BAB IV

ANALISIS DAN PERANCANGAN SISTEM

4.1 Analisis Sistem Yang Sedang Berjalan

Sebelum merancang suatu sistem, ada baiknya terlebih dahulu menganalisis sistem yang sedang berjalan ini bertujuan untuk mengetahui lebih jelas bagaimana cara kerja sistem tersebut dan masalah yang dihadapi sistem untuk dapat dijadikan landasan usulan perancangan sistem yang sedang berjalan.

Selama penulis melakukan analisis sistem yang sedang berjalan pada bengkel Indra Jaya Motor, penulis menemukan permasalahan pada sistem informasi pengolahan data yang berjalan yaitu pengelolaan pelayanan berupa jasa *service* dan penjualan suku cadang , dimana masih dicatat didalam buku besar dan untuk transaksi dicatat dalam nota, dihitung dengan kalkulator. Daftar *stock* barang ini masih dicatat dalam buku besar dan banyak arsip yang bertumpuk yang berkaitan dengan laporan penjualan, laporan pembelian.

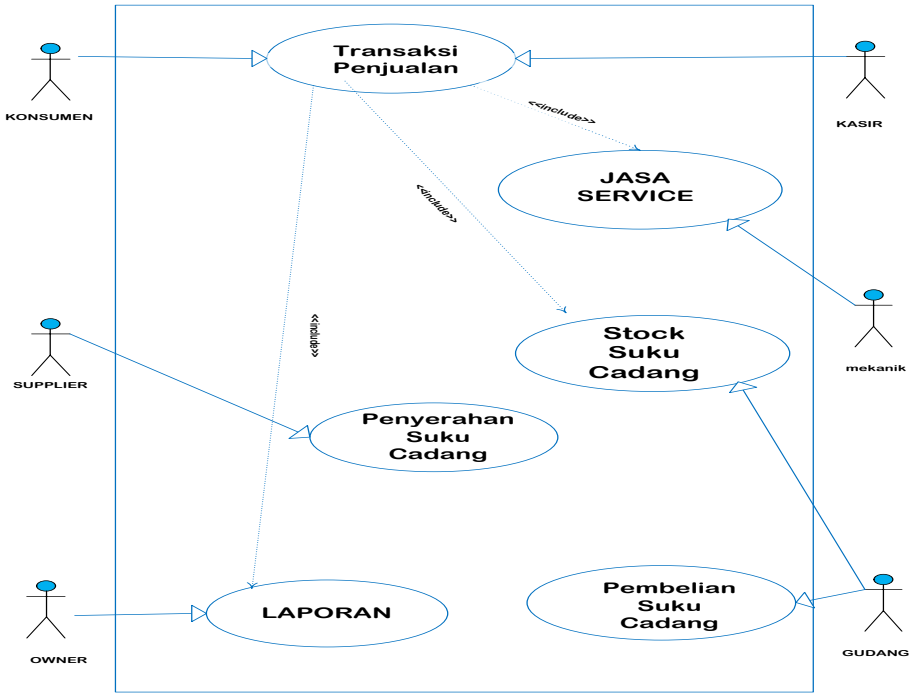
4.1.1 Analisis Prosedur Yang Sedang Berjalan

Sebelum membuat suatu sistem, hendaknya melakukan analisis terlebih dahulu terhadap kebutuhan-kebutuhan apa saja yang diperlukan di bengkel Indra Jaya Motor (IJM) dengan menggunakan metode-metode yang telah ada namun harus sesuai dengan ketentuan yang sudah di analisis di bengkel.

Dalam hal ini penulis menggunakan metodologi berorientasi objek, yaitu dimulai dari:

4.1.1.1 *Use Case* Diagram

Use Case Diagram memperlihatkan hubungan diantara aktor dan *use case*. Aktor merepresentasikan seorang *user* atau subsistem lain yang akan berinteraksi dengan sistem.



Gambar 4.1 *Use Case* Sistem Informasi Pelayanan yang Sedang Berjalan

4.1.1.2 Skenario *Use Case*

Skenario *use case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada tahap-tahap selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Adapun sekenario *use case* Ssistem Informasi Pelayanan di Bengkel Indra Jaya Motor yang sedang berjalan adalahb:

1. Nama Use Case : Jasa *service*

   Aktor : Mekanik

   Tujuan : melakukan jasa *service* sesuai permintaan konsumen.

Tabel 4.1 Skenario *Use Case* Jasa *service* yang sedang berjalan

Mekanik	Sistem
1. Menerima permintaan Jasa *Service*	
	2. Melakukan *Service* sesuai permintaan konsumen

2. Nama Use Case : Transaksi Penjualan

Aktor : Kasir

Tujuan : Untuk Melakukan penambahan barang yang kososng

Tabel 4.2 Skenario *Use Case* Transaksi Penjualan yang sedang berjalan

Kasir	Sistem
1. Kasir Membuat Nota Penjualan Jasa *service* dan suku cadang	
	2. Menghitung total bayar.
	3. Membuat nota dan membuat laporan penjualan.
4. Konsumen menerima nota dan membayar total bayar sesuai nota.	

3. Nama Use Case : *Stock* Suku Cadang

Aktor : Gudang

Tujuan : Menyimpan *Stock* Suku Cadang dan mengelola data *stock*

Tabel 4.3 Skenario *Use Case Stock* Suku Cadang yang sedang berjalan

Gudang	Sistem
1. Mengelola *stock* suku cadang di bengkel Indra Jaya Motor	
	2. Membuat pembukuan, pemasukan dan pembelian *stock* suku cadang

4.Nama Use Case : Pembelian Suku Cadang

Aktor : Gudang

Tujuan : Membeli *stock* barang yang sudah habis digudang

Tabel 4.4 Skenario *Use Case* Pembelian yang sedang berjalan

Gudang	Sistem
1. Memeriksa barang yang kosong	

digudang	
	2. Melakukan pemeriksaan barang yang sudah habis .
3. Mendatangi dan membeli barang yang sudah diperiksa.	

5.Nama Use Case : Penyerahan Suku Cadang

Aktor : *supplier*

Tujuan : Menyerahkan barang yang dibeli oleh pihak bengkel

Tabel 4.5 Skenario *Use Case* Penyerahan yang sedang berjalan

Supplier	Sistem
1. Memeriksa catatan yang berisi pemasanan dan menghitung total bayar	
	2. Menghitung dan menyerahkannya

6.Nama Use Case : Laporan

Aktor : *Owner*

Tujuan : Menerima Laporan dari setiap Bagian

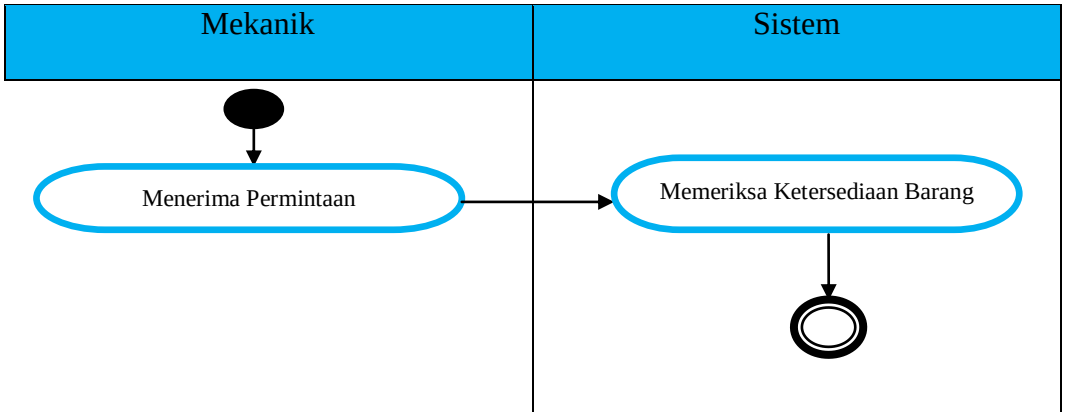
Tabel 4.6 Skenario *Use Case* Laporan yang sedang berjalan

Owner	Sistem
1. Menerima Laporan	
	2. Menyimpan dan mengarsipkan Laporan

4.1.1.3 *Activity Diagram*

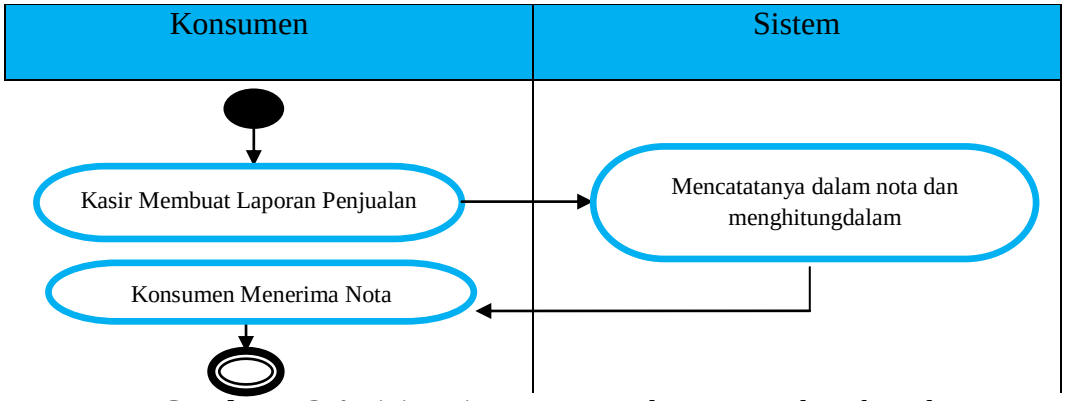
Activity diagram digunakan untuk menggambarkan kegiatan-kegiatan yang ada di dalam suatu sistem. maka perlu dibuatkan *activity diagram* tentang sistem yang sedang berjalan, yaitu:

1. Deskripsi :Mekanik menerima permintaan jasa *service* kemdian melakukan sesuai dengan yang diinginkan konsumen.



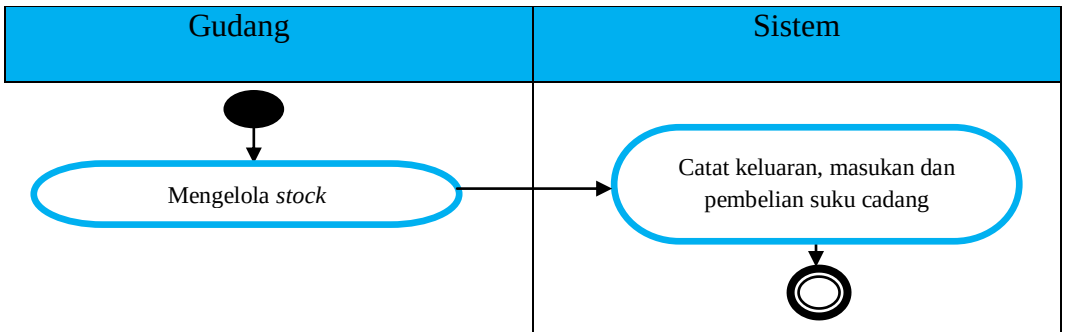
Gambar 4.2 *Activity Diagram* Jasa *service* yang sedang berjalan

2. Deskripsi : Kasir melakukan transaksi penjualan setelah konsumen memesan.



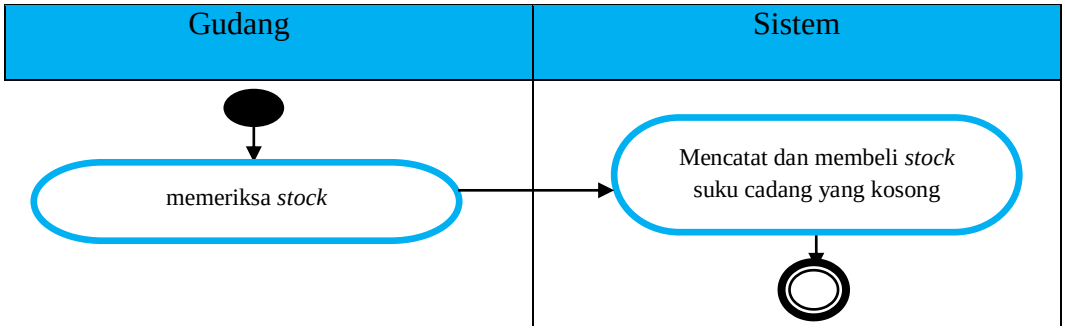
Gambar 4.3 *Activity Diagram* Penjualan yang sedang berjalan

3. Deskripsi : Bagian Gudang mengelola *stock* suku cadang



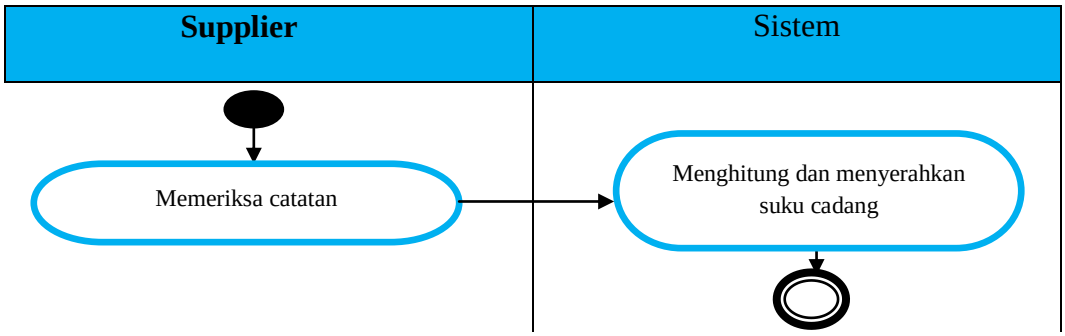
Gambar 4.4 *Activity Diagram Stock* Suku Cadang yang sedang berjalan

4. Deskripsi : setelah barang habis di gudang maka bagian gudang membeli suku cadang atau barang yang akan habis atau yang sudah habis ke *supplier*.



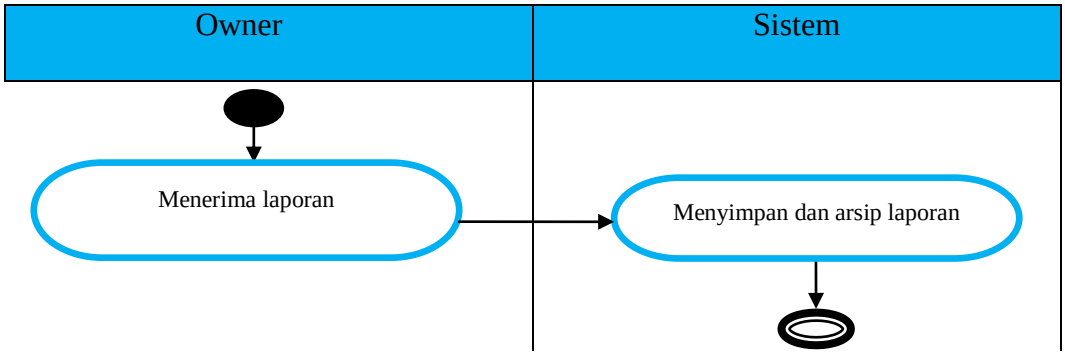
Gambar 4. 5 *Activity Diagram* Pembelian yang sedang berjalan

5. Deskripsi : *supplier* menghitung dan menyerahkan pesanan bengkel.



Gambar 4.6 *Activity Diagram* Penyerahan yang sedang berjalan

6. Deskripsi : *Owner* menerima laporan catatan yang sudah dilakukan oleh setiap bagian.



Gambar 4.7 *Activity Diagram* Laporan yang sedang berjalan

4.1.2 Evaluasi Sistem Yang Berjalan

Setelah melakukan analisis terhadap sistem yang sedang berjalan, baik analisis terhadap prosedur yang ada, penulis menemukan beberapa permasalahan baik itu secara langsung maupun tidak langsung.

Table 4.7 Evaluasi Sistem Yang Sedang Barjalan

NO	MASALAH	SOLUSI
1	Proses Pengelolaan dan penghitungan transaksi Pelayanan jasa *service* dan suku cadang untuk konsumen dan Pembelian suku cadang kepada *supplier* dilakukan dengan ditulis dalam sebuah buku kemudian proses penghitungannya masih mengandalkan kalkulator	Merancang aplikasi agar dapat memberikan kemudahan dalam melakukan transaksi dan dalam mengelola data dan memiliki *database* agar terintegrasi
2	Kurangnya informasi pemantauan *stock* barang suku cadang dan belum efektifnya pengecekan ketersediaan yang ada pada bagian gudang.	Dibangun aplikasi yang mengetahui *stock* barang minimum agar lebih efektif dalam ketersediaan suku cadang.
3	Seringkali pihak pengelola kebingungan dalam mengatur arsip hasil laporan yang berceceran dan menumpuk dari bagian pelayanan jasa *service*, penjualan suku cadang, pembelian suku cadang, *stock* barang dan transaksi.	Dibangunnya sebuah aplikasi yang berbasis *client-server* agar data lebih akurat sehingga tidak sulit dalam pembuatan laporannya.

4.2 Perancangan Sistem

Perencanaan sistem ini menyangkut estimasi dari kebutuhan-kebutuhan fisik, tenaga kerja dan dana yang dibutuhkan untuk mendukung pengembangan sistem.

4.2.1 Tujuan Perancangan Sistem

Perancangan sistem yang akan diaplikasikan ini bertujuan untuk memberikan gambaran secara umum kepada pengguna tentang sistem yang akan

dibangun. Tujuan perancangan sistem informasi pelayanan ini akan diuraikan sebagai berikut :

1. Untuk mengatasi masalah yang terjadi pada sistem yang lama, Merancang aplikasi agar dapat memberikan kemudahan dalam melakukan transaksi dan dalam mengelola data, data *stock* barang.
2. Tersimpan dalam database terintegrasi dengan baik dan aman.
3. Dengan adanya Sistem Informasi Pelayanan berbasis *Client-Server* pada Bengkel Indra Jaya Motor, diharapkan dapat meningkatkan pelayanan dan memberikan kemudahan dalam proses pengelolahan data.

4.2.2 Gambaran Umum Sistem yang Diusulkan

Gambaran umum dari sistem yang diusulkan adalah untuk memberikan gambaran kepada pengguna program (*user*). Pada bagian kasir, tidak perlu menghitung dengan kalkulator dan menuliskan data-data barang pada sebuah nota, karena sistem ini mempunyai fungsi pelayanan transaksi .

Pada bagian Gudang, tidak perlu menghitung stock digudang karena sistem ini mempunyai data *stock* barang, dan sistem akan memberi mempunyai peringatan untuk *stock* minimum barang. Pada bagian *Owner* dapat menggunakan seluruh fungsi dan melihat laporan.

4.2.3 Perancangan Prosedur Yang Diusulkan

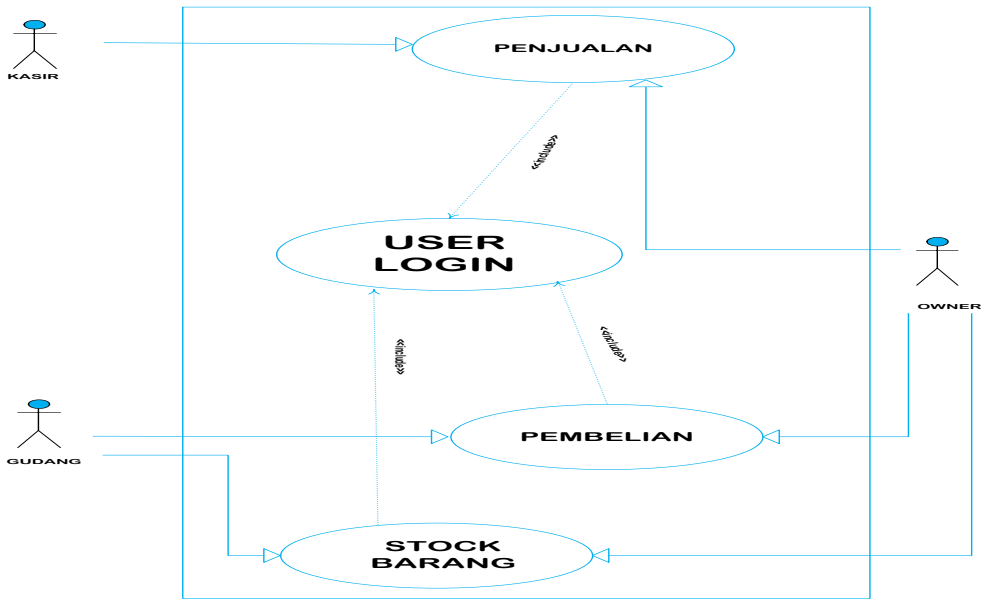
Perancangan ini mencakup *use case diagram,* Skenario *Use Case, activity diagram, sequence diagram, class diagram, component diagram,* dan *deployment diagram* yang menghasilkan sistem lebih baik

.

4.2.3.1 *Uses Case*

Use case adalah gambaran *fungsionalitas* dari suatu sistem, sehingga pengguna sistem paham dan mengerti mengenai kegunaan sistem yang akan dibangun pada bengkel Indra Jaya Motor.

Berikut adalah usecase yang diusulkan dalam merancang sistem informasi pelayanan pada bengkel Indra Jaya Motor Bandung :



Gambar 4.8 *Use Case* Diagram Sistem Informasi Pelayanan di Bengkel Indra Jaya Motor yang Diusulkan

4.2.3.2 Skenario *Uses case*

Skenario *Use Case* digunakan untuk memudahkan dalam menganalisa skenario yang akan kita gunakan pada fase-fase selanjutnya dengan melakukan penilaian terhadap skenario tersebut. Berikut skenario *use case* yang diusulkan :

1. Nama Use Case : *User Login*

   Aktor : *User* / *Admin(owner)*, Kasir, Gudang

   Tujuan : Proses *Validasi Akun User*

Tabel 4.8 Skenario *Use Case User Login*

User	Sistem
1. Masuk ke halaman *login user*	
2. *M*engisi *user* dan *password* pada form *login* akun member.	
	3. Memverifikasi *User* dan *password* pada sistem database.
	4. Jika *account* terdaftar, maka secara otomatis akan masuk ke halaman utama, jika tidak ada data yang dimaksud maka sistem akan memberikan peringatan *user name atau password* salah dan kembali ke halaman *login user*.

2. Nama Use Case : Penjualan

   Aktor : Kasir

   Tujuan : melakukan transaksi penjualan jasa *service* dan suku cadang

Tabel 4.9 Skenario *Use Case* Penjualan

Kasir	Sistem
1. Masuk ke menu transaksi penjualan	
	2. Menampilkan halaman transaksi penjualan untuk input data-data penjualan
3. Input kepada, Kode Barang, Banyaknya barang dan Uang bayar	
	4. Menampilkan rincian pembayaran
5. Pilih atau *klik* simpan untuk menyimpan ke database	
	6. Menyimpan data *valid* ke *database*
	7. tampil ‘Penambahan Sukses’ kemudian tampil struk penjualan

3. Nama Use Case : Pembelian

   Aktor : Gudang

   Tujuan : Pembuatan PO apabila keadaan barang *stock* minimum kemudian melakukan pembelian barang.

Tabel 4.10 Skenario *Use Case* pembelian

Gudang	Sistem
	1. Peringatan Stock Minimum tampil
2. Masuk ke menu *PurchaseOrder* untuk membuat form pembelian barang	
	3. Menampilkan halaman *PurchaseOrder*
4. *preview PurchaseOrder klik* untuk menampilkan barang yang harus dibeli, pilih *supplier* dan buat *form PurchaseOrder*	
	5. Menyimpan *PurchaseOrder*
	6. Menampilkan Form *PurchaseOrder*

4. Nama Use Case : *stock* barang

   Aktor : Gudang

   Tujuan : Tahap berikutnya dari *PurchaseOrder* prosesnya sama seperti memasukan *stock* barang ke penyimpanan gudang.

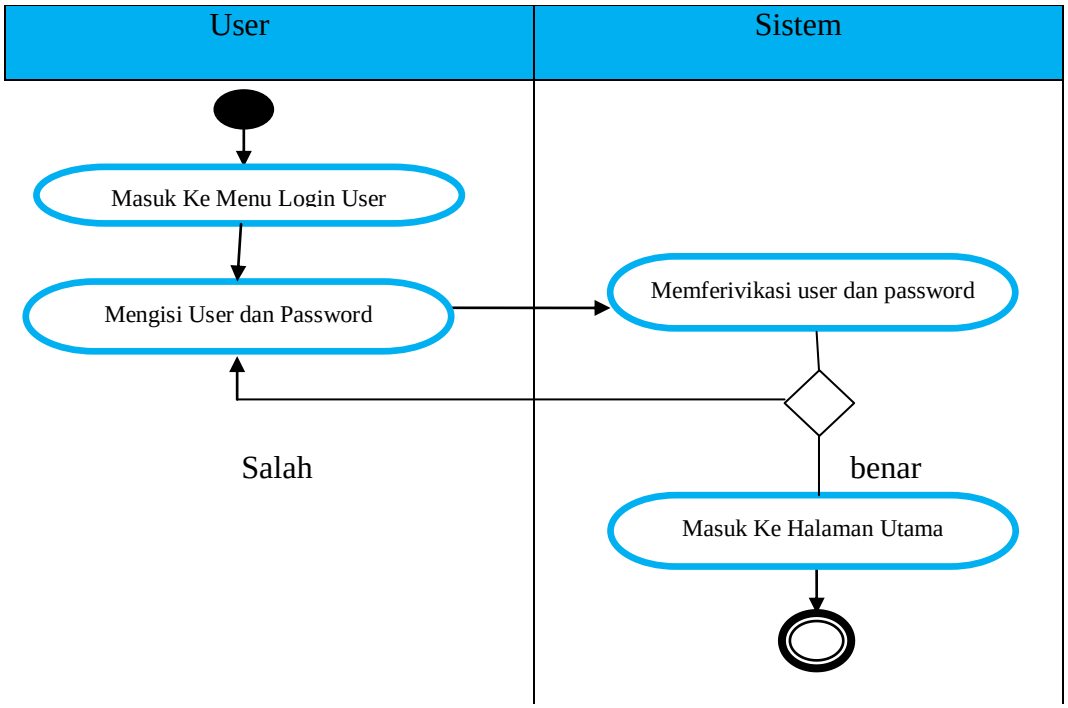
Tabel 4.11 Skenario *Use Case stock* Barang

Gudang	Sistem
1. Setelah barang datang Masuk ke menu *Verifikasi PurchaseOrder*	
	2. Menampilkan Halaman Restock barang
3. Inputkan Kode PO	
	4. Tampil Rincian PO dan Simpan apabila sesuai.
5. Apabila ada yang berubah maka edit data-data sesuai dengan perubahan	
	6. Menampilkan Rincian *Stock* barang yang sudah dibeli berdasarkan *PurchaseOrder* atau edit perubahan dan menyimpan data *valid* ke *database.*

4.2.3.3 *Activity Diagram*

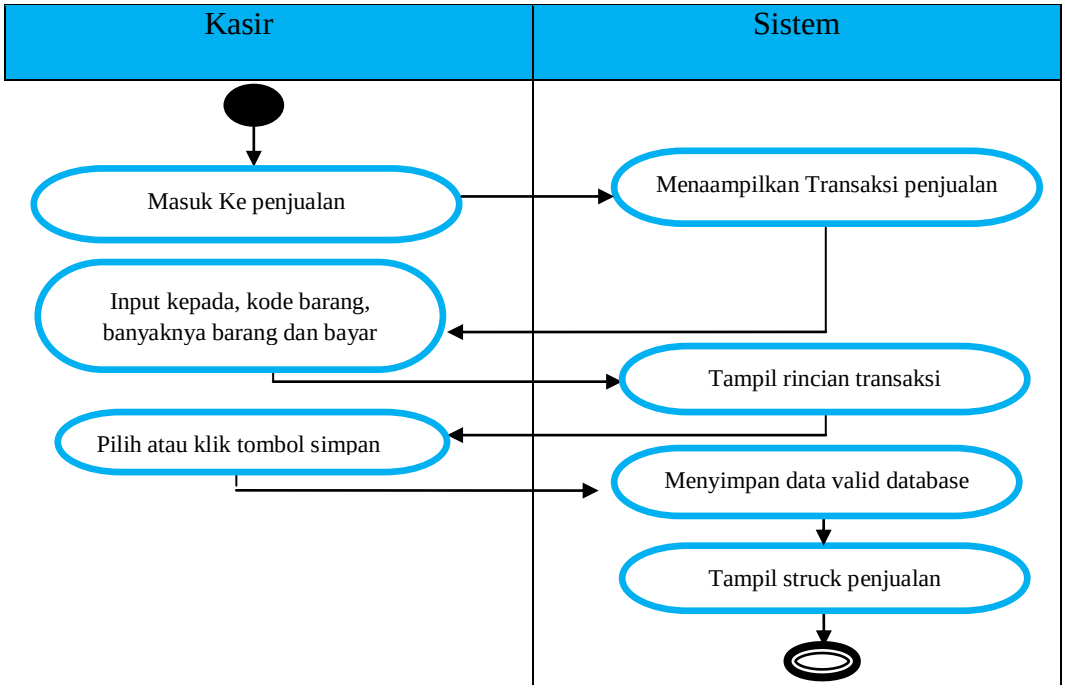
Activity diagram berikut ini memperlihatkan secara rinci aliran data secara logika tanpa mempertimbangkan lingkungan fisik dimana data mengalir.

1. Deskripsi: *User owner*, kasir dan gudang melakukan *login*, Jika *account* terdaftar, maka secara otomatis akan masuk ke halaman utama, jika tidak ada data memberikan peringatan *user name* atau *password* salah kemudian *user* dapat melakukan pengisian *username* dan *password* kembali.



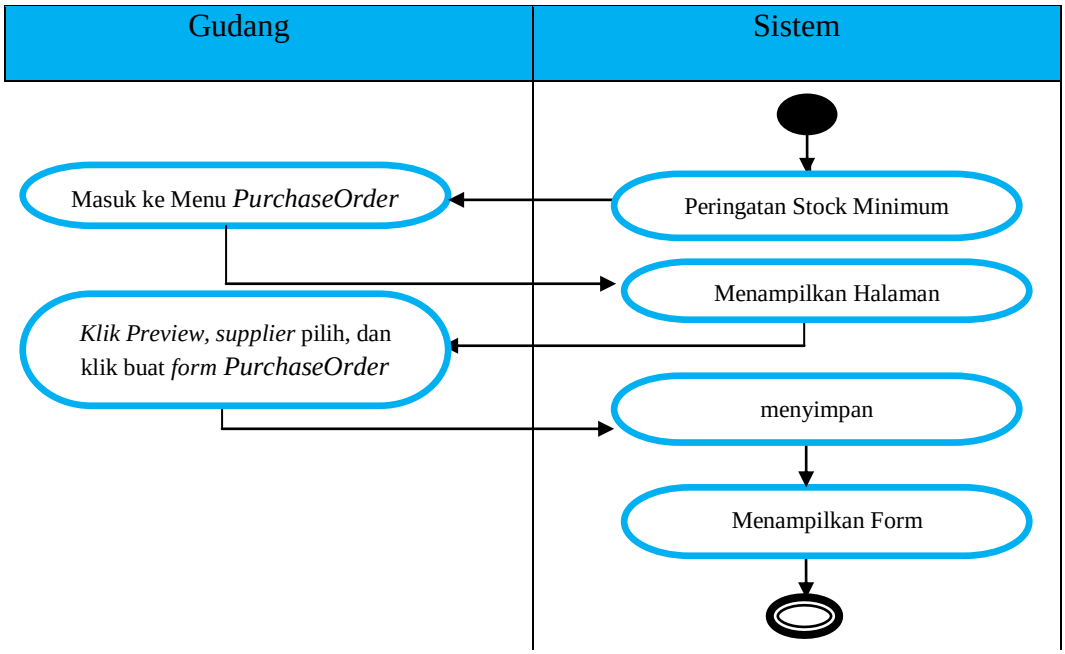
Gambar 4.9 *Activity Diagram User Login* Yang Diusulkan

2. Deskripsi: kasir melakukan penjualan, transaksi penjualan kepad konsumen sistem akan meproses sendirinya kemudian tersimpan otomatis di databsse dan tampil *struk* untuk diberikan kepada konsumen dan laporan untuk owner, *owner* dapat melakukan proses transaksi juga namun tidak *owner* pergunakan karena dia sebagai admin yang memiliki sistem in, sebagai pembuat laporan dan mengelola data penambahan atau pengeditan data.



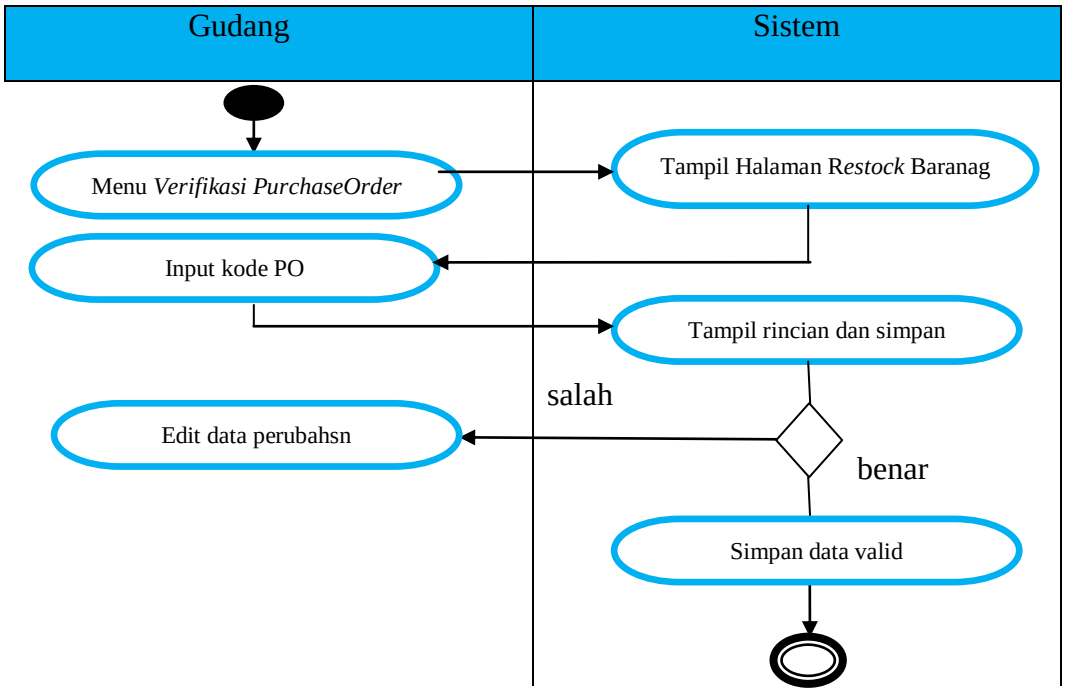
Gambar 4.10 *Activity Diagram* Penjualan yang Diusulkan

3. Deskripsi: Gudang dapat melakukan pembelian dengan membuat *PurchaseOrder* pada barang yang sudah mencapai *stock* minimum dan membuatkan formulir untuk membeli barang tersebut dan dapat mengedit



Gambar 4.11 *Activity Diagram* Pembelian Yang Diusulkan

4. Deskripsi: gudang dapat melakukan *stock* barang ini untuk menambahkan barang yang ada atau barang yang sudah dibeli, Tahap berikutnya dari *PurchaseOrder* adalah *verifikasi PurchaseOrder* prosesnya sama seperti *restock* jika barang yang usdah dibeli oleh bagian gudang berdasarkan *form PurchaseOrder* apabila ada perubahan akan bisa diedit data-data yang dirubahnya kemudian disimpan ke dalam database dan barang akan otomatis bertambah.



Gambar 4.12 *Activity Diagram stock* barang Yang Diusulkan

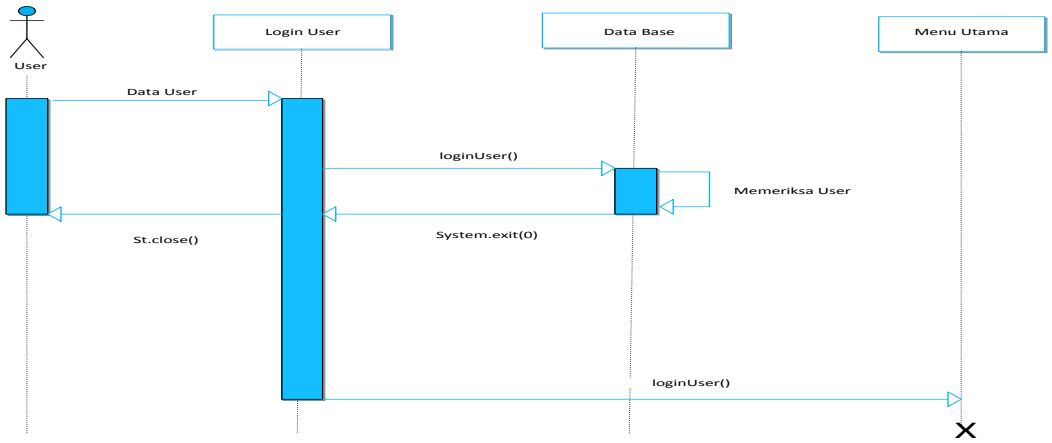
4.2.3.4 *Sequence Diagram*

Sequence Diagram digunakan untuk menggambarkan interaksi antar objek dalam waktu yang berurutan. Tetapi pada dasarnya *sequence Diagram* selain digunakan dalam lapisan abstraksi model objek.

Berikut adalah *sequence diagram* yang ada pada sistem Informasi Pelayanan di Bengkel Indra Jaya Motor, yaitu:

1. ***Squence Diagram User Login***

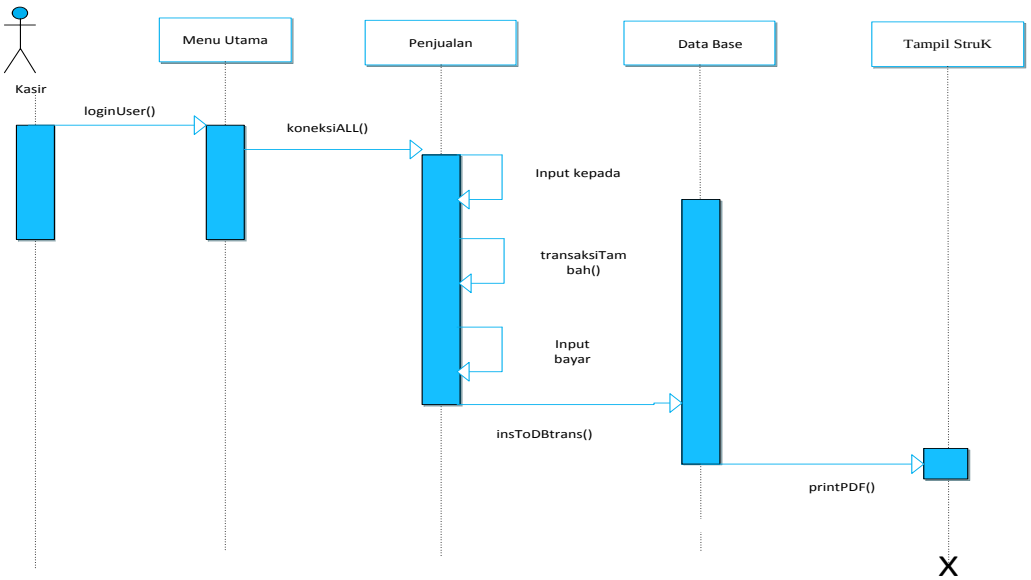
Pengguna Sistem pada bengkel Indra Jaya Motor Harus melewati *login* ,Tampilan aplikasi yang muncul kali pertama adalah *form login* yang meminta user aplikasi untuk memasukkan *User* dan *Password*.



Gambar 4.13 Squence Diagram *User Login*

2. ***Squence Diagram* Untuk Melakukan Transaksi Penjualan**

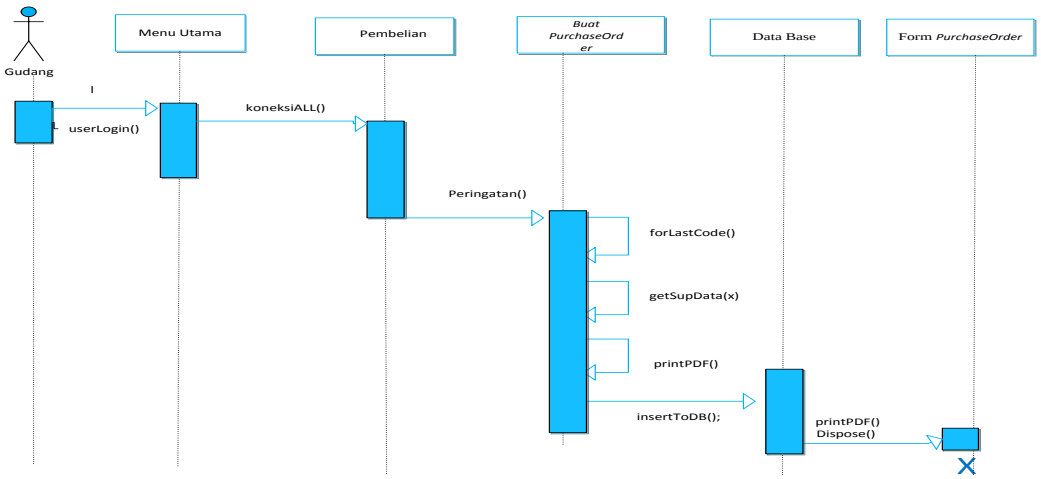
Bagian Kasir melakukan transaksi penjualan dengan masuk ke menu transaksi penjualan. Kemudian akan disimpan ke *database* secara otomatis.



Gambar 4.14 *Squence Diagram* Penjualan

3. *Squance Diagram* Membuat Pembelian dari *Stock* Minimum

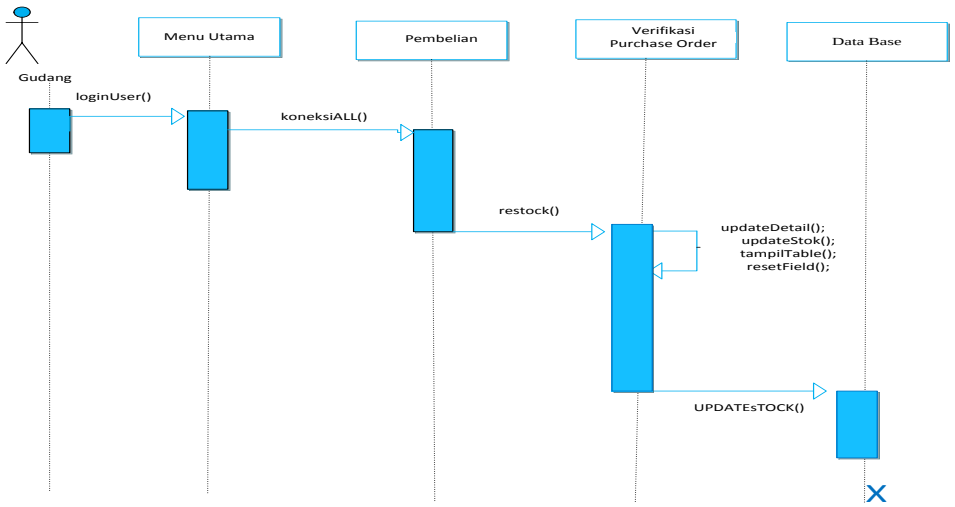
Bagian gudang akan melakukan *PurchaseOrder* sebelum membeli suku cadang ke s*upplier*, ketika peringatan barang minimum muncul maka gudang akan dapat membuat *PurchaseOrder* data yang tampil sesuai *stock* minimum.



Gambar 4.15 *Squence Diagram* Pembelian

4. *Squance Diagram Stock* Gudang

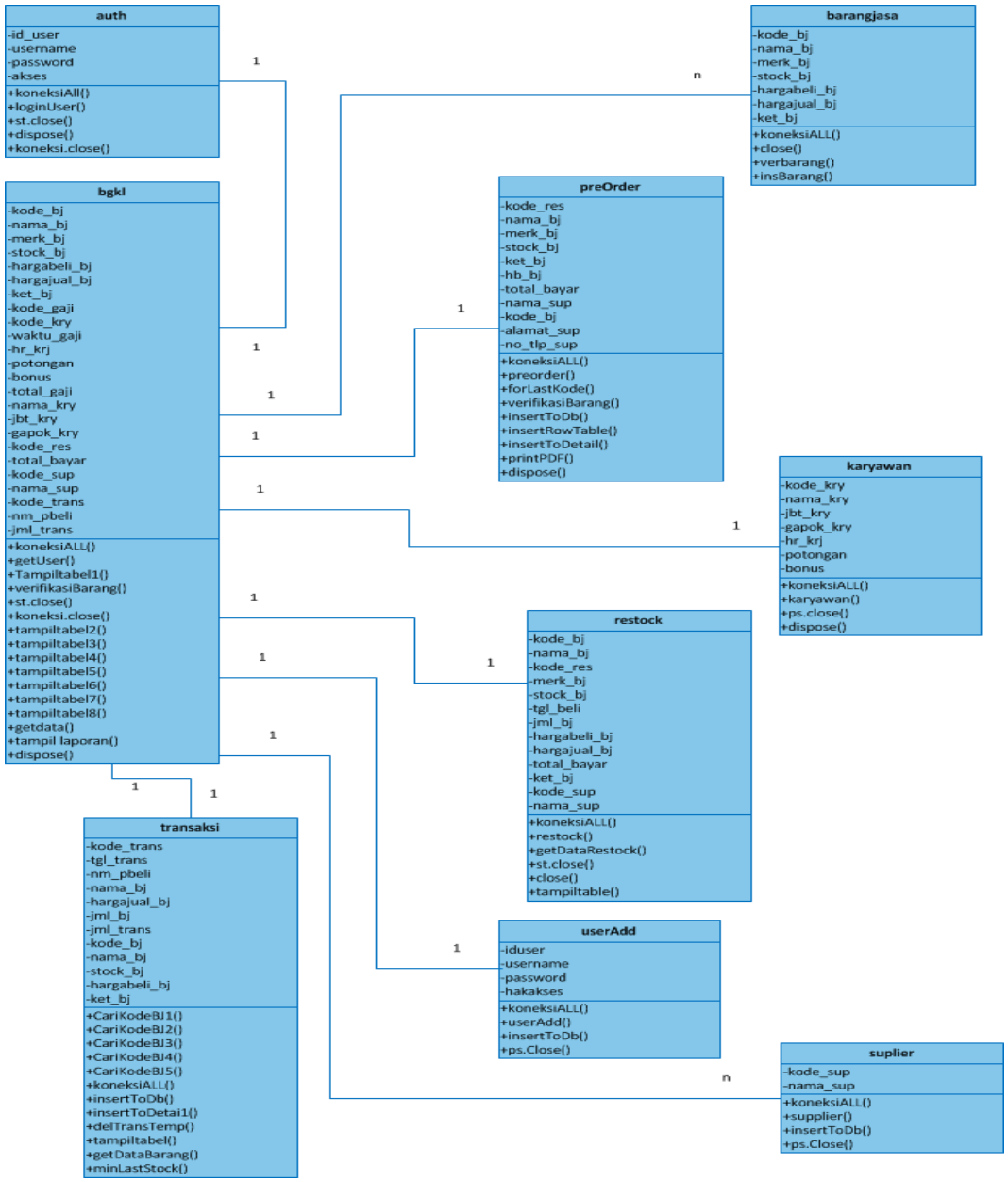
Barang atau suku cadang yang sudah ada atau sudah dibeli berdasarkan form PO diatas akan dimasukan datanya dengan *Verifikasi PurchaseOrder*, proses ini sama seperti *restock* barang hanya tinggal input kode PO dan simpan.



Gambar 4.16 *Squence Diagram Stock*

4.2.3.5 *Class Diagram*

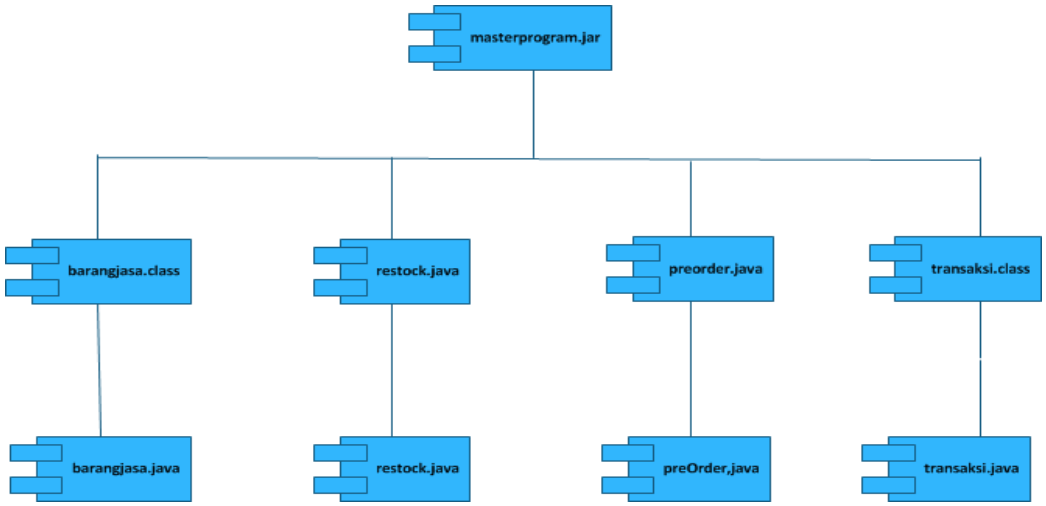
Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut *diagram class* pada sistem informasi pelayanan di bengkel indar jaya motor.



Gambar 4.17 *Class Diagram* Sistem Informasi Pelayanan

4.2.3.6 Component Diagram

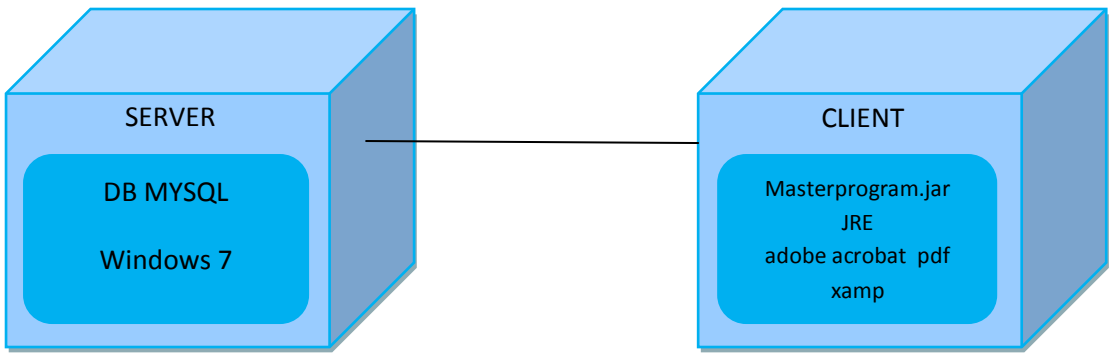
Component Diagram adalah unit fisik yang nyata yang menjadi bagian dari *deployment independent*. *Component* ini diimplementasikan meskipun pada sistem yang kecil. Perancangan menu pada aplikasi yang diusulkan digambarkan oleh gambar *component diagram* berikut.



Gambar 4.18 *Component Diagram* Sistem Informasi Pelayanan

4.2.3.7 Deployment Diagram

Asumsi yang digunakan pada perangkat lunak ini memakai sistem *client-server* untuk menunjukan konfigurasi komponen dalam proses eksekusi aplikasi.

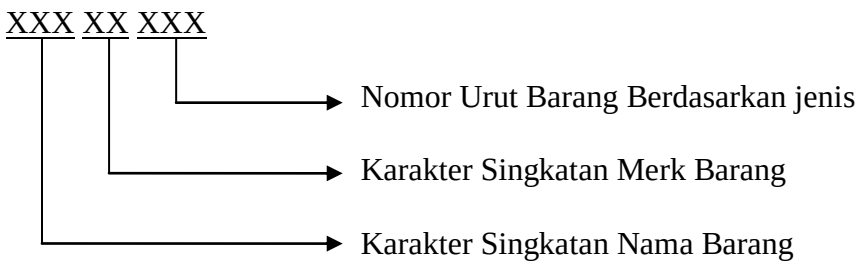


Gambar 4. 19. *Deployment Diagram* Sistem Informasi Pelayanan

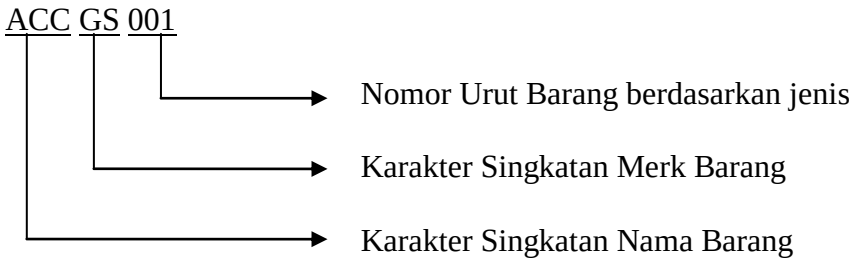
4.2.3.8 Kodefikasi

Kodefikasi ini digunakan sebagai identitas untuk setiap data yang akan diinputkan. dalam hal ini akan di jelaskan mengenai kode-kode yang akan digunakan.

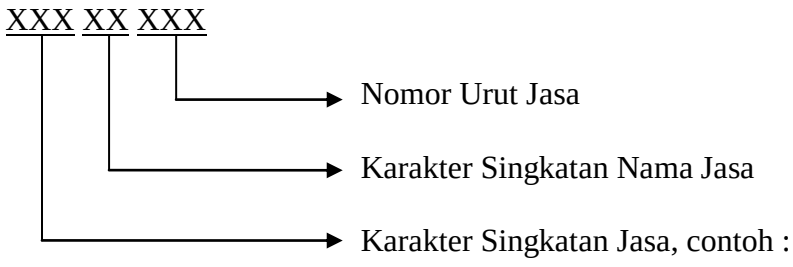
1. Kode Barang

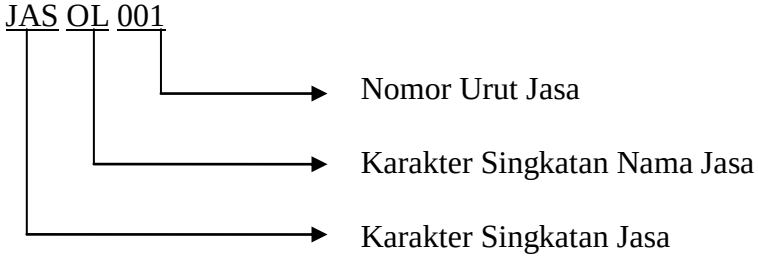


Contoh :



2. Kode Jasa





4.2.4 Perancangan Antar Muka

Perancangan antar muka dirancang untuk memudahkan pemakai *(user)* dalam mengakses informasi-informasi yang dibutuhkan. Adapun perancangan antar muka pada Sistem Informasi di Indra Jaya Motor adalah sebagai berikut:

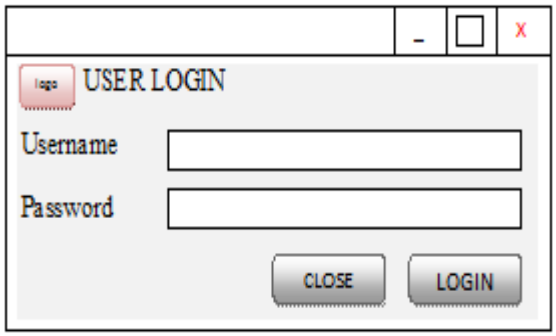
4.2.4.1 Perancangan *Input*

Perancangan input terdiri dari 4 ,yaitu Perancangan input *Login User, owner,* Kasir dan Gudang .

1. **Perancangan *Login User (owner, kasir, gudang)***

   a. **Halaman *Login User (owner, kasir, gudang)***

   Halaman *Login user* ini merupakan tahapan awal melakukan proses selanjutnya.



Gambar 4.20 Tampilan Halaman *Login user*

2. **Perancangan *Input* Kasir**

   a. **Halaman Transaksi Penjualan**

   Halaman Transaksi ini digunakan oleh kasir untuk melakukan transaksi penjualan jasa *service* dan suku cadang.



Gambar 4.21. Tampilan Halaman Transaksi Penjualan

3. Perancangan *Input* Gudang

a. Halaman *Purchase Order*

halaman ini berfungsi untuk melakukan PO jika terjadi *stock* minimum.



Gambar 4.22. Tampilan *Purchase Order*

b. Halaman *Verifikasi Purchase Order*

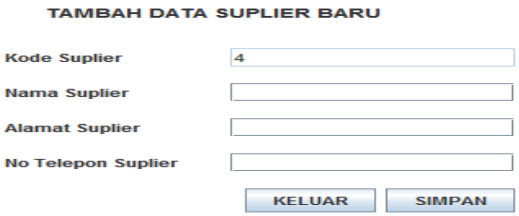
Barang atau suku cadang yang sudah ada atau sudah dibeli berdasarkan form PO diatas akan dimasukan datanya dengan *Verifikasi Purchase Order*, proses ini sama seperti *restock* .



Gambar 4.23. Tampilan *Verifikasi Purchase Order*

c. Halaman Tambah Data *Supplier*

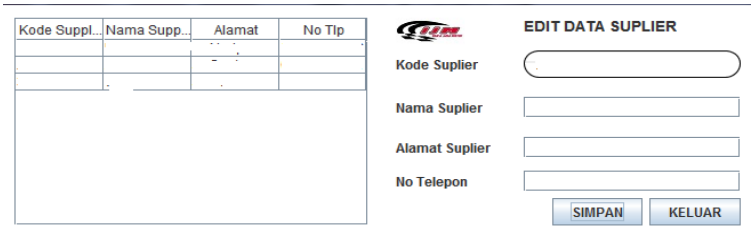
Halaman ini berfungsi untuk menambahkan data *supplier* baru.



Gambar 4.24. Tampilan Halaman Tambah Data *Supplier*

d. Halaman Edit Data *Suplier*

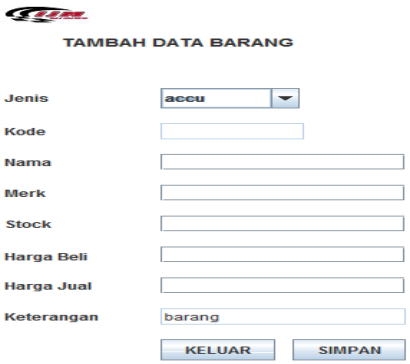
Halaman ini berfungsi untuk mengubah data-data lama *supplier* .



Gambar 4.25. Tampilan Halaman Edit Data *Supplier*

e. Halaman Tambah Data Barang

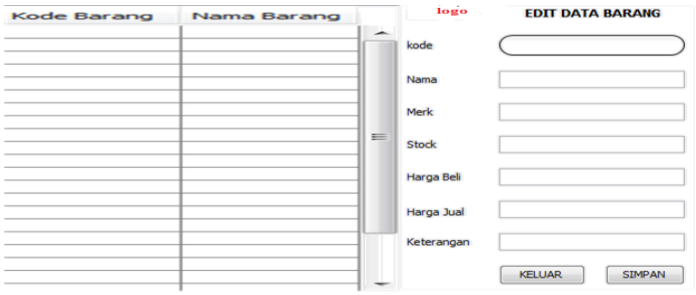
Halaman ini berfungsi untuk menambahkan data barang baru yang dilakukan oleh bagian gudang dengan penyimpanan otomatis.



Gambar 4.26. Tampilan Halaman Tambah Data Barang

f. Halaman Edit Barang

Halaman ini berfungsi untuk mengubah data-data lama barang yang ada di *database*, dengan pencarian dan penyimpanan otomatis.



Gambar 4.27. Tampilan Halaman Edit Barang

4. Perancangan *Input Owner*

a. Halaman Tambah Jasa

Tambah Jasa ini berfungsi untuk menambahkan jasa *service* yang baru pada bengkel oleh *owner*.



Gambar 4.28 Tampilan Halaman Tambah Jasa

b. Halaman Edit Jasa

Edit jasa ini berfungsi untuk mengubah jasa yang lama, dengan pencarian, pengubahan dan penyimpanan yang otomatis.



Gambar 4.29. Tampilan Halaman Edit Jasa

c. Halaman Data Karyawan

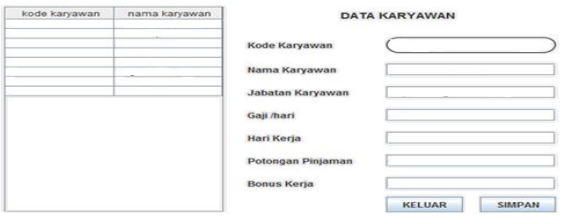
Menambahkan Karyawan Baru di lakukan oleh *owner* dengan menginputkan data-data karyawan.



Gambar 4.30. Tampilan Halaman data karyawan

d. Halaman Edit Data Karyawan

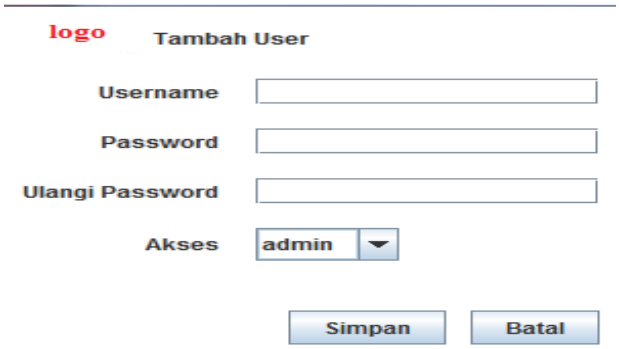
edit karyawan ini berfungsi untuk mengubah karyawan yang lama, dengan pencarian, pengubahan dan penyimpanan yang otomatis.



Gambar 4.31. Tampilan Halaman edit Data Karyawan

e. Halaman Tambah *User*

Menambahkan data *User* dengan menginputkan Data



Gambar 4.32. Tampilan Halaman tambah user

f. Halaman Edit *User*

Mengubah data *user* seperti mengubah *password* , dan *username* atau pun data hak akses juga.



Gambar 4.33. Tampilan Halaman Edit Jasa

g. Halaman *Input* Laporan

Owner dapat menampilkan laporan dengan *periode* yang dapat diatur.



Gambar 4.34. Tampilan laporan

4.2.4.2 Perancangan *Output*

Perancangan *output* merupakan bentuk tampilan keluaran berupa laporan-laporan. Adapaun perancangan *output* pada sistem informasi pelayanan pada bengkel indra jaya motor adalah sebagai berikut :

1. **Perancangan *Output* Kasir**

   a. **Halaman *Output* Transaksi/Cetak *Struk***

   Tampilan *Output* setelah Kasir melakukan Transaksi.



Gambar 4.35. Tampilan *Output* Transaksi/Cetak *Struk*

2. **Perancangan *Output* Gudang**

   a. **Tampilan *Output Stock* minimum Barang**

   Tampilan *output* ini akan tampil setiap kali barang akan habis atau menemui *stock* minimum.



Gambar 4.36. Tampilan *Output Stock* Minimum Barang

   b. **Tampilan O*utput* Buat *Purchase Order***

   Tampilan *output* ini akan tampil setiap kali barang akan habis atau menemui *stock* minimum.

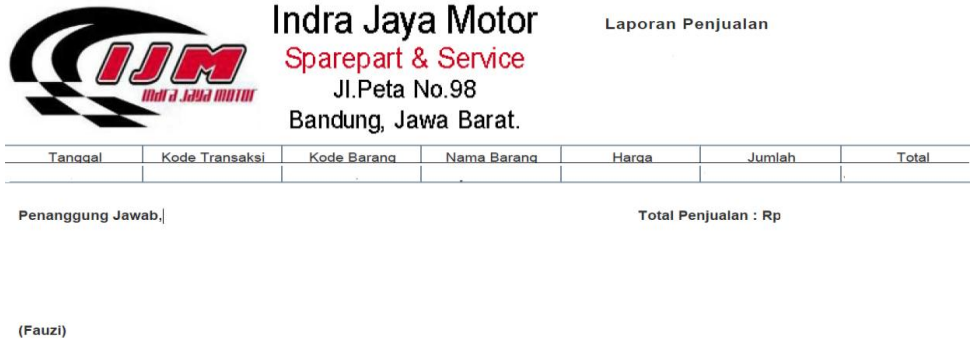


Gambar 4.37. *Tampilan Buat PurchaseOrder*

3. Perancangan *Output Owner*

a. Tampilan *Output* Laporan Penjualan

Tampilan Laporan Penjualan atau menjual suku cadang dan jasa *service*.



Gambar 4.38. Tampilan Laporan Penjualan

b. Tampilan *Output* Laporan Pembelian

Tampilan Laporan Pembelian atau membeli suku cadang



Gambar 4.39. Tampilan Laporan Pembelian

c. Tampilan *Output* Laporan *stock* minimum

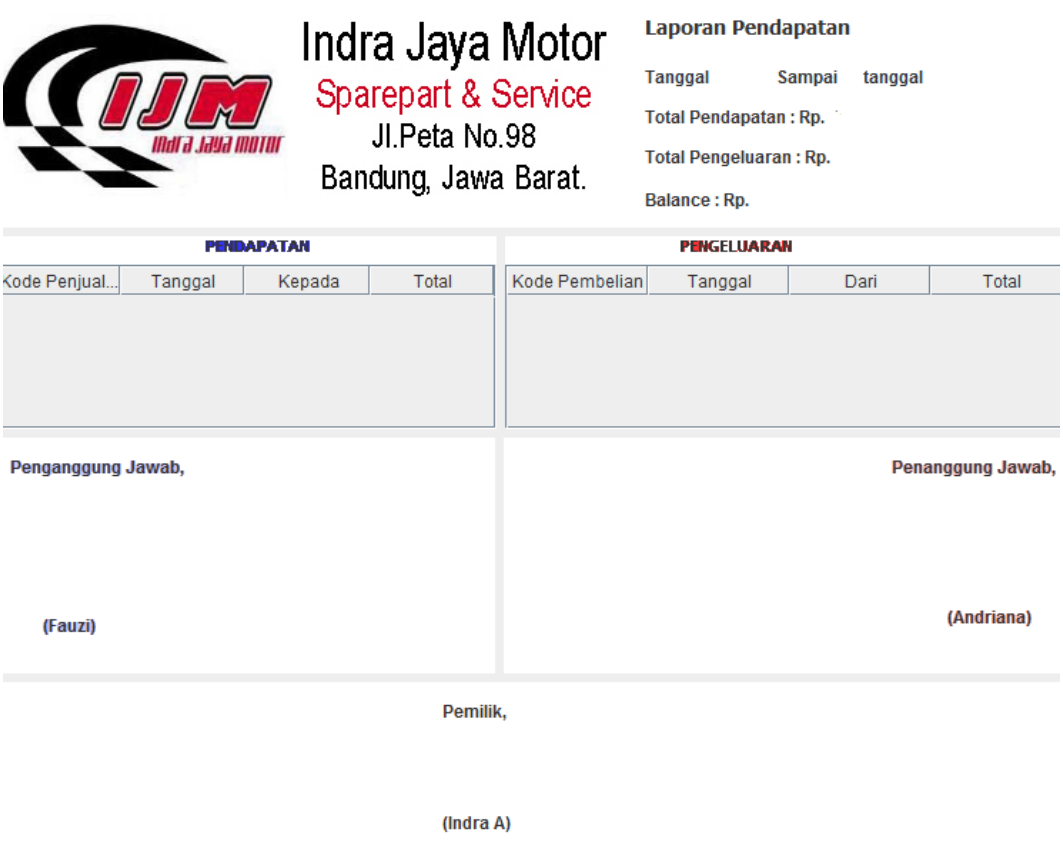
Untuk menampilkan laporan stock minimum dalam perusahaan.



Gambar 4.40. Tampilan Laporan *stock* minimum

d. Tampilan Output Pendapatan

Untuk menampilkan laporan pendapatan dan pengeluaran di perusaahaan Indra Jaya Motor.



Gambar 4.41. Tampilan Laporan Pembelian

4.2.5 Perancangan Arsitektur Jaringan

Adapun Kebutuhan Jaringan untuk mendukung program ini adalah sebagai berikut :

1. Tipe jaringan yang digunakan adalah LAN (*Local Area Network)* digunakan untuk menghubungkan komputer yang berbeda didalam suatu area yang kecil, misalnya didalam suatu gedung perkantoran atau kampus.

2. Media Penyambungnya menggunakan Kabel UTP dan *Switch* sebagai *multiport bridge. Switch* berfungsi sebagai sentral atau konsentrator pada sebuah *network.*