses Instalasi Program 5**

## 6. Klik *Start*

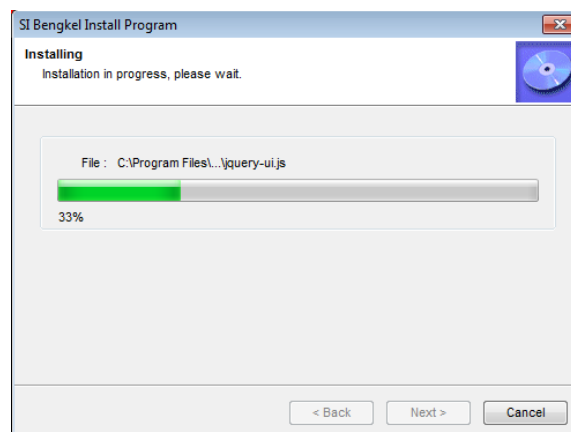
Setelah mengklik *yes* lalu mengklik *next* pada halaman sebelumnya komputer akan menampilkan halamn untuk memulai penginstallan sistem informasi ini.



**Gambar 4.42 Tampilan Proses Instalasi Program 6**

## 7. Tunggu Proses Instalasi Selesai

Setelah mengklik *start* pada halaman sebelumnya. Maka komputer akan memulai instalasi seperti dibawah. Jangan tutup halaman ini. Biarkan sampai garis berwarna hijau penuh.



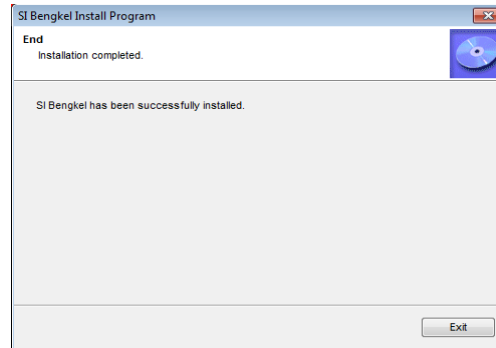
**Gambar 4.43 Tampilan Proses Instalasi Program 7**

## 8. Aplikasi Telah Dapat Digunakan

Setelah garis hijau penuh, komputer akan menampilkan halaman seperti dibawah. Ini berarti sistem informasi bengkel telah terinstall dikomputer. Dan



telah dapat digunakan.



**Gambar 4.44 Tampilan Proses Instalasi Tahap Akhir**

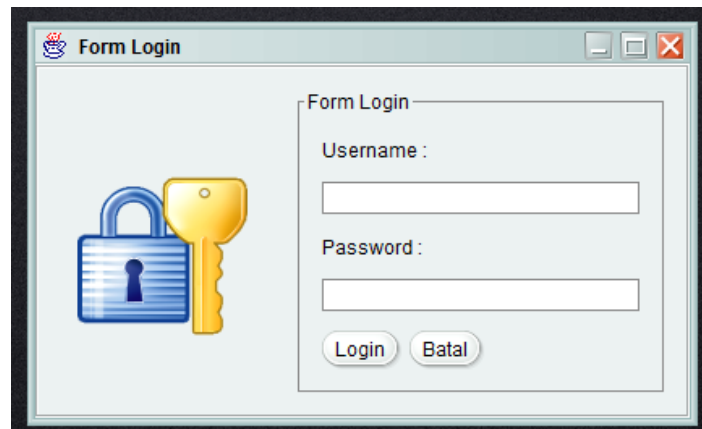
#### **4.5.6 Penggunaan Program**

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program di bengkel *2Q-Speed Concept*. Adapun cara pengoperasiannya adalah sebagai berikut:

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program di bengkel *2Q-Speed Concept*. Adapun cara pengoperasiannya adalah sebagai berikut:

1. *Form Login*

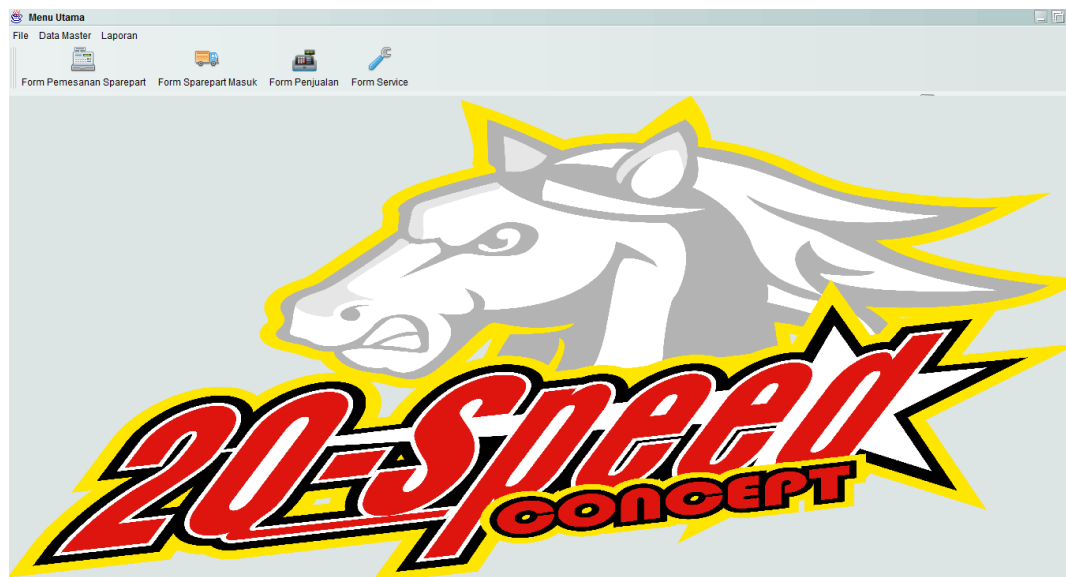
*Form login* ini berfungsi sebagai penentu bagi pengguna program sistem informasi, sehingga tidak sembarangan orang bisa mengakses program ini. ada 2 tombol pada *form* ini, tombol *login* untuk masuk kedalam *system* dan tombol batal untuk membatalkan *login* ke sistem:



**Gambar 4.45 Form Login**

## 2. Halaman Utama

Halaman utama setelah user seperti kasir, gudang dan pemilik login.

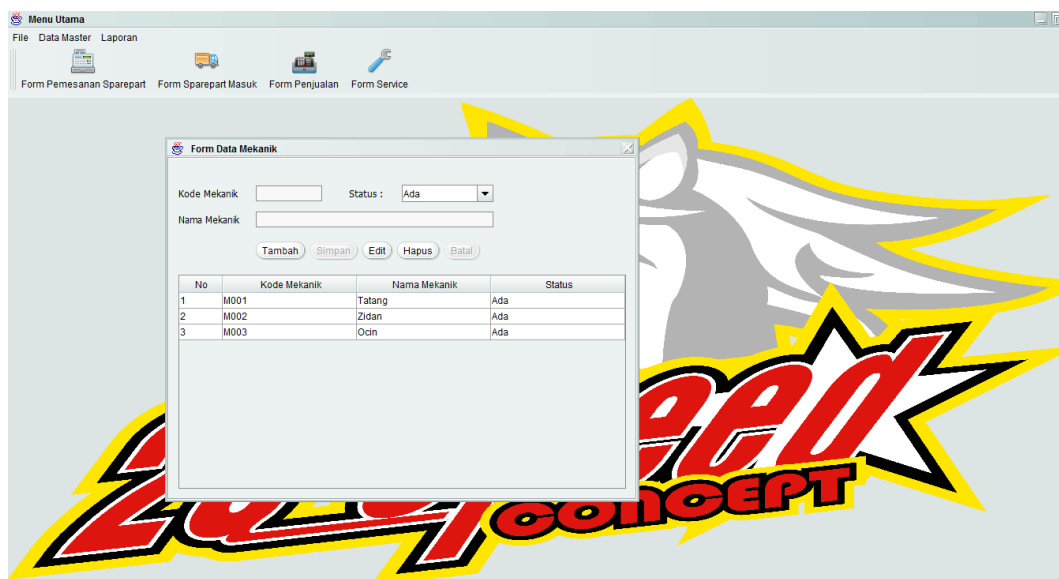


**Gambar 4.46 Halaman Utama**

## 3. Form Data Mekanik

Dalam menu data terdapat beberapa menu seperti data user, data mekanik, data *supplier*, data kategori dan data *sparepart*. Klik “Tambah” untuk menambah

data. Setelah *field* yang telah terisi maka klik “Simpan” untuk menyimpan data kedalam *database*. Untuk mengubah sebuah data, klik terlebih dahulu data yang akan diubah. Setelah data muncul pada *field* ubah data yang ingin diubah lalu klik “Ubah” pembaruan akan tersimpan secara otomatis. Begitu juga dengan hapus klik terlebih dahulu data yang akan dihapus, setelah data yang akan dihapus muncul pada *field* klik tombol hapus, maka data akan hilang dari tabel. Untuk membatalkan perubahan atau penambahan data klik tombol “Batal” maka perubahan atau penambahan akan dibatalkan.



**Gambar 4.47 Form Data Mekanik**

#### 4. Form Data *Supplier*

Dalam menu data terdapat beberapa menu seperti data user, data mekanik, data *supplier*, data kategori dan data *sparepart*. Klik “Tambah” untuk menambah data. Setelah *field* yang telah terisi maka klik “Simpan” untuk menyimpan data

kedalam *database*. Untuk mengubah sebuah data, klik terlebih dahulu data yang akan diubah. Setelah data muncul pada *field* ubah data yang ingin diubah lalu klik “Ubah” pembaruan akan tersimpan secara otomatis. Begitu juga dengan hapus klik terlebih dahulu data yang akan dihapus, setelah data yang akan dihapus muncul pada *field* klik tombol hapus, maka data akan hilang dari tabel. Untuk membatalkan perubahan atau penambahan data klik tombol “Batal” maka perubahan atau penambahan akan dibatalkan.

No	Kode Suppler	Nama Suppler	Alamat Suppler	No Telepon
1	KS001	Honda JKT	Jakarta Timur	089109090900
2	KS002	CV. SPR Honda	Bandung	081214870800

**Gambar 4.48 Form Data Supplier**

### 5. Form Data Sparepart

Dalam menu data terdapat beberapa menu seperti data user, data mekanik, data *supplier*, data kategori dan data *sparepart*. Klik “Tambah” untuk menambah data. Setelah *field* yang telah terisi maka klik “Simpan” untuk menyimpan data kedalam *database*. Untuk mengubah sebuah data, klik

terlebih dahulu data yang akan diubah. Setelah data muncul pada *field* ubah data yang ingin diubah lalu klik “Ubah” pembaruan akan tersimpan secara otomatis. Begitu juga dengan hapus klik terlebih dahulu data yang akan dihapus, setelah data yang akan dihapus muncul pada *field* klik tombol hapus, maka data akan hilang dari tabel. Untuk membatalkan perubahan atau penambahan data klik tombol “Batal” maka perubahan atau penambahan akan dibatalkan.

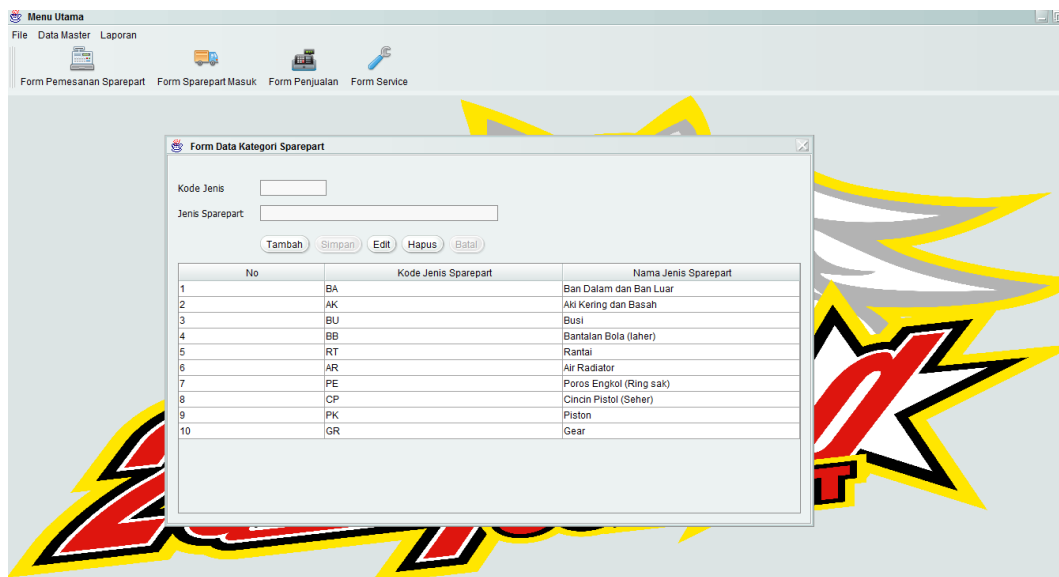
No	Kode Barang	Nama Barang	Harga Jual	Harga Beli	Minimal	Stok	Satuan	Kode Supplier
1	BA0001	Ban Dalam 17	72000	60000	10	23	Pcs	KS001
2	BA0002	Ban Tubles ukura...	27600	23000	5	27	Pcs	KS001
3	PK0001	Piston	17000	10000	100	100	Pcs	KS002
4	AK0001	AKI Astra	18000	15000	5	20	Pcs	KS002
5	SRT0001	Rantai SSS	130000	110000	10	0	pcs	KS002
6	FBA0003	Ban FDR MP76	360000	340000	5	0	pcs	KS002

**Gambar 4.49 Form Data Sparepart**

#### 6. Form Data Kategori Sparepart

Dalam menu data terdapat beberapa menu seperti data user, data mekanik, data *supplier*, data kategori dan data *sparepart*. Klik “Tambah” untuk menambah data. Setelah *field* yang telah terisi maka klik “Simpan” untuk menyimpan data kedalam *database*. Untuk mengubah sebuah data, klik terlebih dahulu data yang

akan diubah. Setelah data muncul pada *field* ubah data yang ingin diubah lalu klik “Ubah” pembaruan akan tersimpan secara otomatis. Begitu juga dengan hapus klik terlebih dahulu data yang akan dihapus, setelah data yang akan dihapus muncul pada *field* klik tombol hapus, maka data akan hilang dari tabel. Untuk membatalkan perubahan atau penambahan data klik tombol “Batal” maka perubahan atau penambahan akan dibatalkan.



**Gambar 4.50** *Form Data Kategori Sparepart*

## 7. *Form Pemesanan Sparepart*

Menu pemesanan yang berfungsi untuk melakukan pencatatan pemesanan *sparepart* yang habis atau stok minim. Untuk melakukan pencatatan pesanan gudang harus memilih kode *supplier* yang akan dituju untuk setiap pesanan *sparepart*. Lalu cari data barang yang akan dipesan dan klik data barang yang akan dipesan. Lalu isikan jumlah barang yang akan dipesan. Lakukan hal yang

sama sampai barang yang akan dipesan pada supplier tersebut tercatat. Jika sudah selesai klik tombol simpan pesanan, maka sistem akan menampilkan nota pemesanan yang harus dicetak dan diberikan pada pemilik untuk ditandatangani sebelum diberikan pada supplier.

The screenshot shows the 'Form Pemesanan Sparepart Supplier' interface. It includes the following elements:

- Form Fields:**
  - Kode Pemesanan: 2207201804
  - Tanggal: 2018/07/22
  - Kode Supplier: KS002 (dropdown)
  - Nama Supplier: CV. SPR Honda
  - Alamat: Bandung
  - Telepon: 081214870800
- Data Sparepart Table:**

No	Kode Bar...	Nama Barang	Harga Jual	Harga Beli	Minimal	Stok	Satuan
1	PK0001	Piston	17000	10000	100	100	Pcs
2	AK0001	Aki Astra	18000	15000	5	20	Pcs
3	SRT0001	Rantai SSS	130000	110000	10	0	pcs
4	FBA0003	Ban FDR M...	360000	340000	5	0	pcs
- INPUT SPAREPART YANG DIPESAN :**
  - Kode Barang: PK0001
  - Nama Barang: Piston
  - Jumlah: (empty field)
  - Buttons: Tambah, Hapus
- Daftar sparepart yang dipesan:** (Empty table with columns: Kode Barang, Nama Barang, Jumlah, Satuan)
- Buttons:** SIMPAN PEMESANAN

**Gambar 4.51 Form Pemesanan Sparepart**

## 8. Form Pencatatan Sparepart Masuk

Menu *sparepart* masuk ini berfungsi untuk melakukan pencatatan barang yang dikirim oleh *supplier*. Bertujuan untuk menyocokkan kecocokan data pesanan dengan yang dikirimkan. Bag.gudang menekan tombol pilih untuk memilih tanggal pemesanan untuk mencocokkan data yang dikirim. Jika jumlah dan barang sesuai tekan tombol simpan barang masuk. Jika jumlah pesanan dan jumlah pengiriman tidak sesuai maka klik data pada tabel dan ubah jumlah pada kolom jumlah terima dan tekan tombol simpan barang masuk.

**Gambar 4.52** *Form Pencatatan Sparepart Masuk*

### 9. *Form Penjualan Sparepart*

Kasir melakukan klik data barang yang ada pada kiri bawah *form*. Setelah data muncul pada *form* isi jumlah yang dibeli, lalu klik tombol tambah. Lakukan hal yang sama untuk setiap barang yang lain. Jika penjualan selesai klik tombol cetak nota penjualan untuk sebagai data konsumen dan sebagai data penjualan yang akan dijadikan laporan penjualan.



**Gambar 4.53 Form Penjualan Sparepart**

## 10. Laporan Stok Spare part

Berikut ini merupakan tampilan dari laporan stok *spare part*:

No	Kode Barang	Nama Sparepart	Jumlah Stok
1	BA0001	Ban Dalam 17	12 Pcs
2	BA0002	Ban Tubles ukuran 15	27 Pcs
3	PK0001	Piston	100 Pcs
4	AK0001	Aki Astra	15 Pcs

**Gambar 4.54 Laporan Stok Spare part**

## 11. Form Laporan Pembelian

Dalam menu laporan ini berfungsi untuk mencetak laporan pembelian setiap periodenya (biasanya dalam 1 bulan). Cara menggunakan *form* ini menginput periode laporan pembelian yang diinginkan ada yang perlu menginputkan tanggal periode terlebih dahulu.

**Gambar 4.55 Form Laporan Pembelian**

## 12. Laporan Pembelian

Beriku ini adalah tampilan laporan pembelian

**2Q-Speed Concept**  
Jln. Ciguruwk Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung

**Laporan Pembelian**

Periode : 2018/07/01 - 2018/07/31

Kode Pembelian : BM05072018001	Tanggal : 05/07/2018		
Nama Sparepart	Jumlah	Sub Total	
Piston	100	1,000,000.00	
Aki Astra	5	75,000.00	
<b>Total Pembelian :</b>		<b>1,075,000.00</b>	

Kode Pembelian : BM15072018002	Tanggal : 15/07/2018		
Nama Sparepart	Jumlah	Sub Total	
Ban Tubes ukuran 15	10	230,000.00	
<b>Total Pembelian :</b>		<b>230,000.00</b>	

**Total Pembelian Periode Ini : 2,380,000.00**

Mengetahui Pemilik, Mengetahui Gudang,

**Gambar 4.56 Laporan Pembelian**

### 13. Form Laporan Penjualan

Dalam menu laporan ini berfungsi untuk mencetak laporan penjualan setiap periodenya (biasanya dalam 1 bulan). Cara menggunakan *form* ini menginput periode laporan penjualan yang diinginkan ada yang perlu menginputkan tanggal periode terlebih dahulu.

**Gambar 4.57 Form Laporan Penjualan**

### 14. Laporan Penjualan

Berikut ini adalah tampilan laporan penjualan

**2Q-Speed Concept**  
Jln. Cigunuwik Kp.Babakan Sumedang, Cileunyi Kabupaten Bandung

**Laporan Penjualan**  
Periode : 2018/07/01 - 2018/07/31

Kode Penjualan	Tanggal	Nama Sparepart	Jumlah	Sub Total
INV05072018001	05/07/2018	Piston	1	17,000.00
			<b>Total Penjualan :</b>	<b>17,000.00</b>
<b>Total Penjualan Periode ini :</b>				<b>17,000.00</b>

Mengetahui Pemilik, \_\_\_\_\_  
Mengetahui Gudang, \_\_\_\_\_

**Gambar 4.58 Laporan Penjualan**

### 15. Form Laporan *Service*

Dalam menu laporan ini berfungsi untuk mencetak laporan *service* setiap periodenya (biasanya dalam 1 bulan). Cara menggunakan *form* ini menginput periode laporan *service* yang diinginkan ada yang perlu menginputkan tanggal periode terlebih dahulu.

**Gambar 4.59 Form Laporan Service**

### 16. Laporan *Service*

Berikut ini adalah tampilan laporan *service*

No service	Tanggal	Konsumen	Merk Motor	Plat Motor	Mekanik	Total Service
SV05072018001	05/07/2018	Rifi	Beat	F 14045	Dadang	75,000.00
1507201801	15/07/2018	Udin	Mio Matic Injection	F 14045	Tatang	105,000.00
2107201801	21/07/2018	Oki	Honda Beat/Manual	F 001	Kudis	105,000.00
<b>Total Pendapatan :</b>						<b>285,000.00</b>

**Gambar 4.60 Laporan Service**