BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

Kegiatan implementasi atau penerapan dilakukan dengan dasar yang telah direncanakan dalam rencana implementasi. Pada penerapan sistem yang diusulkan tentu saja harus disediakan beberapa aspek yang akan mendukung kelancaran operasi sistem dan hal yang akan dijadikan faktor pendukung adalah penentuan kebutuhan sumber daya.

Implementasi adalah proses penerapan rancangan program yang telah dibuat pada bab sebelumnya atau aplikasi dalam melaksanakan sistem informasi pemrograman yang dibuat.

5.1. Implementasi

Tahap selanjutnya dalam membangun sebuah perangkat lunak adalah implementasi. Implementasi merupakan kumpulan dari elemen-elemen yang telah didesain ke dalam bentuk pemrograman untuk menghasilkan suatu tujuan yang dibuat berdasarkan kebutuhan yang telah dianalisis sebelumnya. Langkah-langkah dalam tahap implementasi ini adalah urutan kegiatan awal sampai akhir yang harus dilakukan untuk mewujudkan sistem yang telah dirancang.

5.1.1. Batasan Implementasi *(optional)*

Dalam mengimplementasikan perangkat lunak pemantauan dan pengendalian aplikasi ini ada beberapa hal yang menjadi batasan implementasi, yaitu :

1. Tidak semua kebutuhan sistem informasi ini dikerjakan, tetapi hanya subsistem-subsistem yang mendukung khususnya proses pada transaksi penjualan.
2. *Administrator* pada saat implementasi dikerjakan oleh Bagian *Stock.*

5.1.2. Implementasi Perangkat Lunak

Perangkat lunak yang dibutuhkan dan digunakan dalam pembuatan aplikasi sistem informasi penjualan ini adalah sebagai berikut :

1. *Windows Seven Ultimate*, sebagai sistem operasi yang dipakai oleh penulis.
2. *NetBeans IDE 7.3.*sebagai *software* yang penulis gunakan dalam membangun sistem informasi penjualan ini.
3. *iReport*, sebagai aplikasi untuk pembuatan laporan.
4. *MySQL*, sebagai pengembang basis data dan perangkat lunak lainnya menggunakan *XAMPP*.

5.1.3. Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan berdasarkan kebutuhan minimal yang harus terpenuhi antara lain :

1. *Server*
   a. Menggunakan minimal prosessor berkecepatan 2.50 *Ghz*
   b. Menggunakan RAM minimal 512 MB.

c. Tersedianya *HardDrive* untuk media penyimpanan, minimal 10 MB untuk *server*, diluar basisdata.

d. *Mouse*, *Keyboard*, dan *Monitor* sebagai peralatan antarmuka.

2. *Client*

a. Menggunakan minimal prosessor berkecepatan 2.50 *Ghz*

b. Menggunakan RAM minimal 256 MB

c. *Mouse*, *Keyboard*, dan *Monitor*, dan *Printer* sebagai peralatan antarmuka.

5.1.4. Implementasi Basis Data (Sintaks *SQL*)

Pembuatan basisdata dilakukan dengan menggunakan bahasa *SQL*, dimana *DBMS* yang digunakan adalah *MySQL*. Implementasi basisdatanya dalam bahasa *SQL* adalah sebagai berikut :

```
-- phpMyAdmin SQL Dump

-- version 3.5.2.2

-- http://www.phpmyadmin.net

--

-- Host: 127.0.0.1

-- Generation Time: Jul 08, 2013 at 06:26 AM

-- Server version: 5.5.27
```

```
-- PHP Version: 5.4.7

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";


-- Database: `hali`
--


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


--

-- Table structure for table `data_barang`

--

CREATE TABLE IF NOT EXISTS `data_barang` (

  `number` varchar(30) COLLATE latin1_general_ci NOT NULL,

  `kode_brg` varchar(30) COLLATE latin1_general_ci NOT NULL,

  `jenis_brg` varchar(30) COLLATE latin1_general_ci NOT NULL,

  `nama_brg` varchar(100) COLLATE latin1_general_ci NOT NULL,

 `size` varchar(30) COLLATE latin1_general_ci NOT NULL,

  `harga_brg` int(30) NOT NULL,
   `qty` int(30) NOT NULL
)
 ENGINE=MyISAM DEFAULT CHARSET=latin1;
 COLLATE=latin1_general_ci;


--

-- Dumping data for table `data_barang`

--
```

```
INSERT INTO `data_barang` (`number`, `kode_brg`, `jenis_brg`, `nama_brg`, `size`, `harga_brg`, `qty`) VALUES

('1', 'SRfcc', 'Sarung', 'Sarung FM Cap Cen', 'ASZ', 50000, 286),

('2', 'SRaa', 'Sarung', 'Sarung Al - Aqso', 'ASZ', 60000, 215),

('3', 'SRsd', 'Sarung', 'Sarung Sapphire Donggala', 'ASZ', 70000, 200),

('4', 'SRap', 'Sarung', 'Sarung Atlas Premium', 'ASZ', 75000, 180),

('5', 'Srgde', 'Sarung', 'Sarung Gajah Duduk Exclusive', 'ASZ', 80000, 200),

('6', 'SRms', 'Sarung', 'Sarung Mangga Songket', 'ASZ', 95000, 200),

('7', 'SRsmrnd', 'Sarung', 'Sarung Samarinda ', 'ASZ', 350000, 200),

('8', 'SRbhs', 'Sarung', 'Sarung BHS Kembang', 'ASZ', 450000, 200),

('9', 'BKak', 'BajuKoko', 'Baju Koko Akmal', 'S', 50000, 200),

('10', 'BKap', 'BajuKoko', 'Baju Koko Annas Panjang', 'M', 70000, 200),


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


-
-

-- Table structure for table `detail_nota_pembelian`

--


CREATE TABLE IF NOT EXISTS `detail_nota_pembelian`
(

  `no_nota` varchar(10) NOT NULL,

`tanggal` date NOT NULL,

`jenis_brg` varchar(10) NOT NULL,

`size` varchar(30) NOT NULL,
```

```
`kode_brg` varchar(20) NOT NULL,

  `nama_brg` varchar(100) NOT NULL,

`nama_suplier` varchar(20) NOT NULL,

`banyak_brg` int(5) NOT NULL,

`hrga_satuan` int(10) NOT NULL,

 `harga_brg` int(10) NOT NULL,

`total` int(10) NOT NULL,

`harga_jual` int(10) NOT NULL
)
 ENGINE=MyISAM DEFAULT CHARSET=latin1;


--

-- Dumping data for table `detail_nota_pembelian`

--


INSERT INTO `detail_nota_pembelian`
(`no_nota`, `tanggal`, `jenis_brg`, `size`, `kode_brg`, `nama_brg`, `nama_suplier`,
`banyak_brg`, `hrga_satuan`, `harga_brg`, `total`, `harga_jual`) VALUES



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


-
-

-- Table structure for table `detail_pembelian_barang`

--

CREATE TABLE IF NOT EXISTS `detail_pembelian_barang`
(
```

```
  `no_pembelian` varchar(10) NOT NULL,

 `tgl_beli` date NOT NULL,

 `kode_barang` varchar(10) NOT NULL,

 `jenis_barang` varchar(15) NOT NULL,

  `nama_barang` varchar(100) NOT NULL,

  `size` varchar(30) NOT NULL,

`qty` int(15) NOT NULL,

 `nama_suplier` varchar(15) NOT NULL,

  `kode_suplier` varchar(5) NOT NULL
)
ENGINE=MyISAM DEFAULT CHARSET=latin1;


--

-- Dumping data for table `detail_pembelian_barang`

--


INSERT INTO `detail_pembelian_barang`
(`no_pembelian`, `tgl_beli`, `kode_barang`, `jenis_barang`, `nama_barang`, `size`,
`qty`, `nama_suplier`, `kode_suplier`) VALUES



-- Table structure for table `detail_return`

--


CREATE TABLE IF NOT EXISTS `detail_return` (

 `number` varchar(30) CHARACTER SET latin1 NOT NULL,

`no_nota` varchar(30) CHARACTER SET latin1 NOT NULL,

`tgl_transaksi` date NOT NULL,
```

```
  `jenis_brg` varchar(30) CHARACTER SET latin1 NOT NULL,

  `nama_brg` varchar(100) CHARACTER SET latin1 NOT NULL,

 `size` varchar(30) CHARACTER SET latin1 NOT NULL,

`harga` int(30) NOT NULL,
 `jumlah` int(30) NOT NULL,
 `subtotal` int(30) NOT NULL,
 `bayar` int(30) NOT NULL,
 `uang_kembalian` int(30) NOT NULL,
 `tgl_return` date NOT NULL,
  `jml_return` int(30) NOT NULL)
 ENGINE=MyISAMDEFAULTCHARSET=latin1 COLLATE=latin1_general_ci;


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


-- Table structure for table `detail_transaksi`

--

CREATE TABLE IF NOT EXISTS `detail_transaksi` (

  `no_nota` varchar(15) NOT NULL,

 `tgl_transaksi` date NOT NULL,

  `jenis_brg` varchar(20) NOT NULL,

  `nama_brg` varchar(100) NOT NULL,

  `size` varchar(30) NOT NULL,

 `harga_satuan` int(20) NOT NULL,

  `jumlah` int(20) NOT NULL,

 `subtotal` int(10) NOT NULL,

  `bayar` int(20) NOT NULL,

`uang_kembalian` int(11) NOT NULL)
 ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--

-- Dumping data for table `detail_transaksi`

--


INSERT INTO `detail_transaksi` (
`no_nota`, `tgl_transaksi`, `jenis_brg`, `nama_brg`, `size`, `harga_satuan`, `jumlah`, `subtotal`, `bayar`, `uang_kembalian`) VALUES


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

-- Table structure for table `nota_pembelian`

--


CREATE TABLE IF NOT EXISTS `nota_pembelian` (

  `no_nota` varchar(10) NOT NULL,
  `tanggal` date NOT NULL,
 `jenis_brg` varchar(10) NOT NULL,
 `nama_suplier` varchar(20) NOT NULL,
 `total` int(10) NOT NULL)
 ENGINE=MyISAM DEFAULT CHARSET=latin1;

--

-- Dumping data for table `nota_pembelian`

--


INSERT INTO `nota_pembelian` (
`no_nota`, `tanggal`, `jenis_brg`, `nama_suplier`, `total`) VALUES


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

-
```

```
-- Table structure for table `obral`

--

CREATE TABLE IF NOT EXISTS `obral` (

`jenis_brg` varchar(30) NOT NULL,

`nama_brg` varchar(100) NOT NULL,

`size` varchar(5) NOT NULL,

 `jumlah` int(30) NOT NULL,

 `harga` int(30) NOT NULL,

 `total` float NOT NULL,

 `bayar` float NOT NULL)
 ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

-- Dumping data for table `obral`

--

INSERT INTO `obral` (`jenis_brg`, `nama_brg`, `size`, `jumlah`, `harga`, `total`,
`bayar`) VALUES


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

-- Table structure for table `pembelian_barang`

--

CREATE TABLE IF NOT EXISTS `pembelian_barang` (

  `no_pembelian` varchar(10) NOT NULL,

 `tgl_beli` date NOT NULL,
```

```
 `kode_barang` varchar(10) NOT NULL,

 `jenis_barang` varchar(10) NOT NULL,

 `nama_barang` varchar(100) NOT NULL,

 `size` varchar(30) NOT NULL,

 `qty` int(10) NOT NULL)
ENGINE=MyISAM DEFAULT CHARSET=latin1;

--

-- Dumping data for table `pembelian_barang`

--

INSERT INTO `pembelian_barang` (`no_pembelian`, `tgl_beli`, `kode_barang`,
`jenis_barang`, `nama_barang`, `size`, `qty`) VALUES

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

-- Table structure for table `return1`

--


CREATE TABLE IF NOT EXISTS `return1` (

 `number` varchar(30) NOT NULL,

 `no_nota` varchar(30) NOT NULL,

 `tgl_return` date NOT NULL)
ENGINE=MyISAM DEFAULT CHARSET=latin1;


-- --------------------------------------------------------
-
- Table structure for table `return_cacat`

--

CREATE TABLE IF NOT EXISTS `return_cacat` (

  `no_nota` int(15) NOT NULL,
```

```
 `tgl_transaksi` date NOT NULL,

  `jenis_brg` varchar(30) NOT NULL,

  `nama_brg` varchar(100) CHARACTER SET latin1 COLLATE
   latin1_general_ci NOT NULL,

  `jumlah` int(30) NOT NULL,

 `size` varchar(30) NOT NULL)
 ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `return_cacat`
--

INSERT INTO `return_cacat` (`no_nota`, `tgl_transaksi`, `jenis_brg`, `nama_brg`,
`jumlah`, `size`) VALUES

-- --------------------------------------------------------
-- Table structure for table `return_ukuran`
--


CREATE TABLE IF NOT EXISTS `return_ukuran` (

  `no_nota` int(15) NOT NULL,

 `tgl_transaksi` date NOT NULL,

 `jenis_brg` varchar(30) NOT NULL,

  `nama_brg` varchar(100) NOT NULL,

  `jumlah` int(20) NOT NULL,

 `size` varchar(30) NOT NULL)
 ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Dumping data for table `return_ukuran`
--

INSERT INTO `return_ukuran` (`no_nota`, `tgl_transaksi`, `jenis_brg`,
`nama_brg`, `jumlah`, `size`) VALUES

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

```
--

-- Table structure for table `supplier`

--


CREATE TABLE IF NOT EXISTS `supplier` (

  `kode_suplier` varchar(10) NOT NULL,

 `nama_suplier` varchar(20) NOT NULL
)
ENGINE=MyISAM DEFAULT CHARSET=latin1;


--

-- Dumping data for table `supplier`

--


INSERT INTO `supplier` (`kode_suplier`, `nama_suplier`) VALUES


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


--

-- Table structure for table `transaksi`

--


CREATE TABLE IF NOT EXISTS `transaksi` (

  `no_nota` varchar(15) NOT NULL,

  `tgl_transaksi` date NOT NULL,

 `jenis_brg` varchar(10) NOT NULL,
```

```
 `nama_brg` varchar(100) NOT NULL,

 `size` varchar(30) NOT NULL,

`harga_satuan` int(15) NOT NULL,

 `jumlah` int(11) NOT NULL,

`subtotal` int(15) NOT NULL)
 ENGINE=MyISAM DEFAULT CHARSET=latin1;


--

-- Dumping data for table `transaksi`

--

INSERT INTO `transaksi` (`no_nota`, `tgl_transaksi`, `jenis_brg`, `nama_brg`, `size`, `harga_satuan`, `jumlah`, `subtotal`) VALUES


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

5.1.5. Implementasi Antar Muka

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program sistem informasi penjualan pada Adis Sportindo. Cara penggunaanya adalah sebagai berikut :

5.1.5.1. Implementasi Form Menu Utama

Implementasi Form yang terdapat ketik data pengguna menginputkan Form login dapat dilihat pada tabel 5.1 berikut :

Tabel 5.1. Implementasi Sub Menu Form Utama

Sub Menu	Deskripsi	Nama File
File	Menu ini untuk menampilkan *Login, Logout* dan E*xit* pada saat user akan menggunakan aplikasi ini.	Menu
Data	Menu ini untuk menampilkan Input *Stock* Barang, Stock Barang, Input Data Pembelian dan Return.	Menu
Transaksi	Menu ini menginputkan Transaksi Penjualan langsung , Pembelian barang dan Obral	Menu
Laporan	Menu ini menampilkan Laporan Data Stock Barang, Laporan Data Penjualan, dan Laporan Data Pembelian.	Menu

5.1.5.2. Implementasi Form Input Stock Barang

Implementasi form stock barang digunakan untuk menginputkan data barang baru dan menampilkan barang yang akan habis dapat dilihat pada tabel 5.2 berikut:

Tabel 5.2 Implementasi Form Input Stock Barang

Sub Menu	Deskripsi	Nama File
TAMBAH	Tombol ini berfungsi untuk menginput data barang baru agar field bisa terisi	Data Stock Barang
SIMPAN	Tombol ini berfungsi untuk menyimpan data barang ke database	Data Stock Barang
EDIT	Tombol ini berfungsi untuk mengedit nama barang, apabila terjadi kesalahan pada penginputan *ART* (Nama Barang).	Data Stock Barang
CARI	Tombol ini berfungsi untuk melakukan pencarian barang, bisa berdasarkan Nama Barang dan kode barang.	Data Stock Barang
CEK STOCK	Tombol ini berfungsi untuk mengetahui barang yang akan habis dengan *quantity* kurang dari 20.	Data Stock Barang
BATAL	Tombol ini berfungsi untuk membatalkan mengedit nama barang barang , jenis barang dan kode barang	Data Stock Barang
HAPUS	Tombol ini berfungsi untuk menghapus data barang	Data Stock Barang

REFRESH	Tombol ini berfungsi untuk melihat quantity apabila sudah terjadi transaksi penjulalan	Data Stock Barang

5.1.5.3. Implementasi Form Stock Barang

Implementasi form Data Jenis Barang digunakan untuk mengecek persediaan barang dapat dilihat pada tabel 5.3 berikut :

Tabel 5.3 Implementasi Form Stock Barang

Sub Menu	Deskripsi	Nama File
CARI	Tombol ini berfungsi untuk melakukan pencarian barang berdasarkan nama, ukuran, dan kode barang.	Data jenis barang
REFRESH	Tombol ini berfungsi untuk melihat quantity apabila sudah terjadi transaksi penjulalan	Data jenis barang

5.1.5.4. Implementasi Form Input Data Pembelian

Implementasi form input nota pembelian digunakan untuk menginput data pembelian dan menambah barang yang baru dibeli ke Stock Barang dapat dilihat pada Tabel 5.4 berikut :

Tabel 5.4 Implementasi Form Input Data Pembelian

Sub Menu	Deskripsi	Nama File
TAMBAH	Tombol ini untuk menampilkan data barang baru ke list pembelian.	Detail Nota Pembelian
UPDATE	Tombol ini berfungsi untuk menambah/meng-*update* barang dan secara otomatis barang di *Stock* akan bertambah	Detail Nota Pembelian
HAPUS	Tombol ini untuk menghapus data di list pembelian	Detail Nota Pembelian
HITUNG	Penghitungan subtotal dari list pembelian barang	Detail Nota Pembelian
RESET	Tombol ini untuk mengosongkan semua *field*	Detail Nota Pembelian

5.1.5.5. Implementasi Form Transaksi Penjualan

Implementasi form Transaksi Penjualan digunakan untuk melakukan transaksi langsung dan mencetak nota untuk Konsumen yang dapat dilihat pada tabel 5.5 berikut ini :

Tabel 5.5 Implementasi Form Transaksi Penjualan

Sub Menu	Deskripsi	Nama File
TAMBAH KE DAFTAR	Tombol ini untuk menambahkan barang yang akan dibeli ke daftar pembelian.	transaksi
TAMBAH	Tombol ini untuk menambahkan jumlah total bayar	transaksi
RESET	Tombol ini untuk mengosongkan semua *field*	transaksi
HAPUS	Tombol untuk menghapus barang yang akan dibeli di daftar pembelian.	transaksi
SIMPAN	Tombol ini untuk berfungsi menyimpan data penjualan (untuk laporan) ke *database* dan mencetak nota penjualan untuk konsumen.	transaksi
HITUNG	Penghitungan subtotal barang yang di beli oleh konsumen	transaksi

5.1.5.6. Implementasi Form Beli Barang

Implementasi form beli barang digunakan untuk menginput data barang yang akan dibeli dan mecetak data beli barang untuk diberikan ke *supplier* dapat dilihat pada Tabel 5.6 berikut :

Tabel 5.6 Implementasi Form Beli Barang

Sub Menu	Deskripsi	Nama File
TAMBAH	Tombol ini berfungsi untuk menambah barang yang di beli ke daftar.	Data Pembelian Barang
SIMPAN	Tombol ini berfungsi untuk menyimpan data beli barang agar bisa di cetak	Data Pembelian Barang
CETAK	Tombol ini untuk mencetak data beli barang untuk *Supplier*	Data Pembelian Barang
RESET	Tombol ini untuk mengosongkan semua *field*	Data Pembelian Barang
HAPUS	Tombol untuk menghapus barang yang akan dibeli di daftar pembelian.	Data Pembelian Barang
REFRESH	Tombol ini berfungsi untuk melihat quantity apabila sudah terjadi transaksi penjulalan	Data Pembelian Barang

CEK	Tombol ini berfngsi untuk mengecej stock barang yang hampir habis	Data Pembelian Barang

5.1.5.7. Implementasi Form Obral

Implementasi form obral digunakan untuk mengobral barang retuen yang cacat yang kemudian dijual dengan harga diskon dapat dilihat pada Tabel 5.7 berikut :

Tabel 5.7 Implementasi Form obral

Sub Menu	Deskripsi	Nama File
TAMBAH	Tombol ini berfungsi untuk menambah barang yang di beli ke daftar.	Obral Barang
SIMPAN	Tombol ini berfungsi untuk menyimpan data beli barang agar bisa di cetak	Obral Barang
HITUNG	Penghitungan subtotal barang yang di beli oleh konsumen	Obral Barang
RESET	Tombol ini untuk mengosongkan semua *field*	Obral Barang
HAPUS	Tombol untuk menghapus barang yang akan dibeli di daftar pembelian.	Obral Barang

5.1.5.8. Implementasi Form Laporan Data Stock Barang

Implementasi form laporan data stock barang digunakan untuk menampilkan laporan data stock barang dapat dilihat pada Tabel 5.8 berikut

Tabel 5.8 Implementasi Form Laporan Data Stock Barang

Sub Menu	Deskripsi	Nama File
PREVIEW	Menu ini digunakan untuk menampilkan dan mencetak laporan data persediaan	Laporan data stock barang

5.1.5.9. Implementasi Form Laporan Data Penjualan

Implementasi form laporan penjualan digunakan untuk menampilkan data laporan penjualan dapat dilihat pada Tabel 5.9 berikut

Tabel 5.9 Implementasi Form Laporan Data Penjualan

Sub Menu	Deskripsi	Nama File
PREVIEW	Tombol ini digunakan untuk menampilkan dan mencetak laporan penjualan berdasarkan tanggal,bulan, dan tahun	Laporan Penjualan

5.1.5.10. Implementasi Form Laporan Data Pembelian

Implementasi form laporan pembelian digunakan untuk menampilkan data penjualan dapat dilihat pada Tabel 5.10 berikut :

Tabel 5.10 Implementasi Form Laporan Data Pembelian

Sub Menu	Deskripsi	Nama File
PREVIEW	tombol ini digunakan untuk melihat dan mencetak laporan data pembelian berdasarkan tanggal,bulan, dan tahun.	Laporan Pembelian

5.1.6. Implementasi Instalasi Program

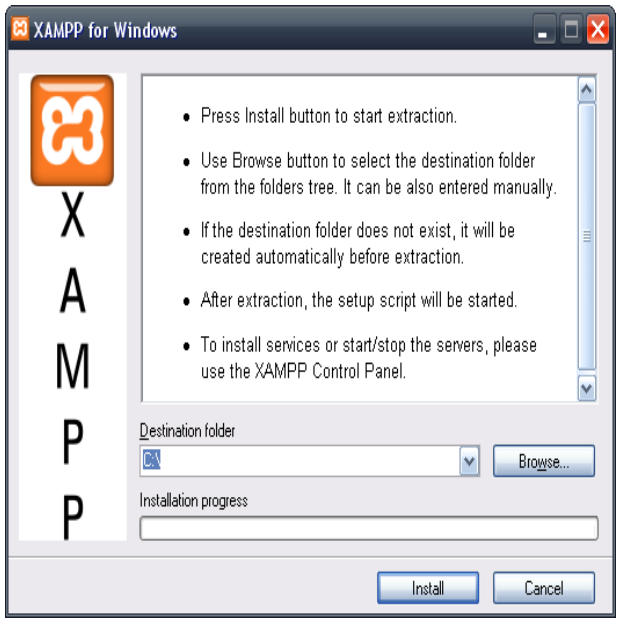
5.1.6.1 Instalasi *XAMPP*

Pertama yang harus dilakukan adalah membuka file instalasi *xampp win32-1.6.6a* yang berekstensi.*exe* lalu *double click* untuk menginstalnya.

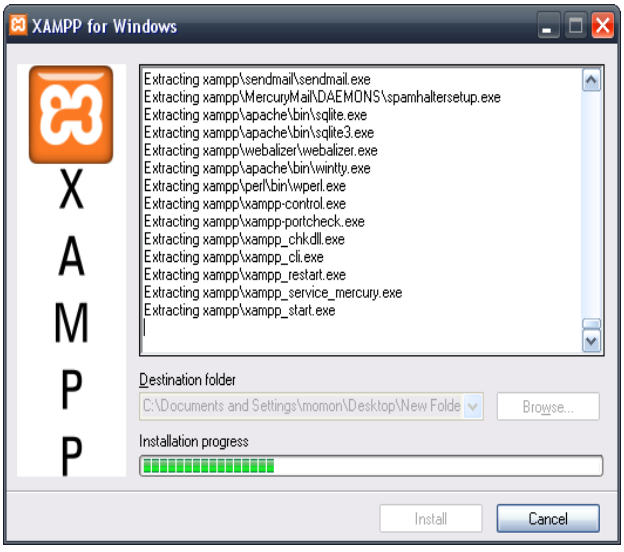


Gambar 5.1 Icon Instalasi *xampp-win32-1.6.6a*

Lalu pilih Instal.

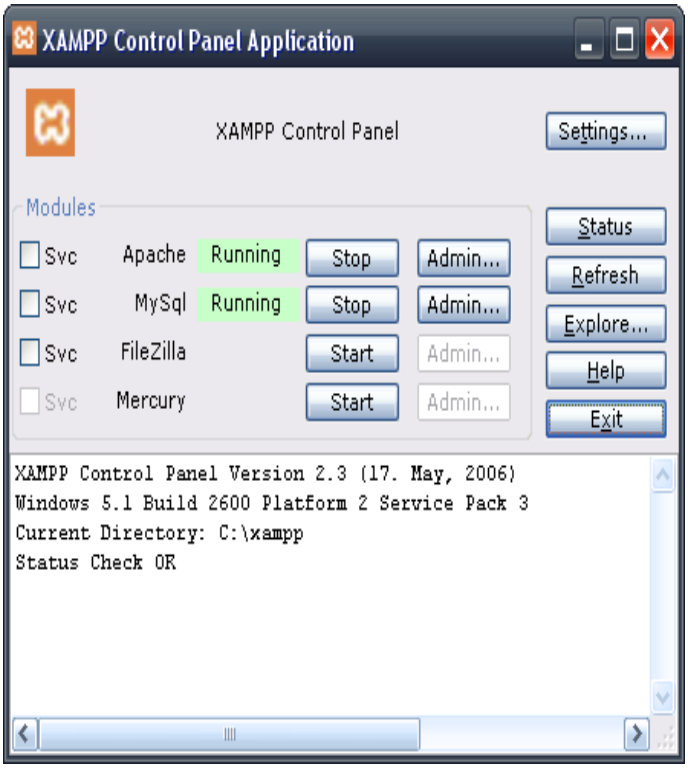


Gambar 5.2 Tampilan Awal Instalasi *xampp*



Gambar 5.3 Tampilan Proses Instalasi *xampp*

Setelah muncul di *desktop*, *double click xampp-control* untuk menjalankannya. Maka akan muncul tampilan seperti pada gambar 5.4



Gambar 5.4 Tampilan xampp control panel

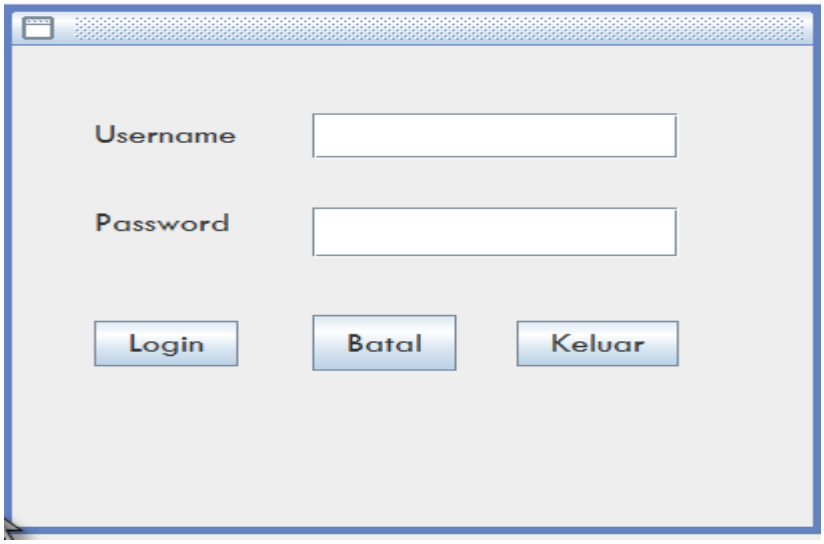
Jalankan *Apache* dan *MySql* nya dengan menekan tombol *start* di sisi kanannya sehingga pada *Apache* dan *MySql* muncul progressbar "*Running*".

5.1.7. Penggunaan Program

Untuk menggunakan program dari perancangan Sistem Informasi Penjualan pada PD .H. ALI dilakukan dengan penginstalan aplikasi Sistem Informasi penjualan terlebih dahulu. Aplikasi ini berfungsi untuk membantu dalam pengolahan data *stock* barang, data pembelian barang, dan transaksi penjualan serta pembuatan laporan penjualan, laporan pembelian dan laporan data stock barang.

a. Form Login

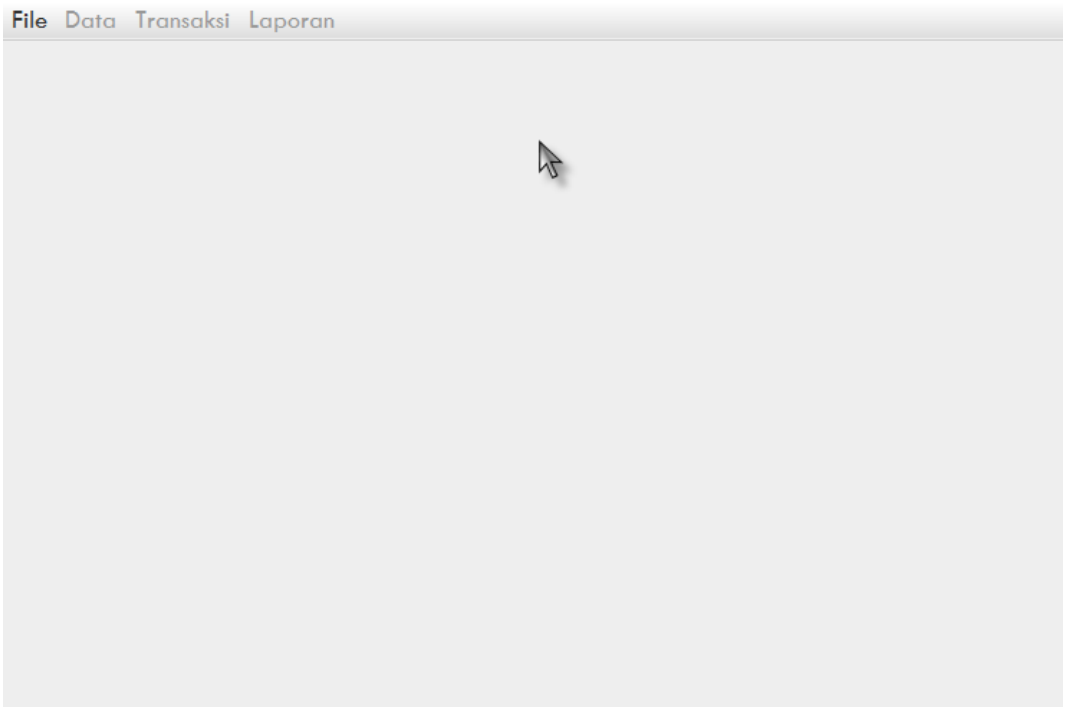
User harus menginputkan *username* dan *passwordnya* untuk bisa mengakses menu lain,



Gambar 5.5 Form login

b. Form Menu Utama

Tampilan awal dari aplikasi ini akan menghubungkan *user* dengan menu selanjutnya yang telah dipilih.



Gambar 5.6 Form Menu Utama

c. Form Stock Barang

Apabila *user* meng-*click* menu lalu pilih Stock Barang, maka akan muncul tampilan sebagai berikut :

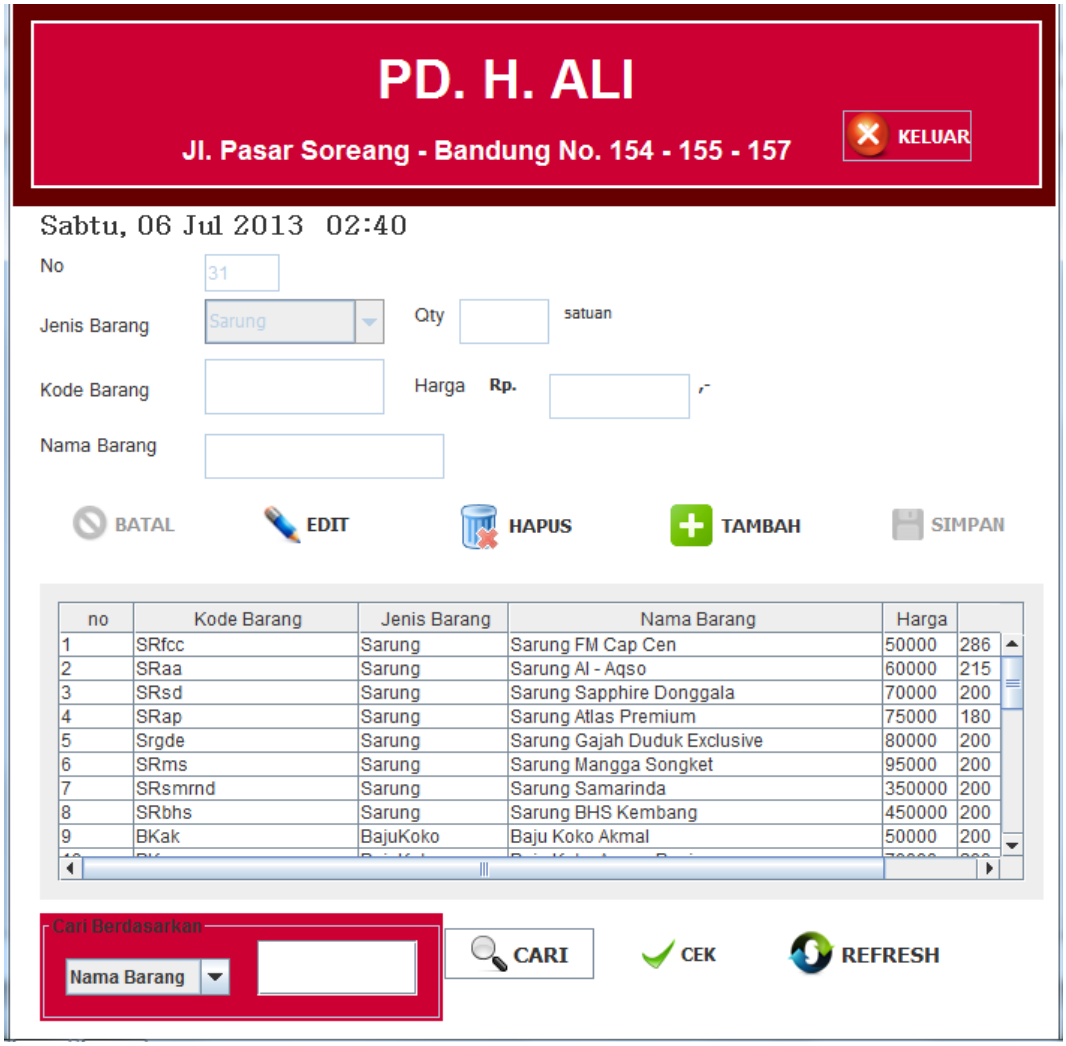


Gambar 5.7 Form Stock Barang

Deengan adanya form stock barang untuk memudahkan Bag.Kasir untuk melakukan pencarian barang.

d. Form Input Stock Barang

Apabila user membuka *menu* Data lalu Input Stock Barang maka akan muncul tampilan sebagai berikut :



Gambar 5.8 Form Input Stock Barang

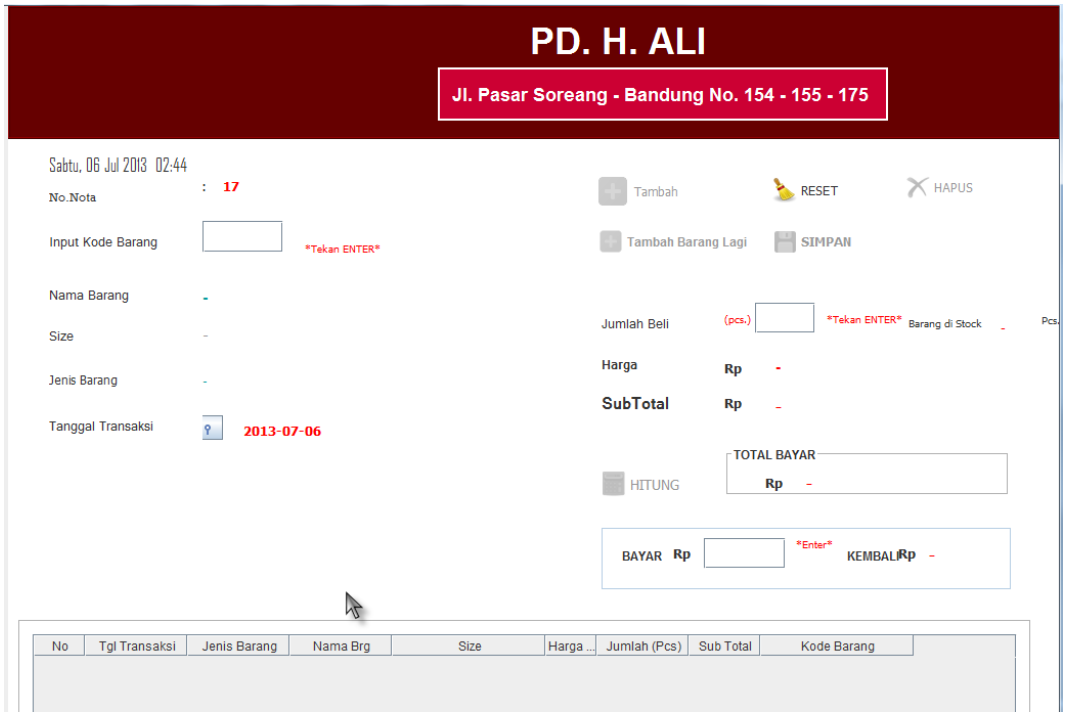
Klik tombol ‘TAMBAH’ untuk menginputkan data barang yang baru kemudian isi sesuai dengan ketenuan, setelah itu klik tombol ‘SIMPAN’ maka data barang tersebut tersimpan ke tabel. Untuk mengubah nama dari data barang klik tombol ‘EDIT’. Untuk kembali ke halaman menu utama klik tombol ‘KELUAR’. Untuk melakukan pencarian barang hanya tinggal menginputkan nama barang, secara otomatis barang akan muncul pada tabel

untuk pencarian berdasarkan ukuran dan kode barang, isi *field*-nya kemudian klik tombol ‘CARI’.

Untuk mengetahui barang yang akan habis (di bawah Qty 20) klik tombol ‘CEK STOCK’, barang akan secara otomatis muncul di tabel.

e. Form Transaksi Penjualan

Apabila Bag.Kasir membuka menu transaksi lalu penjualan maka akan muncul sebuah tampilan, lalu isi semua *field* sesuai permintaan dari konsumen mengenai barang yang akan dibeli. Tampilannya sebagai berikut :

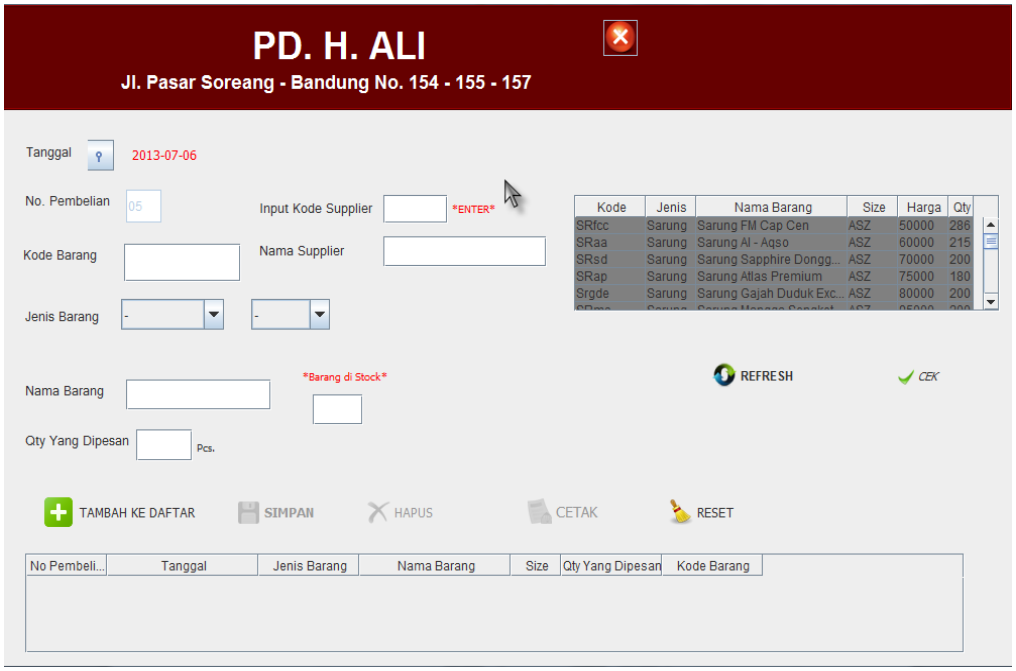


Gambar 5.9 Form Transaksi Penjualan

Setelah ‘Transaksi Berhasil’ lalu klik tombol ‘SIMPAN’ dan secara otomatis nota pembelian mun tercetak, berikut tampilan nota pembelian:

f. Form Pembelian Barang

Apabila Bag.Stock membuka menu transaksi lalu pembelian maka akan muncul tampilan seperti berikut :

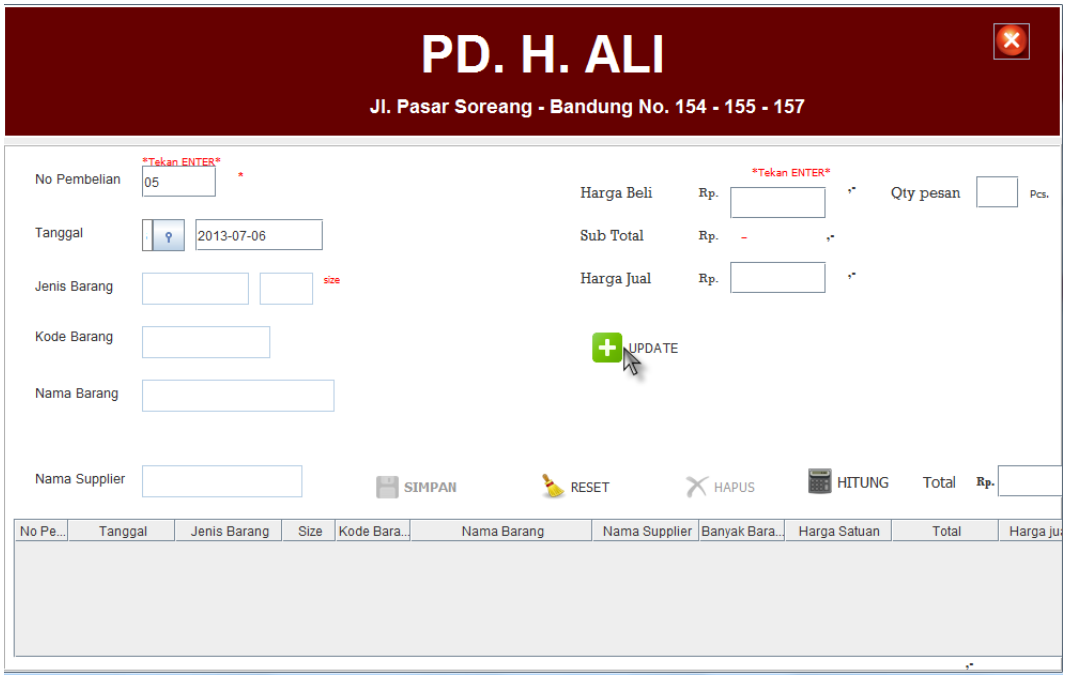


Gambar 5.10 Form Pembelian Barang

“Tambah kedaftar” klik lalu barang yang akan dibeli secara otomatis data barang akan terisi pada *field*, lalu inputkan “JUMLAH” jumlah barang yang akan dibeli, kemudian inputkan kode *supplier* tekan *Enter*. Lalu klik tombol ‘TAMBAH DATA BARANG’ lalu ‘SIMPAN’ dan klik tombol ‘CETAK DAFTAR PEMBELIAN’ untuk mencetak data beli barang untuk diberikan kepada *Supplier*.

g. Form Input Data Pembelian

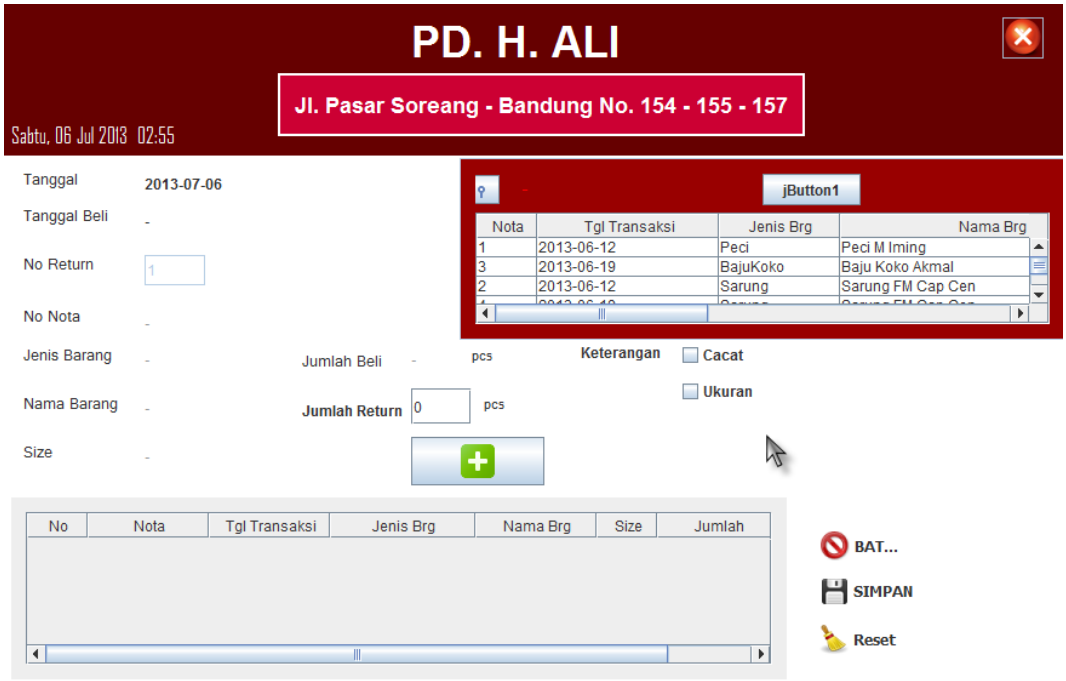
Pada tampilan utama, apabila Bag.*Stock* mengklik menu ‘Data’ lalu ‘Input Data Pembelian’ maka akan muncul tampilan dimana tampilan ini harus melakukan pemanggilan data dari no.pembelian agar barang bisa langsung ditambahkan ke *stock* barang, Ketika data no.pembelian tersebut ada maka secara otomatis data barang akan muncul pada *list*/tabel berikut tampilannya :



Gambar 5.11 Form Pembelian Barang

h. Form Return

Apabila Bag.Kasir membuka menu data maka akan muncul sebuah tampilan, lalu pilih tanggal transaksi penjualan yang akan di return. Tampilannya sebagai berikut :

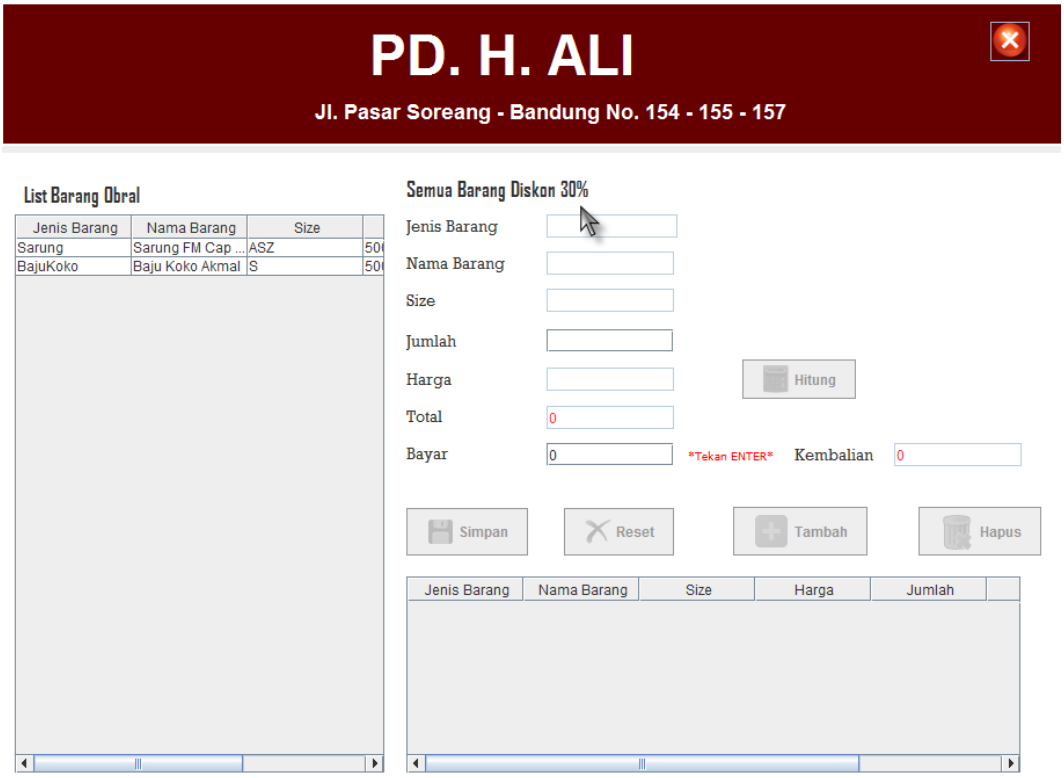


Gambar 5.12 Form Pembelian Barang

Tambah kedaftar" klik tanggal transaksi yang akan di return secara otomatis data return akan terisi pada *field*, lalu inputkan "JUMLAH" jumlah barang yang akan return, kemudian pilih keterangan return cacat atau ukuran. Lalu klik tombol 'SIMPAN' .

i. Form Obral

Apabila Bag.Kasir membuka menu transaksi maka akan muncul sebuah tampilan, lalu pilih obral. Tampilannya sebagai berikut :



Gambar 5.13 obral

Jika ingin mimiih barang obral maka klik namabarang yang ada di tabel sebelah kanan lalu inputkan "JUMLAH" barang obral yang akan di beli lalu klik "HITUNG" maka akan muncul "HARGA BAYAR"

j. Form Laporan Penjualan

Apabila Bag.Kasir membuka menu laporan lalu klik laporan data penjualan maka akan muncul tampilan berikut :



Gambar 5.14 Form Laporan Data Penjualan

Masukkan tanggal awal dan tanggal akhir sesuai dengan ketentuan. Halaman ini akan menampilkan dan mencetak laporan penjualan sesuai tanggal yang telah diinput.

5.2. Pengujian

Menurut Bambang Hariyanto, Ir., MT. (2004 : 569), pengujian adalah proses pemeriksaan atau evaluasi sistem atau komponen sistem secara manual atau otomatis untuk memverifikasi apakah sistem memenuhi kebutuhan-

kebutuhan yang dispesifikasikan atau mengidentifikasi perbedaan-perbedaan antara hasil yang diharapkan dengan hasil yang terjadi. Awalnya, pengujian diartikan sebagai aktivitas yang hanya dilakukan setelah pengkodean (kode program selesai). Namun, pengujian sebenarnya dilakukan dalam skala yang lebih luas. Pengujian dapat dilakukan begitu spesifikasi kebutuhan telah dapat didefinisikan. Evaluasi terhadap spesifikasi dan perancangan juga merupakan teknik dalam pengujian.

Adapun teknik pengujian perangkat lunak yang digunakan penulis dalam membangun sistem informasi penjualan pada PD. H .ALI ini, adalah teknik pengujian *blackbox testing.* Menurut Al-Bahra bin Ladjamudin (2006 : 379), metode *blackbox testing* ini berfokus pada keperluan fungsional dari *software*, yang memungkinkan untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Uji coba *blackbox* merupakan pendekatan yang melengkapi untuk menemukan kesalahan yang terjadi.

5.2.1. Rencana Pengujian

Pengujian perangkat lunak pada aplikasi Sistem Informasi Penjualan pada PD. H .ALI dengan menggunakan pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak yang dibuat. Berikut adalah rencana pengujian aplikasi Sistem Informasi Penjualan pada PD. H .ALI :

1. Pengujian unit yaitu pengujian ini difokuskan pada suatu unit dari program secara sendiri.
2. Pengujian penerimaan, yaitu pengujian yang meyakinkan bahwa sistem telah sesuai dengan kebutuhan perusahaan.

5.2.2. Kasus dan Hasil Pengujian

Setelah melakukan rencana pengujian, maka hal selanjutnya yang perlu dilakukan adalah melakukan studi kasus dan bagaimana hasil dari pengujian studi kasus tersebut. Adapun, kasus dan hasil pengujian yang dilakukan oleh penulis adalah sebagai berikut :

1. Pengecekan Login Admin

Berikut ini adalah kasus dan hasil pengujian login admin:

Tabel 5.10. Kasus dan Hasil Pengujian Pengecekan Login Admin

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username : Admin Password : Admin	Tercantum pada textbox username password	Mengisi login sesuai yang diharapkan	[X] Terima [ ] Ditolak
Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol login	Menampilkan “Login Berhasil” dan dapat mengakses sub menu yang ada dimenu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	[X] Terima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Username : admin Password : kasir	Tercantum pada textbox username password	Admin gagal melakukan login	[X] Terima [ ] Ditolak
Klik tombol login	Menampilkan "Username dan Password TIDAK VALID" dan tidak dapat masuk ke menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	[X] Terima [ ] Ditolak

2. Pengujian Pencarian Data Stock Barang

Berikut ini kasus dan hasil dari pengujian data jenis barang:

Tabel 5.11. Kasus dan Hasil Pengujian Data Stock Barang

Kasus dan Hasil Uji (Cek Barang)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Melakukan Cek barang dengan input nama barang, kode atau ukuran	Seluruh data yang ada dan apabila barang tidak ada akan tampil "Nama Barang Tidak Ada"	Dapat melakukan pencarian data barang dan pengecekan barang.	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Cek Barang)			
Nama barang diinput 'angka'	Pencarian tidak akan berhasil dan menampilkan "Data Tidak Ditemukan"	Tidak dapat melakukan pencarian atau pengecekan barang	[X] Diterima [ ] Ditolak

3. Pengujian Data Input Stock Barang

Berikut ini kasus dan hasil dari pengujian data stock barang :

Tabel 5.12 Kasus dan Hasil Pengujian Pengisian Data Input Stock Barang

Kasus dan Hasil Uji (Data Stock Barang)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Melakukan	Seluruh data barang	Data yang	[X] Diterima

pengecekan setelah menginputkan kode barang	di inputkan tidak terjadi penduplikatan data	diinputkan tersimpan kedalam database	[ ] Ditolak
Kasus dan Hasil Uji (Data Stock Barang Salah)			
Kode barang yang diinput sudah ada	Penyimpanan tidak dapat dilakukan	Tidak dapat melakukan penyimpanan data	[X] Diterima [ ] Ditolak

4. Pengujian Input Data Pembelian

Berikut ini kasus dan hasil dari pengujian input data pembelian :

Tabel 5.13 Kasus dan Hasil Pengujian Input Data Pembelian

Kasus dan Hasil Uji (Input Nota Pembelian)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
No pembelian, tanggal, nama barang, ukuran dan jumlah barang yang di beli muncul ketika di input no pembelian.	Data beli barang muncul pada *field* berikut nama supplier	Tersimpan di database dan barang di stock otomatis bertambah	[X] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Input Nota Pembelian Salah)			
Data transaksi tidak dinputkan dengan lengkap	Penyimpanan tidak dapat dilakukan	Tidak dapat melakukan penyimpanan data	[X] Diterima [ ] Ditolak

5. Pengujian Transaksi Penjualan

Berikut ini kasus dan hasil dari pengujian Transaksi Penjualan :

Tabel 5.14 Kasus dan Hasil Pengujian Transaksi Penjualan

Kasus dan Hasil Uji (Transaksi Penjualan)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Nama konsumen, kode barang apabila diinput lalu di *Enter*.	semua data barang akan muncul pada field ukuran, nama barang,harga. Ketika di simpan secara otomatis nota penjualan tercetak	Tersimpan di database dan nota penjualan langsung cetak	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Transaksi Penjualan)			
Data transaksi tidak dinputkan dengan lengkap	Penyimpanan tidak dapat dilakukan	Tidak dapat melakukan penyimpanan data	[X] Diterima [ ] Ditolak

1. Pengujian Beli Barang

Berikut ini kasus dan hasil dari pengujian Beli Barang :

Tabel 5.15 Kasus dan Hasil Pengujian Beli Barang

Kasus dan Hasil Uji (Beli Barang)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Kode barang, qty yang akan dibeli, kode supplier.	semua data barang akan muncul pada field ukuran, nama barang. Berikut nama supplier dan mencetak data beli barang ketika tombol ‘Simpan’ di klik	Tersimpan di database dan data beli barang langsung tercetak	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Beli Barang)			
Data beli barang tidak diisi lengkap	Penyimpanan dan cetak data belitidak dapat dilakukan	Tidak dapat melakukan barang penyimpanan data	[X] Diterima [ ] Ditolak

		dan cetak data beli barang.	

7. Pengujian Laporan

Berikut ini kasus dan hasil dari pengujian Laporan :

Tabel 5.16 Kasus dan Hasil Pengujian Laporan

Kasus dan Hasil Uji (Laporan)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Seluruh data laporan penjualan, pembelian data beli barang, dan data stock barang.	Klik tombol Preview untuk menampilkan dan mencetak laporan	Data-data laporan dapat ditampilkan serta dicetak	[X] Diterima [ ] Ditolak

5.2.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji *sample* diatas dapat ditarik kesimpulan bahwa perangkat lunak bebas dari kesalahan sintaks dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.