BAB IV

HASIL DAN PEMBAHASAN

4.1 Perancangan Sistem

Perancangan sistem adalah suatu gambaran dari perencanaan dan pembuatan sketsa atau pengaturan dari beberapa elemen yang terpisah kedalam satu kesatuan yang utuh dan berfungsi. Perancangan sistem ini menentukan bagaimana cara kerja suatu sistem akan menyelesaikan apa yang harus di selesaikan. Pada tahap ini berhubungan dengan cara mengkonfigurasikan dari komponen-komponen perangkat lunak dan perangkat keras dari sebuah sistem sehingga setelah instalasi dari sistem benar-benar memuaskan rancangan yang telah ditetapkan pada akhir tahap analisa sistem.

4.1.1 Tujuan Perancangan Sistem

1. Memperbaiki pengolahan data penjualan, pembelian, barang, dan pemasok pada Toko Willmart
2. Dapat mengetahui persediaan barang secara cepat.
3. Dapat membuat laporan secara tepat dan akurat.

4.1.2 Gambaran Umum Sistem Yang Diusulkan

Gambaran sistem yang diusulkan bermaksud untuk membangun suatu sistem informasi penjualan Toko Wilmart dengan secara terkomputerisasi. Sistem ini diharapkan dapat mempercepat dan mempermudah proses penjualan dibandingkan dengan sistem yang lama.

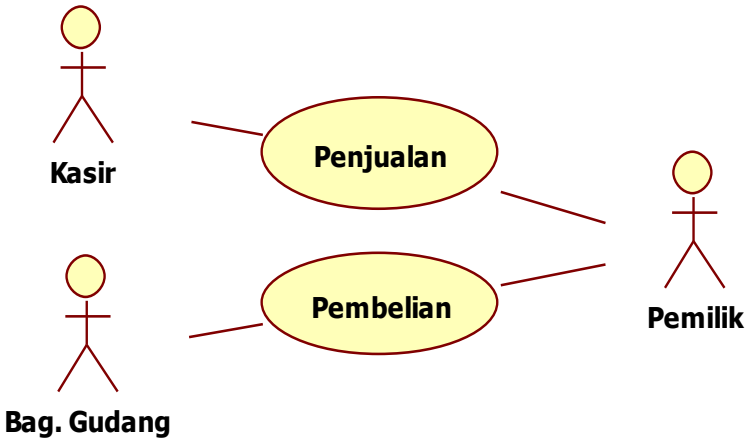
4.1.3 Perancangan Prosedur Yang Diusulkan

Dalam bagian ini perancangan prosedur yang diusulkan meliputi Use Case Diagram, Skenario Use Case, Activity Diagram, Sequence Diagram, dan Class Diagram.

Sistem ini menggunakan bahasa pemograman C# berbasis objek dengan program aplikasi yang digunakan adalah Visual Studio 2012 dan MySQL sebagai program untuk penyimpanan databasenya.

4.1.3.1 Use Case Diagram

Use Case Diagram atau diagram use case merupakan pemodelan untuk menggambarkan aktifitas sistem yang akan dibuat. Diagram use case menjelaskan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat. Diagram use case digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem dan siapa saja yang berhak menggunakan fungsi tersebut. Adapun use case sistem yang diusulkan di Toko Wilmart adalah sebagai berikut :



Gambar 4.1 Use Case Diagram Sistem Penjualan Yang Diusulkan

4.1.3.2 Skenario Use Case

Skenario *use case* menjelaskan tentang alur proses jalannya yang ada dalam *use case* diagram dari awal proses sampai akhir, untuk lebih jelasnya prosesnya sama dengan metodolgi *object oriented* sehingga digunakan untuk menjelaskan prosedur secara global adalah menggunakan scenario use case. Adapun scenario use case sistem yang diusulkan di Toko Wilmart adalah sebagai berikut :

1. Nama Use Case : Login

   Actor : Kasir, Bag. Gudang, Pemilik

   Tujuan : Proses Validasi

Tabel 4.1 Skenario *Use Case* login yang diusulkan

Aktor	Sistem
1. Masukan Id Peg dan Password	
	2. Validasi Id Peg dan Password
	3. Menampilkan notifikasi login tidak valid/salah
4. Masukan ulang Id Peg dan Password	
	5. Validasi Id Pegawai dan Password
	6. Masuk ke menu utama

2. Nama Use Case : Penjualan

   Actor : Kasir

   Tujuan : Transaksi Penjualan

Tabel 4.2 Skenario *Use Case* Penjualan

Aktor	**Sistem**
1. Memilih menu transaksi	
	2. Menampilkan Menu transaksi
3. Memilih transaksi penjualan	
	4. Menampilkan menu transaksi penjualan
5. Input barang	
	6. Menampilkan nama barang dan harga
7. Input jumlah beli	
	8. Menampilkan total
9. Input Bayar	
	10. Menampilkan kembalian
	11. Catat Transaksi

3. Nama Use Case : Pembelian

   Actor : Bag. Gudang

   Tujuan : Transaksi pembelian kepada pemasok

Tabel 4.3 Skenario *Use case* Pembelian

Aktor	Sistem
1. Masuk menu utama	
	2. Menampilkan menu utama
3. Memilih menu data barang	
	4. Menampilkan menu data barang
5. Mengecek stok barang habis	
	6. Menampilkan stok barang habis
7. Membeli barang pada pemasok dan menyimpan stok barang	
	8. Save data barang
9. Cetak Data Barang	

4. Nama Use Case : Laporan

   Actor : Kasir, Bag. Gudang, dan Pemilik

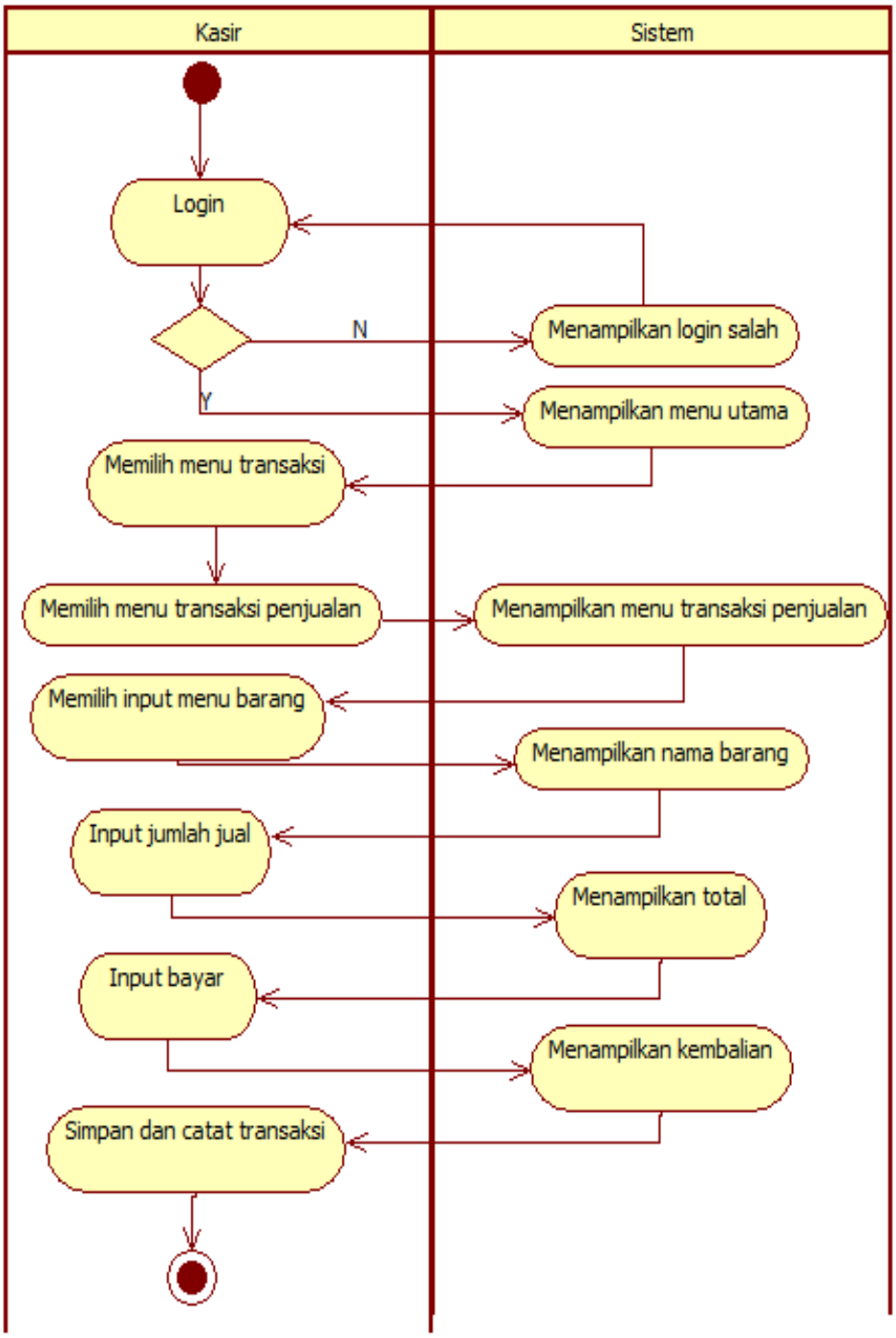
   Tujuan : Mengecek dan Menampilkan Laporan

Tabel 4.4 Skenario *Use Case* Laporan

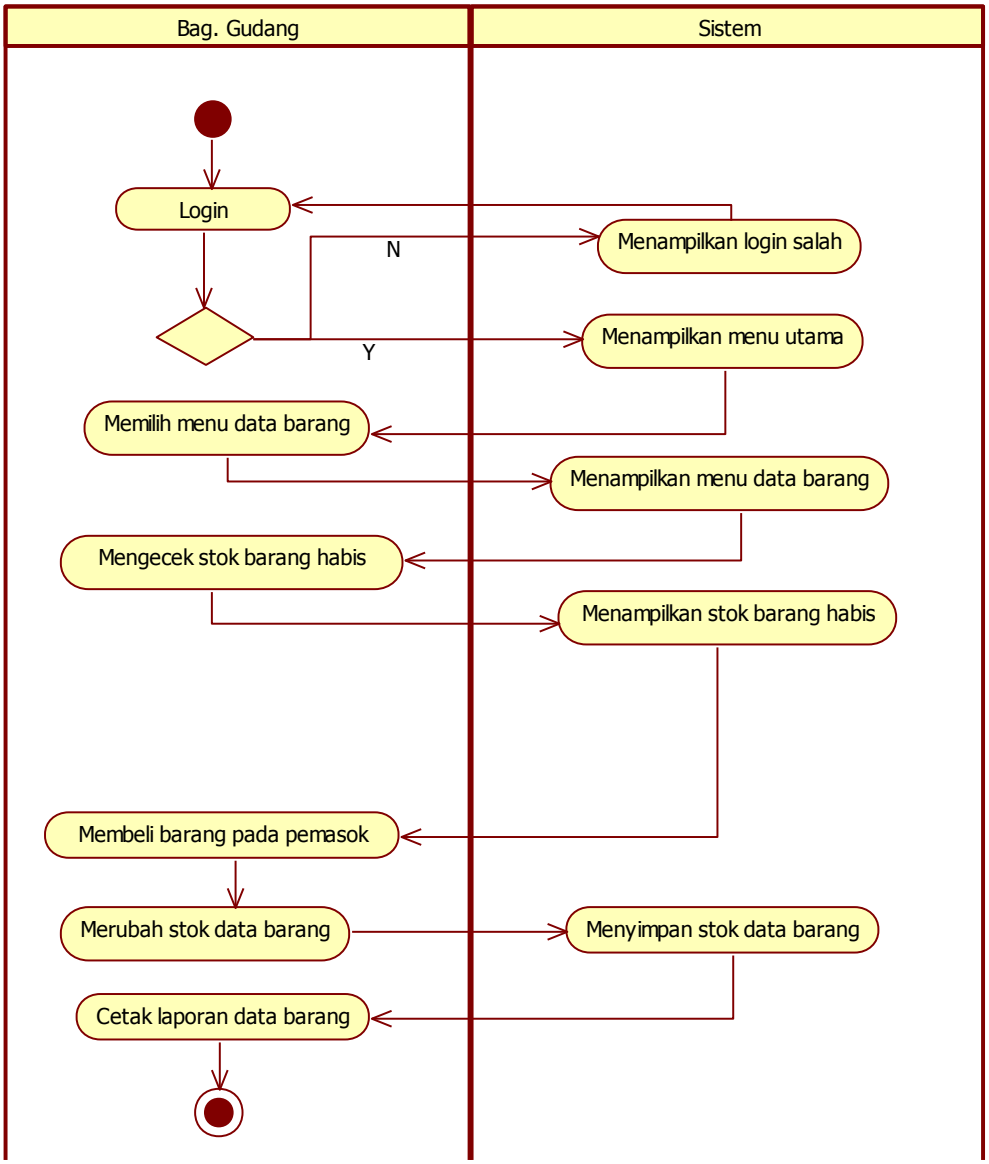
Aktor	Sistem
1. Memilih menu laporan	
	2. Menampilkan Menu laporan
3. Memilih Laporan barang, laporan Stok barang atau laporan transaksi	
	4. Menampilkan laporan yang dipilih

4.1.3.3 Activity Diagram

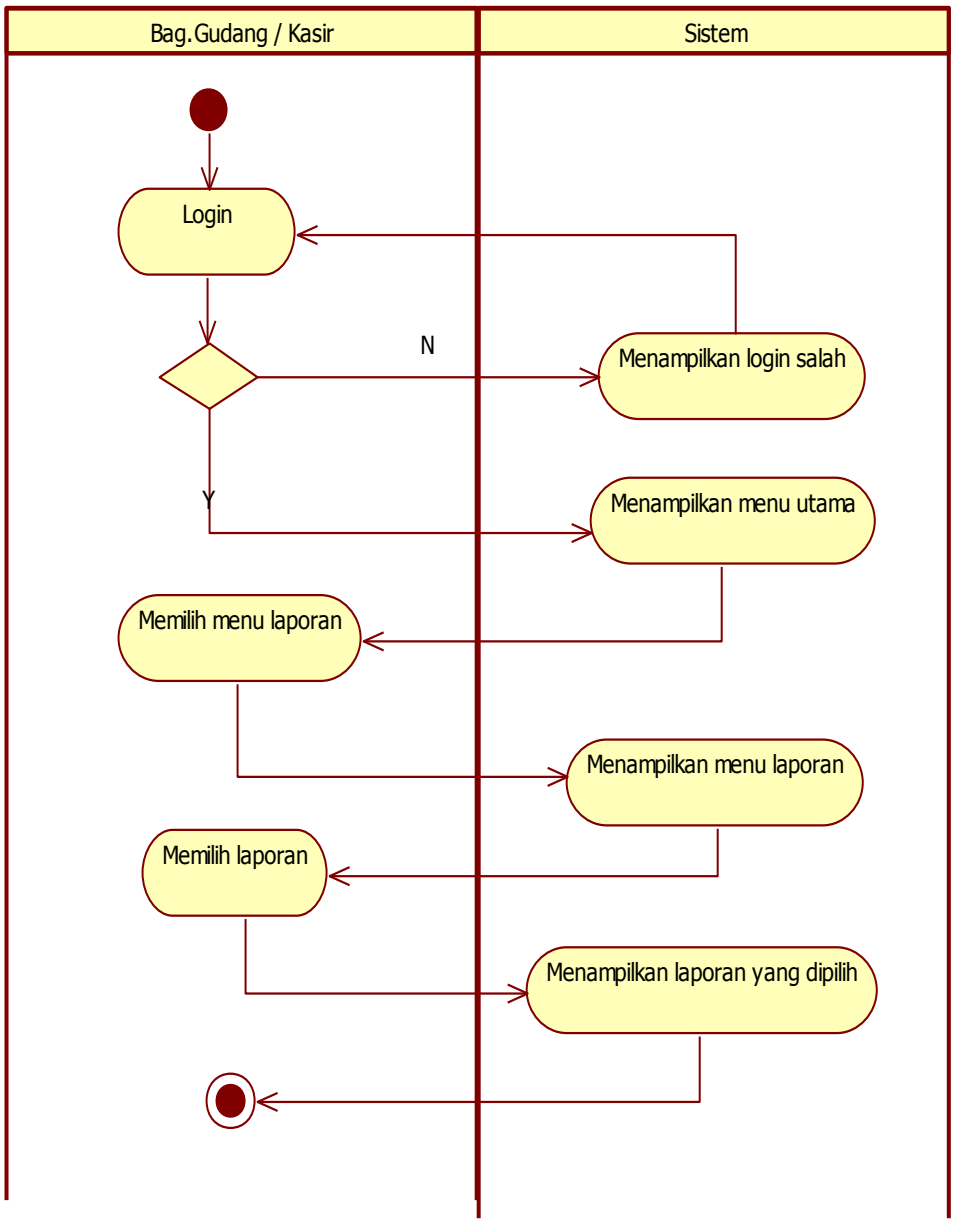
Activity Diagram atau diagram aktifitas dapat digunakan untuk menunjukan aliran kerja bisnis (business work-flow). Dapat juga digunakan untuk menggambarkan aliran kejadian (flow of events) dalam use case. Adapun activity diagram pada Toko Wilmart adalah sebagai berikut :



Gambar 4.2 *Activity* diagram login dan penjualan yang diusulkan



Gambar 4.3 *Activity* diagram pembelian yang diusulkan

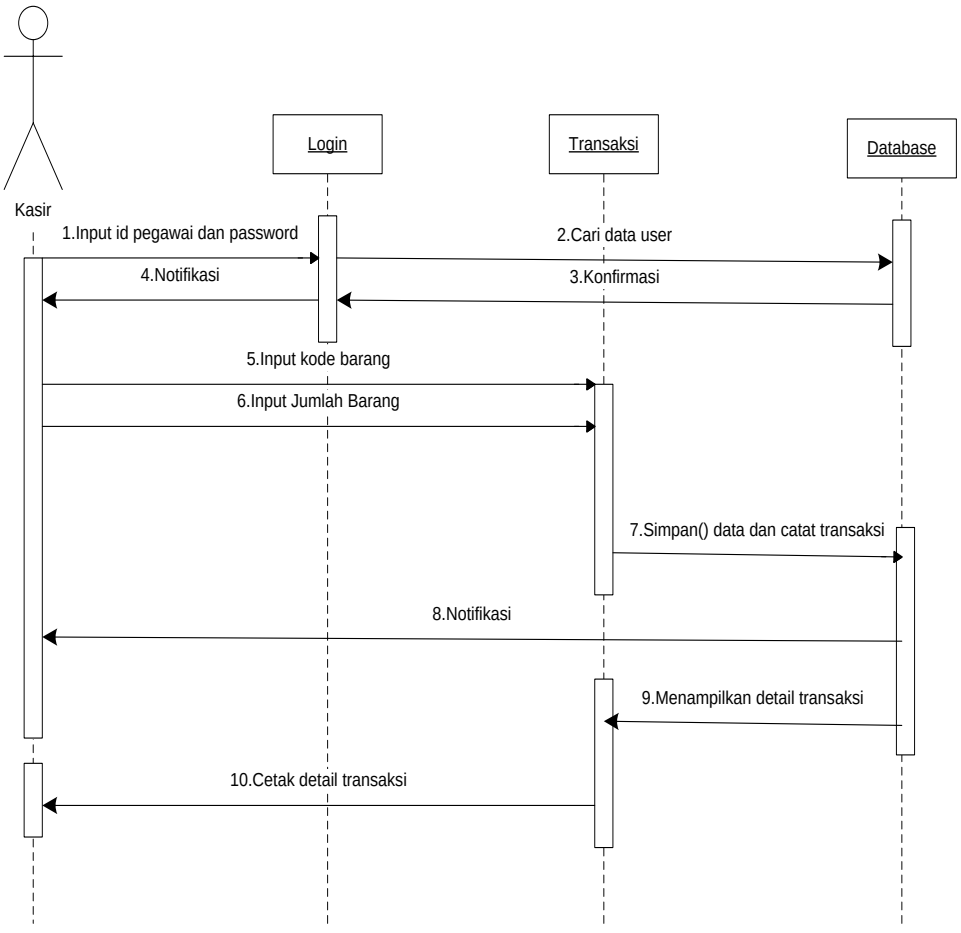


Gambar 4.4 *Activity* diagram laporan yang diusulkan

4.1.3.4 Sequence Diagram

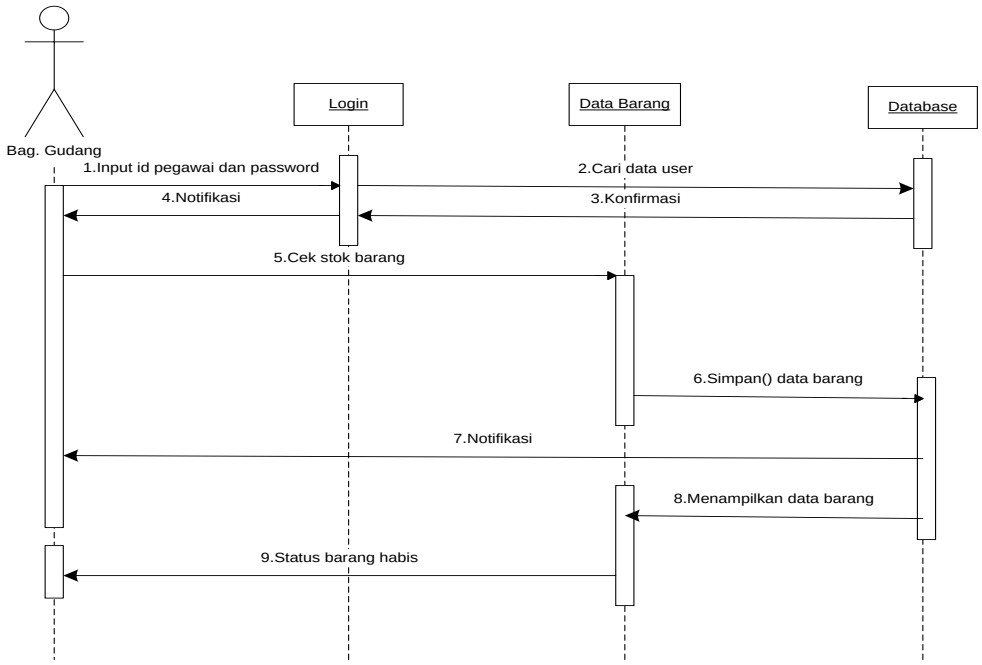
Menggambarkan kolaborasi dinamis antara sejumlah objek. Kegunaannya untuk menunjukkan rangkaian pesan yang dikirim antara objek juga interaksi antara objek, sesuatu yang terjadi pada titik tertentu dalam eksekusi sistem. Komponen utama sequence diagram terdiri atas objek yang dituliskan dengan segiempat bernama pesan diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan proses vertikal.

1. Sequence Diagram Sistem Penjualan



Gambar 4.5 Sequence Diagram Sistem Penjualan

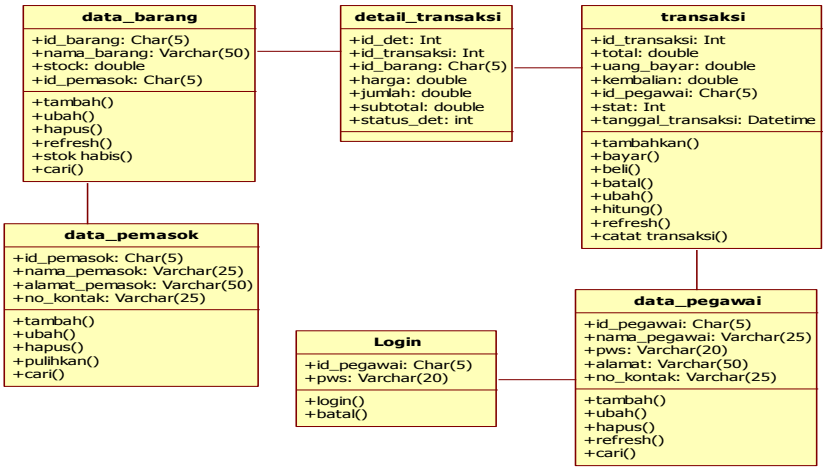
2. Sequence Diagram Sistem Pembelian



Gambar 4.6 Sequence Diagram Sistem Pembelian

4.1.3.5 Class Diagram

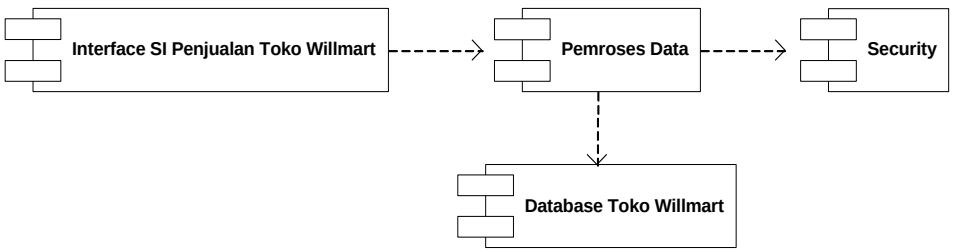
Menggambarkan struktur statis class di dalam sistem. Class merepresentasikan sesuatu yang ditangani oleh sistem. Dengan melihat karakteristik sistem pemasaran produk dari bagian penjualan beserta proses-proses yang terjadi, maka dapat dibuat Class Diagram Berikut Class Diagram Sistem Informasi penjualan pada Toko Wilmart.



Gambar 4.7 Class Diagram

4.1.3.6 Component Diagram

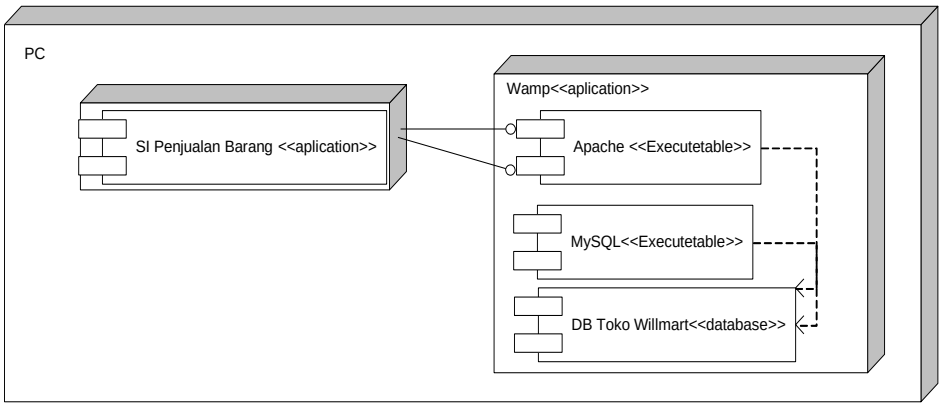
Component diagram menggambarkan struktur dan hubungan antar komponen perangkat lunak, termasuk ketergantungan di antaranya. Component perangkat lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain. Berikut ini adalah Component Diagram yang dibutuhkan :



Gambar 4.8 Component Diagram Penjualan

4.1.3.7 Deployment Diagram

Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan. Diagram ini memuat simpul-simpul beserta komponen-komponen yang ada didalamnya. Deployment diagram berhubungan dengan diagram komponen dimana deployment diagram memuat satu atau lebih komponen-komponen.

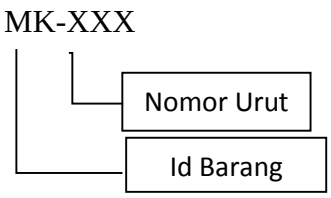


Gambar 4.9 Deployment Diagram Penjualan

4.1.3.8 Kodefikasi

Kodefikasi digunakan untuk menjabarkan item – item data yang bersifat unik. Adapun pengkodean pada struktur file di atas adalah sebagai berikut :

1. Kode Barang



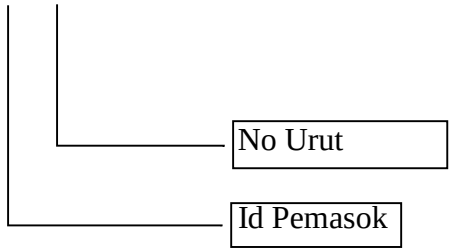
Contoh : MK-001

MK : Menunjukan id Barang

001 : Menunjukkan no urut Barang.

2. Kode Pemasok

MD-XXX



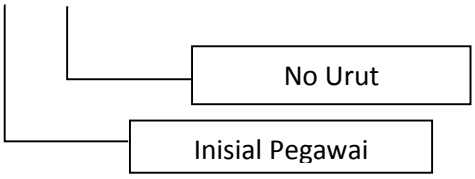
Contoh : MD - 001

MD : Menunjukan id Pemasok

001 : Menunjukan no urut pemasok

3. Kode Pegawai

PG-XXX



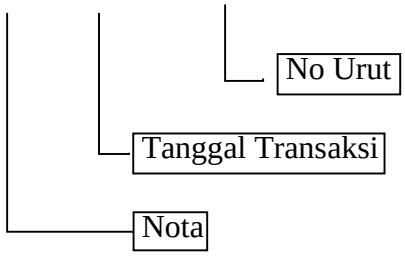
Contoh : PG-001

PG : Menunjukan inisial Pegawai

001 : Menunjukan no urut pegawai

4. No Nota

NXXXXXXXX-XXX



Contoh : N01092014-001

N : Menunjukan inisial Nota

01092014 : Menunjukan tanggal transaksi penjualan

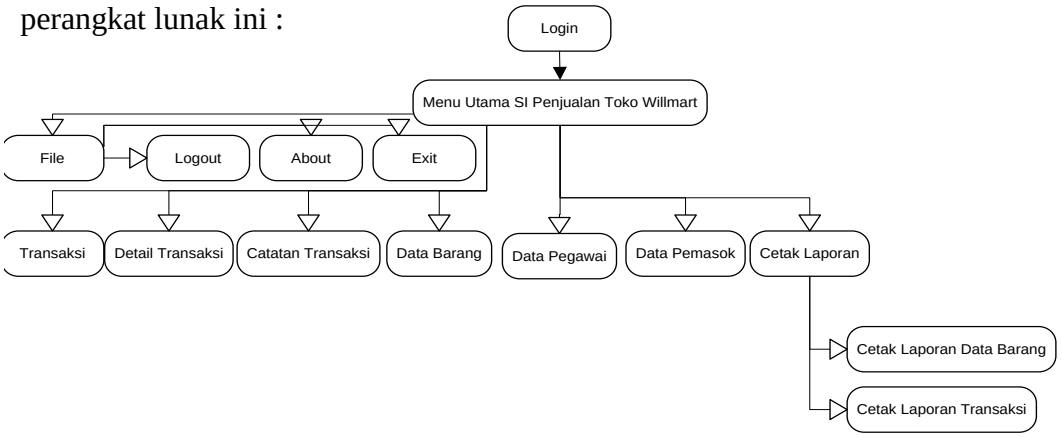
01 : Menunjukan no urut transaksi

4.2 Perancangan Antar Muka

Perancangan antar muka (interface) merupakan suatu tahapan agar aplikasi yang dibuat lebih menarik dan tentunya lebih mudah digunakan. Tujuan dari perancangan antar muka pada adalah untuk mendapatkan suatu criteria yang sangat penting dalam pengoprasian sebuah program aplikasi, yaitu aspek ramah dengan pengguna (user friendly).

4.2.1 Perancangan Struktur Menu

Perancangan menu dibuat sebagai alat antar muka dengan pengguna untuk memudahkan pengoperasian perangkat lunak. Berikut rancangan menu perangkat lunak ini :



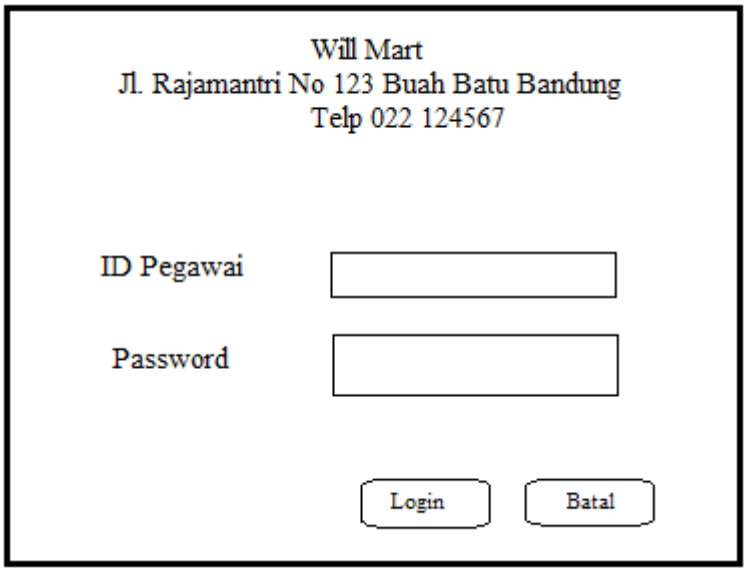
Gambar 4.10 Rancangan Struktur Menu

4.2.2 Perancangan Aplikasi

1. Nama Dialog Screen : Halaman Login

   Fungsi : Sebagai tampilan awal aplikasi

   Bentuk :



Gambar 4.11 Rancangan Tampilan Awal Login

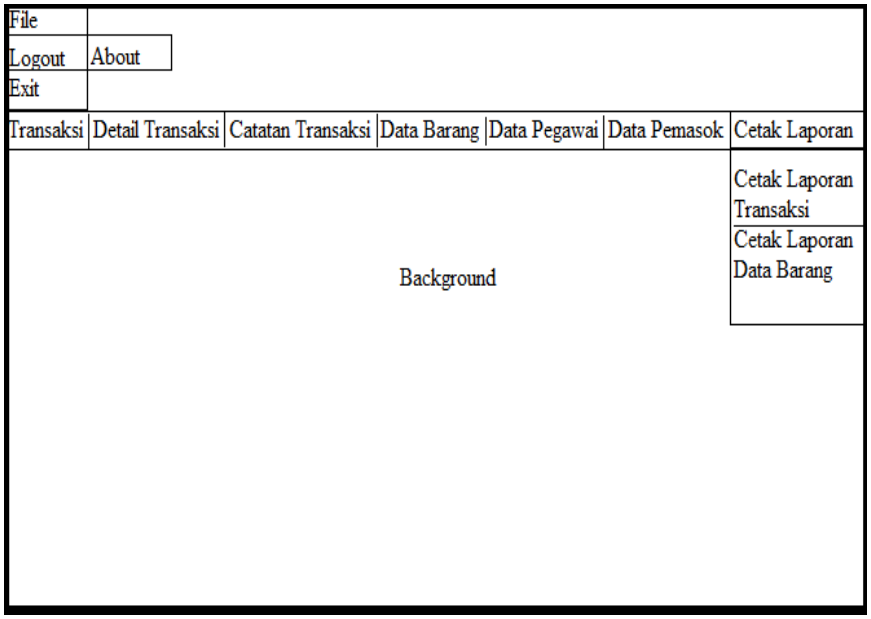
Uraian Penggunaan :

1. Kolom Id Pegawai digunakan untuk input id pegawai.
2. Kolom Password digunakan untuk input password.
3. Tombol login digunakan untuk melakukan proses login.
4. Tombol Batal digunakan untuk keluar halaman login.

2. Nama Dialog Screen : Halaman Menu

   Fungsi : Sebagai tampilan setelah login yang memiliki banyak sub menu

   Bentuk :



Gambar 4.12 Rancangan Tampilan Menu Utama

Uraian Penggunaan :

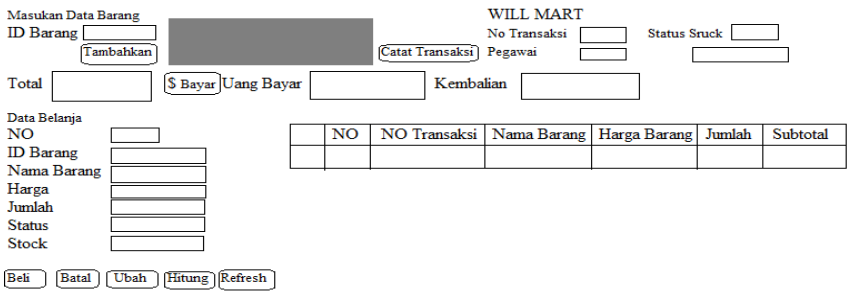
1. Submenu File memiliki pilihan untuk Logout, About dan Exit
2. Pilihan Logout digunakan untuk keluar dari aplikasi kembali ke halaman awal Login.
3. Pilihan Exit adalah untuk keluar dari aplikasi.
4. Pilihan Transaksi digunakan untuk melakukan transaksi pembayaran dan penjualan.
5. Pilihan Detail Transaksi digunakan untuk melihat detail data transaksi yang telah dilakukan.
6. Pilihan Catatan Transaksi digunakan untuk menampilkan Laporan Data Transaksi yang sudah dilakukan.

7. Pilihan Data Barang digunakan untuk menampilkan Data Barang yang ada.
8. Pilihan Data Pegawai adalah untuk menampilkan Data Pegawai yang ada.
9. Pilihan Data Pemasok digunakan untuk menampilkan Data Pemasok yang ada.
10. Submenu Cetak Laporan memiliki pilihan untuk mencetak dan menampilkan Laporan Transaksi dan Laporan Data Barang.
11. Pilihan Cetak Laporan Transaksi digunakan untuk menampilkan dan mencetak Laporan Transaksi.
12. Pilihan Cetak Laporan Data Barang digunakan untuk menampilkan dan mencetak Laporan Data Barang.
13. Pilihan About digunakan untuk menampilkan form about.

3. Nama Dialog Screen : Halaman Transaksi

   Fungsi : Untuk proses Transaksi Penjualan

   Bentuk :



Gambar 4.13 Rancangan Tampilan Transaksi

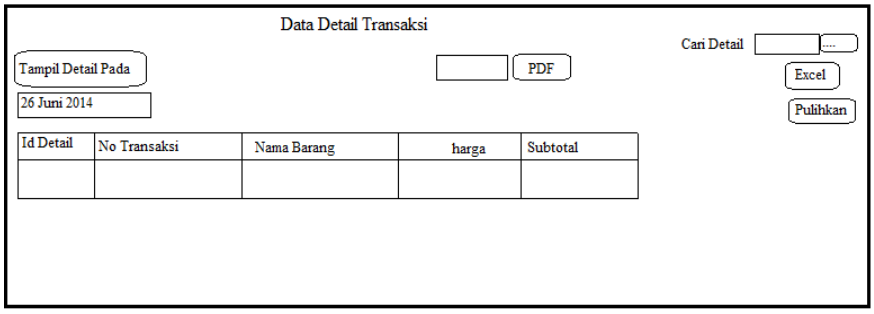
Uraian Penggunaan :

1. Tombol Tambahkan digunakan untuk menambah barang yang akan ditransaksi.
2. Tombol Bayar digunakan untuk menghitung jumlah pembayaran dan kembalian yang ditransaksikan.
3. Tombol Catat Transaksi digunakan untuk mencatat transaksi yang telah dilakukan.
4. Tombol Beli digunakan untuk membeli barang yang akan ditransaksi.
5. Tombol Batal digunakan untuk membatalkan transaksi.
6. Tombol Ubah digunakan untuk merubah data barang yang akan ditransaksikan.
7. Tombol Hitung digunakan untuk menghitung jumlah barang yang akan ditransaksikan.
8. Tombol Refresh digunakan untuk membersihkan dan mengulang transaksi.

4. Nama Dialog Screen : Halaman Detail Transaksi

   Fungsi : Untuk menampilkan Data Transaksi

   Bentuk :



Gambar 4.14 Rancangan Tampilan Data Transaksi

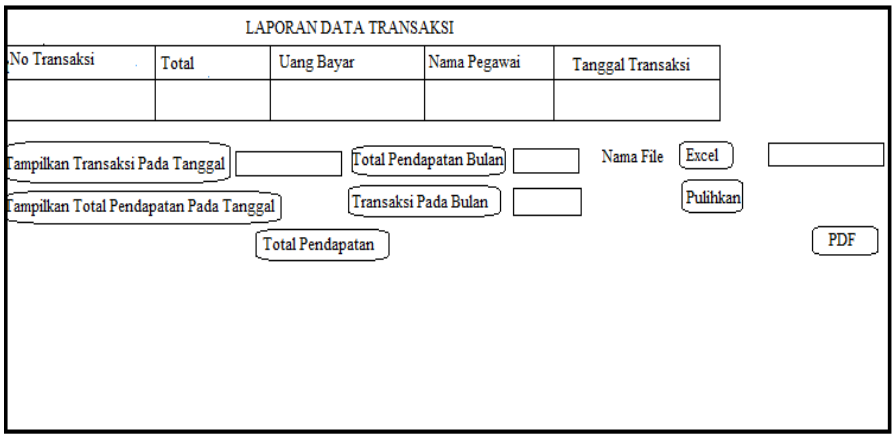
Uraian Penggunaan :

1. Tombol Tampil Detail Pada digunakan untuk menampilkan data transaksi pada waktu dan tanggal yang ingin dilihat.
2. Tombol PDF digunakan untuk mencetak Detail Transaksi dalam bentuk PDF.
3. Tombol titik – titik digunakan untuk mencari detail data.
4. Tombol Excel digunakan untuk menampilkan Detail Transaksi Pada Ms. Excel.
5. Tombol Pulihkan digunakan untuk memulihkan Data Detail Transaksi.

5. Nama Dialog Screen : Halaman Catatan Transaksi

   Fungsi : Untuk menampilkan Laporan Transaksi

   Bentuk :



Gambar 4.15 Rancangan Tampilan Catatan Transaksi

Uraian Penggunaan :

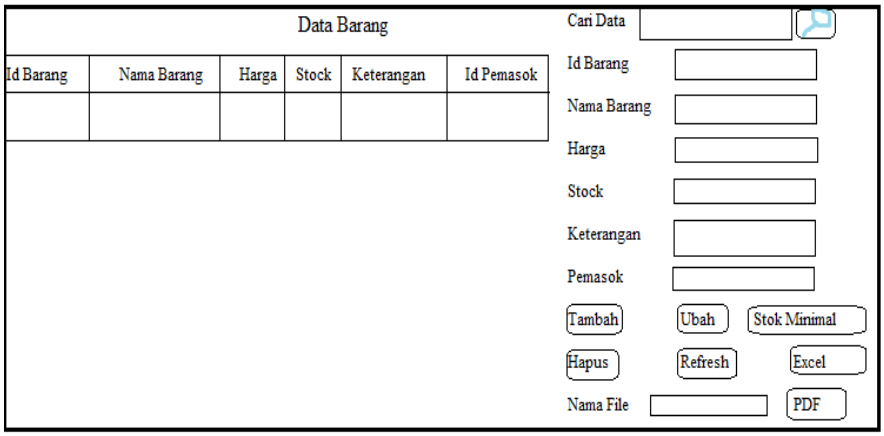
1. Tombol Tampilkan Transaksi Pada Tanggal digunakan untuk menampilkan data transaksi pada tanggal yang ingin ditampilkan.

2. Tombol Total Pendapatan Pada Tanggal digunakan untuk menampilkan total data pendapatan pada tanggal yang ingin ditampilkan.
3. Tombol Total Pendapatan Bulan digunakan untuk menampilkan total data pendapatan pada bulan yang ingin ditampilkan.
4. Tombol Transaksi Pada Bulan digunakan untuk menampilkan data transaksi pada bulan yang ingin ditampilkan.
5. Tombol Excel digunakan untuk menampilkan Laporan Data Transaksi ke dalam Ms. Excel.
6. Tombol Pulihkan digunakan untuk memulihkan Laporan Data Transaksi.
7. Tombol PDF digunakan untuk mencetak Laporan Data Transaksi dalam bentuk PDF.
8. Tombol Total Pendapatan untuk menghitung keseluruhan data transaksi.

6. Nama Dialog Screen : Halaman Data Barang

   Fungsi : Untuk menampilkan Data Barang

   Bentuk :



Gambar 4.16 Rancangan Tampilan Data Barang

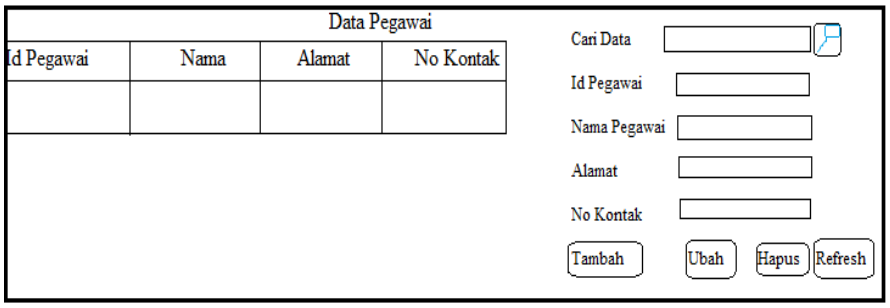
Uraian Penggunaan :

1. Tombol Search digunakan untuk mencari data barang.
2. Tombol Tambah digunakan untuk menambahkan stok barang.
3. Tombol Hapus digunakan untuk menghapus data barang.
4. Tombol Ubah digunakan untuk merubah data barang.
5. Tombol Refresh digunakan untuk memulihkan data barang.
6. Tombol Excel digunakan untuk menampilkan data barang ke dalam bentuk Ms. Excel.
7. Tombol Stok Minimal digunakan untuk menampilkan data stok barang yang sudah mau habis.
8. Tombol PDF digunakan untuk mencetak Data Barang ke dalam bentuk PDF.

7. Nama Dialog Screen : Halaman Data Pegawai

   Fungsi : Untuk menampilkan Data Pegawai

   Bentuk :



Gambar 4.17 Rancangan Tampilan Data Pegawai

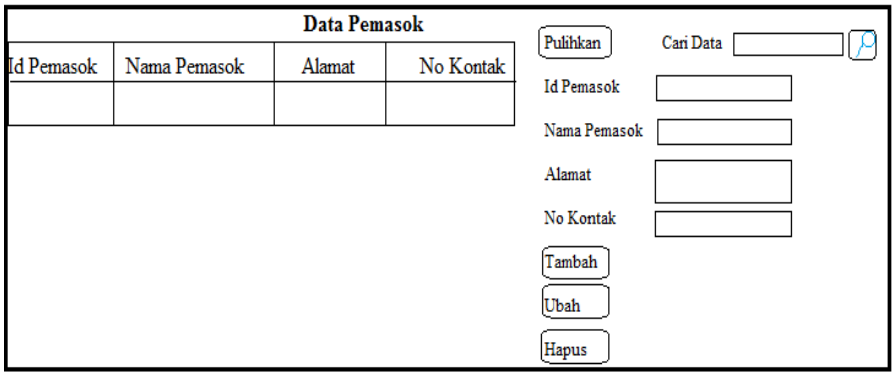
Uraian Penggunaan :

1. Tombol Tambah digunakan untuk menambahkan Data Pegawai.
2. Tombol Ubah digunakan untuk merubah Data Pegawai.
3. Tombol Hapus digunakan untuk menghapus Data Pegawai.
4. Tombol Search digunakan untuk mencari Data Pegawai.
5. Tombol Refresh digunakan untuk memulihkan Data Pegawai.

8. Nama Dialog Screen : Halaman Data Pemasok

   Fungsi : Untuk menampilkan Data Pemasok

   Bentuk :



Gambar 4.18 Rancangan Tampilan Data Pemasok

Uraian Penggunaan :

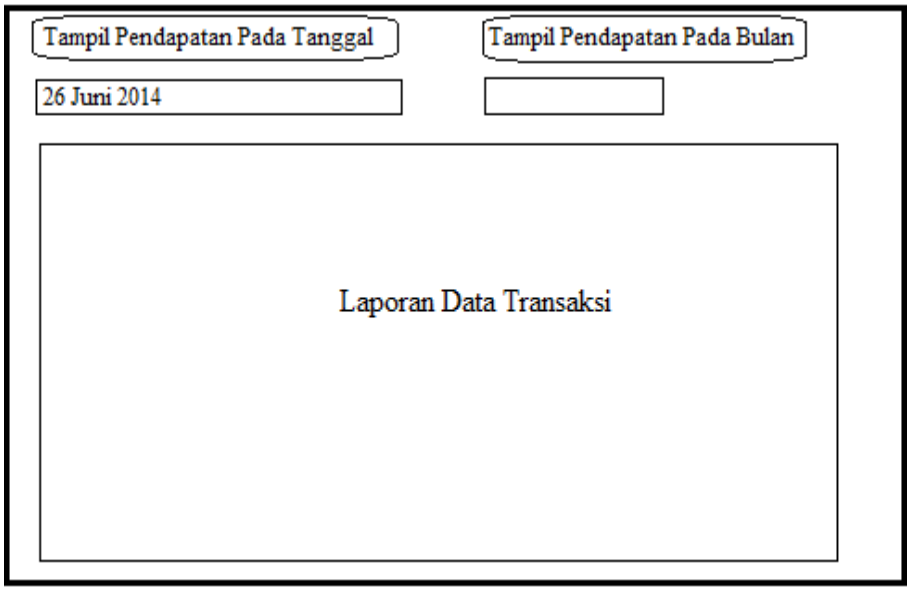
1. Tombol Search digunakan untuk mencari Data Pemasok.

2. Tombol Tambah digunakan untuk menambahkan Data Pemasok.
3. Tombol Ubah digunakan untuk merubah Data Pemasok.
4. Tombol Hapus digunakan untuk menghapus Data Pemasok.
5. Tombol Pulihkan digunakan untuk memulihkan halaman Data Pemasok.

9. Nama Dialog Screen : Halaman Cetak Laporan Transaksi

   Fungsi : Untuk mencetak Laporan Transaksi

   Bentuk :



Gambar 4.19 Rancangan Tampilan Cetak Laporan Transaksi

Uraian Penggunaan :

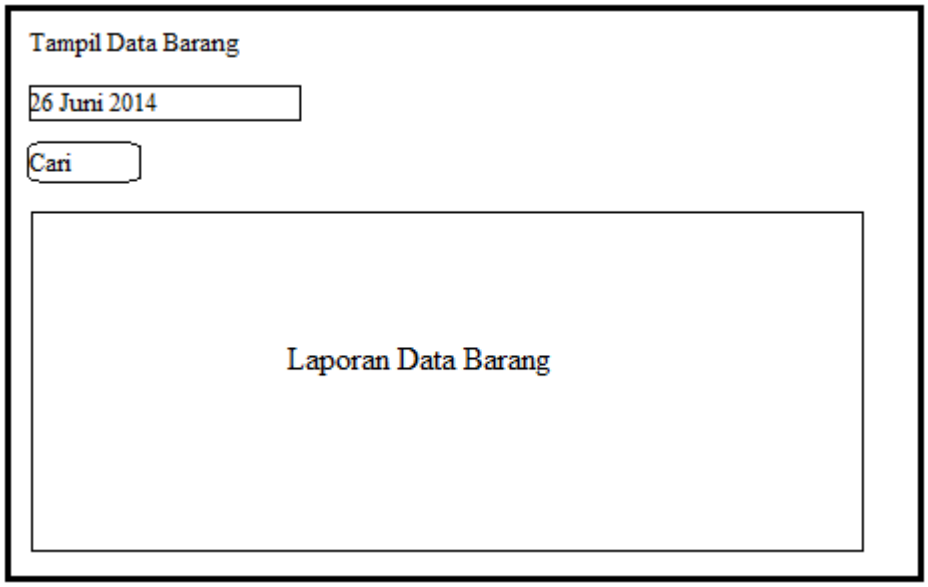
1. Tombol Tampil Pendapatan Pada Tanggal digunakan untuk menampilkan Laporan Data Transaksi pada tanggal yang ingin ditampilkan dan dicetak.

2. Tombol Tampil Pendapatan Pada Bulan digunakan untuk menampilkan Laporan Data Transaksi pada bulan yang ingin ditampilkan dan dicetak

10. Nama Dialog Screen : Halaman Cetak Laporan Data Barang

    Fungsi : Untuk mencetak Laporan Data Barang

    Bentuk :



Gambar 4.20 Rancangan Tampilan Cetak Laporan Data Barang

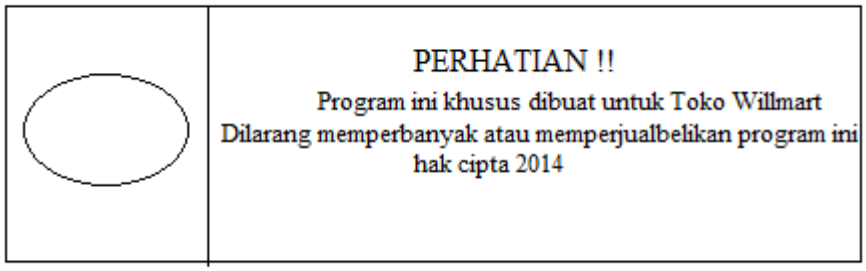
Uraian Penggunaan :

1. Tombol Cari digunakan untuk menampilkan Laporan Data Barang yang akan dicetak.

11. Nama Dialog Screen : Halaman About

    Fungsi : Untuk menampilkan informasi aplikasi

    Bentuk :



Gambar 4.21 Rancangan Tampilan About

4.3 Implementasi

Implementasi merupakan akhir dari sebuah pengembangan system informasi. Implementasi sistem tentunya mengacu pada rancangan sistem yang telah dibuat. Pada umumnya untuk implementasi sebuah sistem diperlukan konfigurasi pada perangkat lunak serta perangkat keras sesuai kebutuhan sistem supaya sistem informasi bisa berjalan dengan semestinya.

4.3.1 Batasan Implementasi

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

1. Dalam proses penjualan hanya melayani pembayaran tunai
2. Dalam proses pembelian tidak membahas mengenai laporan pembelian.
3. Basis data yang digunakan dalam pengimplementasian ini adalah MySQL.
4. Bahasa pemograman adalah C#.

5. Sistem ini tidak diintegrasikan dengan sistem yang dimiliki pemasok, dikarenakan pihak pemasok belum siap menggunakan sistem yang pemilik gunakan.

4.3.2 Implementasi Perangkat Lunak

Pada tahap implementasi perangkat lunak ini, didefinisikan perangkat lunak lainnya yang dijadikan pendukung bagi sistem. Perangkat lunak tersebut diantaranya :

1. Sistem Operasi yang digunakan adalah Microsoft Windows 7
2. WampServer
3. Microsoft Visual Studio C# 2012
4. InstallShield Wizard

4.3.3 Implementasi Perangkat Keras

Kebutuhan perangkat keras minimum ntuk bisa mendukung perancangan program dan untuk pengoperasian program yang dirancang adalah sebagai berikut :

1. Processor minimum Intel Pentium IV
2. Hardisk 80 GB
3. RAM 1 GB
4. VGA 128 MB
5. Mouse, Keyboard, Monitor, dan Printer

4.3.4 Implementasi Basis Data

Pada bagian implementasi basis data akan diuraikan pembuatan basis data yang dilakukan dengan menggunakan bahasa SQL, dimana DBMS yang

digunakan adalah MySQL. Implementasi basis data dalam bahasa SQL adalah sebagai berikut :

```
/*
SQLyog Ultimate v9.50
MySQL - 5.5.20-log : Database - si_willmart
*********************************************************************
********
*/

/*!40101 SET NAMES utf8 */;

/*!40101 SET SQL_MODE=''*/;

/*!40014 SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;
/*!40111 SET @OLD_SQL_NOTES=@@SQL_NOTES, SQL_NOTES=0 */;
```

```
CREATE DATABASE /*!32312 IF NOT EXISTS*/`si_willmart` /*!40100
DEFAULT CHARACTER SET latin1 */;

USE `si_willmart`;

/*Table structure for table `data_barang` */

DROP TABLE IF EXISTS `data_barang`;

CREATE TABLE `data_barang` (
  `id_barang` char(5) NOT NULL,
  `nama_barang` varchar(50) DEFAULT NULL,
  `harga_barang` double unsigned NOT NULL,
  `stock` double unsigned NOT NULL,
  `keterangan` varchar(25) NOT NULL DEFAULT 'Tidak ada keterangan',
  `id_pemasok` char(5) NOT NULL,
  PRIMARY KEY (`id_barang`),
  KEY `id_pemasok` (`id_pemasok`),
  CONSTRAINT `data_barang_ibfk_1` FOREIGN KEY (`id_pemasok`)
REFERENCES `data_pemasok` (`id_pemasok`) ON DELETE CASCADE
ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Data for the table `data_barang` */
```

```
insert into `data_barang`(`id_barang`,`nama_barang`,`harga_barang`,`stock`,`keterangan`,`id_pemasok`) values ('','COKLAT',1500,10,'BATANG','MK001'),('MK001','Indomie Ayam Special',1500,16,'2 dus','MK001'),('MK002','Indomie Goreng',2000,25,'4 dus','MK001'),('MK003','Indomie Goreng Ijo',2000,14,'3 dus','MK001'),('MK004','Indomie',2000,47,'stock 24 bungkus','MK001'),('MK005','Fanta Botol',3500,16,'12 botol','MK003'),('MK006','Bengbeng',1000,0,'','MK002'),('MK007','Susu Indomilk 150 ml',3500,19,'nn','MK002'),('MK008','Susu Milo',4500,20,'nmgh','MK001'),('MK009','Top',1000,20,'nmgh','MK001'),('MK010','Richis Nabati',1000,20,'nmgh','MK001'),('MK012','OReo',1000,20,'nmgh','MK001'),('MK020','Moca chino',1500,21,'2 dus','MK001'),('MK021','Torabika',1500,21,'2 dus','MK001'),('MK022','Torabika',1500,21,'2 dus','MK001'),('MK023','Torabika',1500,21,'2 dus','MK001'),('MK024','Kopi Indocafe',1000,12,'harga 1000 per kopi','MK002'),('MKN00','Mie Sedap Kari ',1500,63,'12','MK001');

/*Table structure for table `data_pegawai` */

DROP TABLE IF EXISTS `data_pegawai`;
```

```
CREATE TABLE `data_pegawai` (
  `id_pegawai` char(5) NOT NULL,
  `nama_pegawai` varchar(25) NOT NULL,
  `pws` varchar(20) NOT NULL,
  `alamat` varchar(50) NOT NULL,
  `no_kontak` varchar(25) NOT NULL,
  PRIMARY KEY (`id_pegawai`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Data for the table `data_pegawai` */

insert into
`data_pegawai`(`id_pegawai`,`nama_pegawai`,`pws`,`alamat`,`no_kontak`)
values
('PG001','Iwan','abcd','CIamis','123'),('PG002','Aceng','12345','Bandung','08
123456'),('PG003','Samid baharudin','123456','Jl Belahmana Bumina No ah
sabaraha wae','1234556');

/*Table structure for table `data_pemasok` */

DROP TABLE IF EXISTS `data_pemasok`;

CREATE TABLE `data_pemasok` (
```

```
  `id_pemasok` char(5) NOT NULL,
  `nama_pemasok` varchar(25) DEFAULT NULL,
  `alamat_pemasok` varchar(50) NOT NULL,
  `no_kontak` varchar(25) NOT NULL,
  PRIMARY KEY (`id_pemasok`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

/*Data for the table `data_pemasok` */

insert into `data_pemasok`(`id_pemasok`,`nama_pemasok`,`alamat_pemasok`,`no_kontak`) values ('MK001','Indofood','Jl.Pagarsih Barat','08123456789'),('MK002','Sosro','Jl. Pejaung No 123','081234587'),('MK003','Cocacola','Jl Ci Baregbeg','10393847');

/*Table structure for table `detail_transaksi` */

DROP TABLE IF EXISTS `detail_transaksi`;

CREATE TABLE `detail_transaksi` (
  `id_det` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `id_transaksi` int(10) unsigned NOT NULL,
  `id_barang` char(5) NOT NULL,
  `harga` double DEFAULT NULL,
```

```
  `jumlah` double unsigned DEFAULT '0',
  `subtotal` double DEFAULT NULL,
  `status_det` int(11) DEFAULT '0',
  PRIMARY KEY (`id_det`),
  KEY `id_transaksi` (`id_transaksi`),
  KEY `id_barang` (`id_barang`),
  CONSTRAINT `detail_transaksi_ibfk_1` FOREIGN KEY (`id_transaksi`)
REFERENCES `transaksi` (`id_transaksi`) ON DELETE CASCADE ON
UPDATE CASCADE,
  CONSTRAINT `detail_transaksi_ibfk_2` FOREIGN KEY (`id_barang`)
REFERENCES `data_barang` (`id_barang`) ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=71 DEFAULT
CHARSET=latin1;

/*Data for the table `detail_transaksi` */

insert into
`detail_transaksi`(`id_det`,`id_transaksi`,`id_barang`,`harga`,`jumlah`,`subto
tal`,`status_det`) values
(19,11,'MKN00',1500,2,NULL,1),(20,15,'MKN00',1500,2,NULL,1),(21,14,'
MK004',2000,0,NULL,1),(22,16,'MK002',2000,2,NULL,1),(23,16,'MK001',
2000,1,NULL,1),(24,16,'MK001',2000,1,NULL,1),(25,16,'MK003',2000,1,
NULL,1),(26,16,'MK001',2000,2,NULL,1),(27,16,'MK003',2000,1,NULL,1
),(28,17,'MK003',2000,1,NULL,1),(29,18,'MK004',2000,2,NULL,1),(30,19,'
```

```
MK004',2000,1,NULL,1),(31,20,'MKN00',1500,1,NULL,1),(32,21,'MKN00
',1500,2,NULL,1),(33,22,'MK004',2000,0,NULL,1),(34,22,'MKN00',1500,1
,NULL,1),(36,23,'MKN00',1500,2,NULL,1),(38,24,'MK002',2000,6,NULL,
1),(48,25,'MKN00',1500,2,NULL,1),(49,25,'MKN00',1500,1,NULL,1),(50,
25,'MK001',2000,1,NULL,1),(51,25,'MK002',2000,1,NULL,1),(52,25,'MK0
03',2000,1,NULL,1),(56,26,'MKN00',1500,1,NULL,1),(57,27,'MK004',200
0,2,NULL,1),(58,28,'MKN00',1500,7,NULL,1),(59,29,'MK001',2000,2,NU
LL,1),(60,30,'MK005',3500,1,NULL,1),(61,31,'MK004',2000,2,4000,1),(62,
32,'MK005',3500,1,3500,1),(63,33,'MK004',2000,2,4000,1),(64,34,'MK004'
,2000,2,4000,1),(65,35,'MKN00',NULL,3,4000,1),(66,35,'MK007',3500,2,7
000,1),(67,36,'MK007',3500,1,3500,1),(68,37,'MK001',1500,2,3000,1),(70,3
8,'MK002',2000,2,4000,1);

/*Table structure for table `transaksi` */

DROP TABLE IF EXISTS `transaksi`;

CREATE TABLE `transaksi` (
  `id_transaksi` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `total` double unsigned DEFAULT NULL,
  `uang_bayar` double unsigned DEFAULT NULL,
  `kembalian` double unsigned DEFAULT '0',
  `id_pegawai` char(5) DEFAULT NULL,
  `stat` int(11) DEFAULT '0',
```

```
  `tanggal_transaksi` datetime DEFAULT NULL,
  PRIMARY KEY (`id_transaksi`),
  KEY `id_pegawai` (`id_pegawai`),
  CONSTRAINT `transaksi_ibfk_1` FOREIGN KEY (`id_pegawai`) REFERENCES `data_pegawai` (`id_pegawai`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB AUTO_INCREMENT=122 DEFAULT CHARSET=latin1;

/*Data for the table `transaksi` */

insert into `transaksi`(`id_transaksi`,`total`,`uang_bayar`,`kembalian`,`id_pegawai`,`stat`,`tanggal_transaksi`) values (11,31000,50000,19000,'PG001',1,'2013-11-28 00:00:00'),(12,8000,10000,2000,'PG001',1,'2013-11-28 00:00:00'),(13,1500,2000,500,'PG001',1,'2013-11-28 00:00:00'),(14,1500,2000,500,'PG001',1,'2013-11-28 00:00:00'),(15,3000,4000,1000,'PG001',1,'2013-11-28 00:00:00'),(16,2000,2000,0,'PG001',1,'2013-11-28 00:00:00'),(17,2000,2000,0,'PG001',1,'2013-11-28 00:00:00'),(18,4000,5000,1000,'PG001',1,'2013-11-28 00:00:00'),(19,2000,2000,0,'PG001',1,'2013-11-28 00:00:00'),(20,1500,2000,500,'PG001',1,'2013-11-28 00:00:00'),(21,3000,20000,17000,'PG001',1,'2013-11-28
```

00:00:00'),(22,3500,4000,500,'PG001',1,'2013-11-28

00:00:00'),(23,3000,4000,1000,'PG001',1,'2013-11-28

00:00:00'),(24,12000,30000,18000,'PG001',1,'2013-12-06

00:00:00'),(25,6000,8000,2000,'PG001',1,'2013-12-21

00:00:00'),(26,1500,2000,500,'PG001',1,'2013-12-21

00:00:00'),(27,4000,5000,1000,'PG001',1,'2013-12-21

00:00:00'),(28,10500,50000,39500,'PG001',1,'2013-12-21

00:00:00'),(29,4000,5000,1000,'PG001',1,'2013-12-26

00:00:00'),(30,3500,5000,1500,'PG001',1,'2013-12-27

00:00:00'),(31,9000,1000,0,'PG001',1,'2013-12-30

08:35:50'),(32,7500,10000,2500,'PG001',1,'2013-12-30

08:39:12'),(33,4000,5000,1000,'PG001',1,'2014-01-02

07:09:49'),(34,4000,10000,6000,'PG001',1,'2014-01-02

07:10:25'),(35,7000,10000,3000,'PG001',1,'2014-01-02

07:15:35'),(36,3500,5000,1500,'PG001',1,'2014-01-02

07:17:05'),(37,3000,4000,1000,'PG001',1,'2014-01-18

13:41:45'),(38,4000,4000,0,'PG001',1,'2014-01-18

14:16:37'),(39,0,0,0,NULL,0,NULL),(40,0,0,0,NULL,0,NULL),(41,0,0,0,N

ULL,0,NULL),(42,0,0,0,NULL,0,NULL),(43,0,0,0,NULL,0,NULL),(44,0,0

,0,NULL,0,NULL),(45,0,0,0,NULL,0,NULL),(46,0,0,0,NULL,0,NULL),(4

7,0,0,0,NULL,0,NULL),(48,0,0,0,NULL,0,NULL),(49,0,0,0,NULL,0,NUL

L),(50,0,0,0,NULL,0,NULL),(51,0,0,0,NULL,0,NULL),(52,0,0,0,NULL,0,

NULL),(53,0,0,0,NULL,0,NULL),(54,0,0,0,NULL,0,NULL),(55,0,0,0,NUL

L,0,NULL),(56,0,0,0,NULL,0,NULL),(57,0,0,0,NULL,0,NULL),(58,0,0,0,

NULL,0,NULL),(59,0,0,0,NULL,0,NULL),(60,0,0,0,NULL,0,NULL),(61,0
,0,0,NULL,0,NULL),(62,0,0,0,NULL,0,NULL),(63,0,0,0,NULL,0,NULL),(
64,0,0,0,NULL,0,NULL),(65,0,0,0,NULL,0,NULL),(66,0,0,0,NULL,0,NU
LL),(67,0,0,0,NULL,0,NULL),(68,0,0,0,NULL,0,NULL),(69,0,0,0,NULL,0
,NULL),(70,0,0,0,NULL,0,NULL),(71,0,0,0,NULL,0,NULL),(72,0,0,0,NU
LL,0,NULL),(73,0,0,0,NULL,0,NULL),(74,0,0,0,NULL,0,NULL),(75,0,0,0
,NULL,0,NULL),(76,0,0,0,NULL,0,NULL),(77,0,0,0,NULL,0,NULL),(78,
0,0,0,NULL,0,NULL),(79,0,0,0,NULL,0,NULL),(80,0,0,0,NULL,0,NULL),
(81,0,0,0,NULL,0,NULL),(82,0,0,0,NULL,0,NULL),(83,0,0,0,NULL,0,NU
LL),(84,0,0,0,NULL,0,NULL),(85,0,0,0,NULL,0,NULL),(86,0,0,0,NULL,0
,NULL),(87,0,0,0,NULL,0,NULL),(88,0,0,0,NULL,0,NULL),(89,0,0,0,NU
LL,0,NULL),(90,0,0,0,NULL,0,NULL),(91,0,0,0,NULL,0,NULL),(92,0,0,0
,NULL,0,NULL),(93,0,0,0,NULL,0,NULL),(94,0,0,0,NULL,0,NULL),(95,
0,0,0,NULL,0,NULL),(96,0,0,0,NULL,0,NULL),(97,0,0,0,NULL,0,NULL),
(98,0,0,0,NULL,0,NULL),(99,0,0,0,NULL,0,NULL),(100,0,0,0,NULL,0,N
ULL),(101,0,0,0,NULL,0,NULL),(102,0,0,0,NULL,0,NULL),(103,0,0,0,N
ULL,0,NULL),(104,0,0,0,NULL,0,NULL),(105,0,0,0,NULL,0,NULL),(106
,0,0,0,NULL,0,NULL),(107,0,0,0,NULL,0,NULL),(108,0,0,0,NULL,0,NU
LL),(109,0,0,0,NULL,0,NULL),(110,0,0,0,NULL,0,NULL),(111,0,0,0,NUL
L,0,NULL),(112,0,0,0,NULL,0,NULL),(113,0,0,0,NULL,0,NULL),(114,0,0
,0,NULL,0,NULL),(115,0,0,0,NULL,0,NULL),(116,0,0,0,NULL,0,NULL),
(117,0,0,0,NULL,0,NULL),(118,0,0,0,NULL,0,NULL),(119,0,0,0,NULL,0,
NULL),(120,0,0,0,NULL,0,NULL),(121,0,0,0,NULL,0,NULL);

```
/* Trigger structure for table `detail_transaksi` */

DELIMITER $$

/*!50003 DROP TRIGGER*//*!50032 IF EXISTS */ /*!50003
`update_stock` */$$

/*!50003 CREATE */ /*!50017 DEFINER = 'root'@'localhost' */ /*!50003
TRIGGER `update_stock` AFTER INSERT ON `detail_transaksi` FOR
EACH ROW BEGIN
    UPDATE data_barang SET data_barang.`stock` = `stock` - NEW.jumlah
    WHERE Id_Barang = NEW.id_barang;
  END */$$


DELIMITER ;

/*Table structure for table `sock_baru` */

DROP TABLE IF EXISTS `sock_baru`;

/*!50001 DROP VIEW IF EXISTS `sock_baru` */;
/*!50001 DROP TABLE IF EXISTS `sock_baru` */;
```

```
/*!50001 CREATE TABLE  `sock_baru`(
 `id_barang` char(5) ,
 `nama_barang` varchar(50) ,
 `stock` double unsigned ,
 `jumlah` double unsigned ,
 `Stock Baru` double
)*/;

/*Table structure for table `stk` */

DROP TABLE IF EXISTS `stk`;

/*!50001 DROP VIEW IF EXISTS `stk` */;
/*!50001 DROP TABLE IF EXISTS `stk` */;

/*!50001 CREATE TABLE  `stk`(
 `stock` double unsigned ,
 `jumlah` double unsigned ,
 `Stock Sekarang` double
)*/;

/*View structure for view sock_baru */

/*!50001 DROP TABLE IF EXISTS `sock_baru` */;
```

```
/*!50001 DROP VIEW IF EXISTS `sock_baru` */;

/*!50001 CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `sock_baru` AS select `data_barang`.`id_barang` AS `id_barang`,`data_barang`.`nama_barang` AS `nama_barang`,`data_barang`.`stock` AS `stock`,`detail_transaksi`.`jumlah` AS `jumlah`,(`data_barang`.`stock` - `detail_transaksi`.`jumlah`) AS `Stock Baru` from (`data_barang` join `detail_transaksi`) where (`data_barang`.`id_barang` = `detail_transaksi`.`id_barang`) */;

/*View structure for view stk */

/*!50001 DROP TABLE IF EXISTS `stk` */;
/*!50001 DROP VIEW IF EXISTS `stk` */;

/*!50001 CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `stk` AS select `data_barang`.`stock` AS `stock`,`detail_transaksi`.`jumlah` AS `jumlah`,(`data_barang`.`stock` - `detail_transaksi`.`jumlah`) AS `Stock Sekarang` from (`data_barang` join `detail_transaksi`) */;

/*!40101 SET SQL_MODE=@OLD_SQL_MODE */;
```

```
/*!40014 SET
FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS */;
/*!40014 SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS */;
/*!40111 SET SQL_NOTES=@OLD_SQL_NOTES */;
```

4.3.5 Implementasi Antar Muka

Berikut ini akan dijelaskan tampilan antarmuka (interface) pada aplikasi yang dirancang dari setiap halaman yang dibuat dan dibedakan berdasarkan pegawai dan admin.

4.3.5.1 Implementasi Login

Dalam melakukan login, input-an password dan id pegawai harus benar, apabila salah maka admin tidak bisa mengakses aplikasi penjualan ini.

Tabel 4.5 Implementasi Login

Sub Menu	Deskripsi	Nama File
Login	Untuk login bagian kasir, gudang, dan pemilik	Login.cs
Batal	Untuk keluar dari form login	

4.3.5.2 Implementasi Menu Utama

Tabel 4.6 Implementasi Menu Utama

Sub Menu	Deskripsi	Nama File

Menu Utama	Menampilkan menu utama.	Menu_utama.cs
File	Menampilkan sub menu logout,about,exit	
Transaksi	Untuk transaksi proses pembayaran.	Transaksi.cs
Detail Transaksi	Menampilkan detail transaksi.	Detail_transaksi.cs
Catatan Transaksi	Menampilkan catatan transaksi	Data_transaksi.cs
Data Barang	Menampilkan dan mengedit data barang	Data_barang.cs
Data Pegawai	Menampilkan dan mengedit data pegawai	Data_pegawai.cs
Data Pemasok	Menampilkan dan mengedit data pemasok	Data_pemasok.cs
Cetak Laporan	Menampilkan sub menu Cetak Laporan Transaksi dan Cetak Laporan Data Barang	

4.3.5.3 Implementasi Halaman Berdasarkan Sub Menu Akses

Implementasi halaman berdasarkan sub menu akses dibuat dalam bentuk file program yang dapat dilihat pada tabel berikut ini :

Tabel 4.7 Implementasi Halaman Berdasarkan File

Sub Menu	Deskripsi	Nama File
Logout	Menangani logout pegawai	-
About	File program untuk menampilkan informasi dari program yang dibuat	About.cs
Exit	Perintah Keluar Program	-

Tabel 4.8 Implementasi Halaman Berdasarkan Sub Menu Cetak Laporan

Sub Menu	Deskripsi	Nama File
Cetak Laporan Transaksi	File progran untuk mencetak Laporan Transaksi	Rpt_transaksi_penjualan.cs
Cetak Laporan Data Barang	File program untuk mencetak Laporan Data Barang	Rpt_data_barang.cs

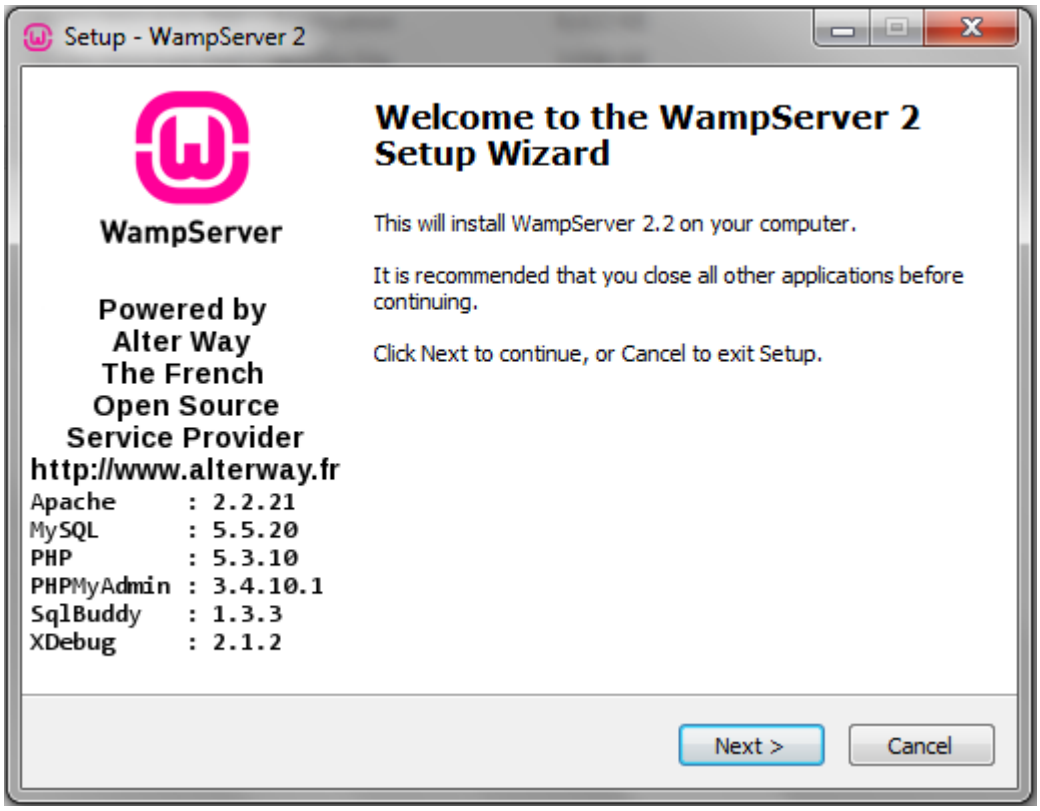
4.3.6 Instalasi WampServer

Pertama yang harus dilakukan adalah membuka file instalasi wampserver2.2d-x32 yang berekstensi.exe kemudian klik dua kali untuk menginstalnya.



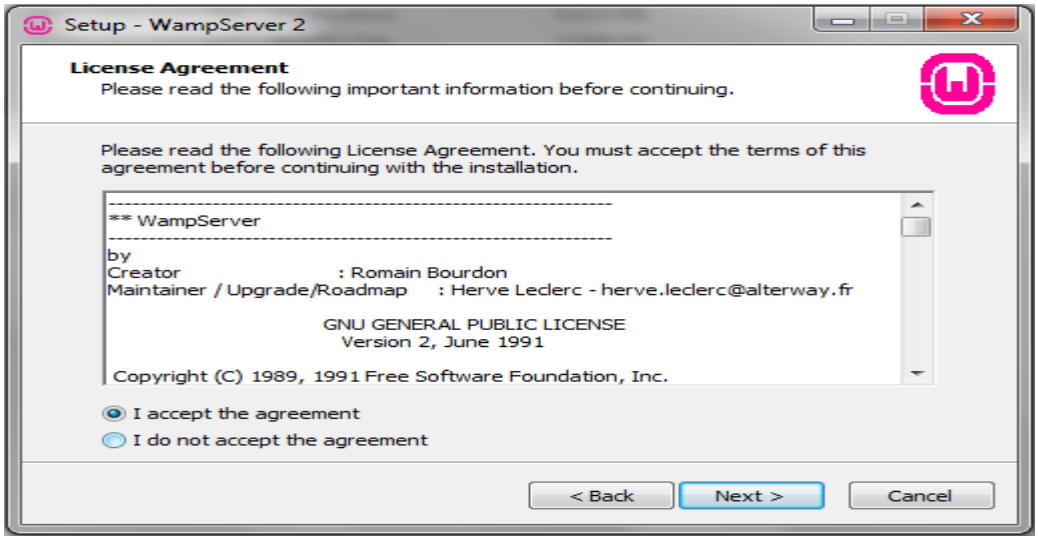
Gambar 4.22 Icon Instalasi wampserver

1. Setelah klik 2 kali pada icon installer wampserver akan muncul pop up window seperti gambar dibawah ini kemudian klik next.



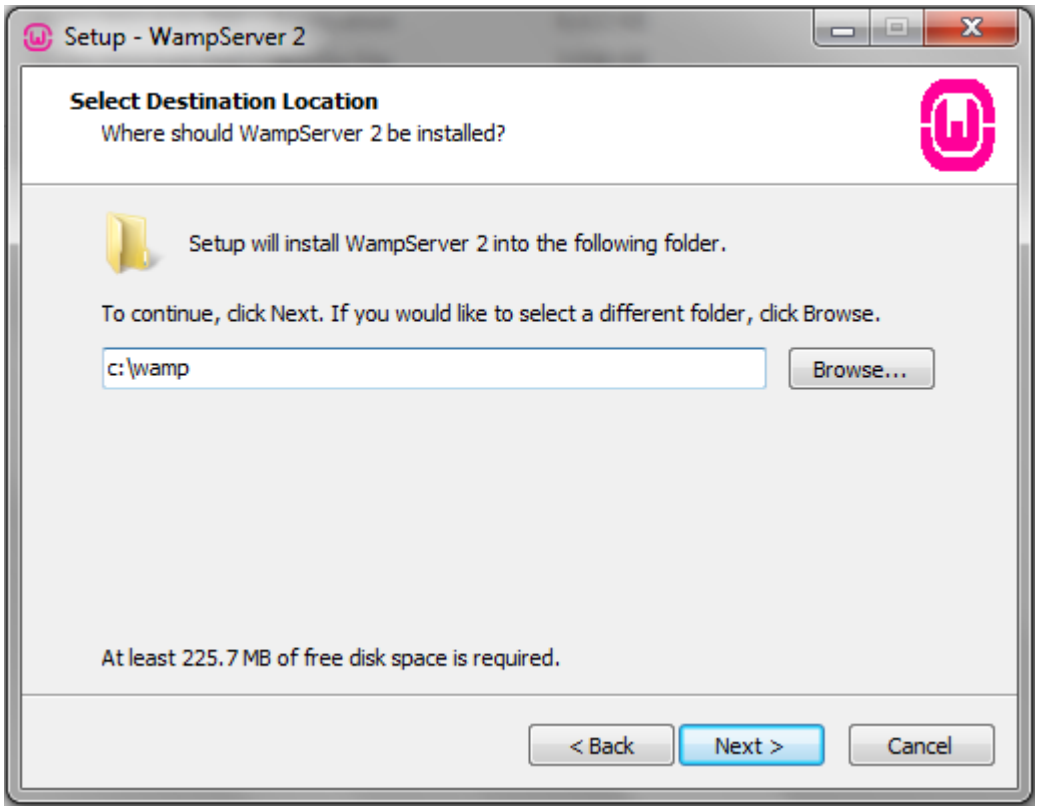
Gambar 4.23 Tampilan Awal Instalasi WampServer

2. Pilih "*I accept the agreement*" dan selanjutnya klik next.



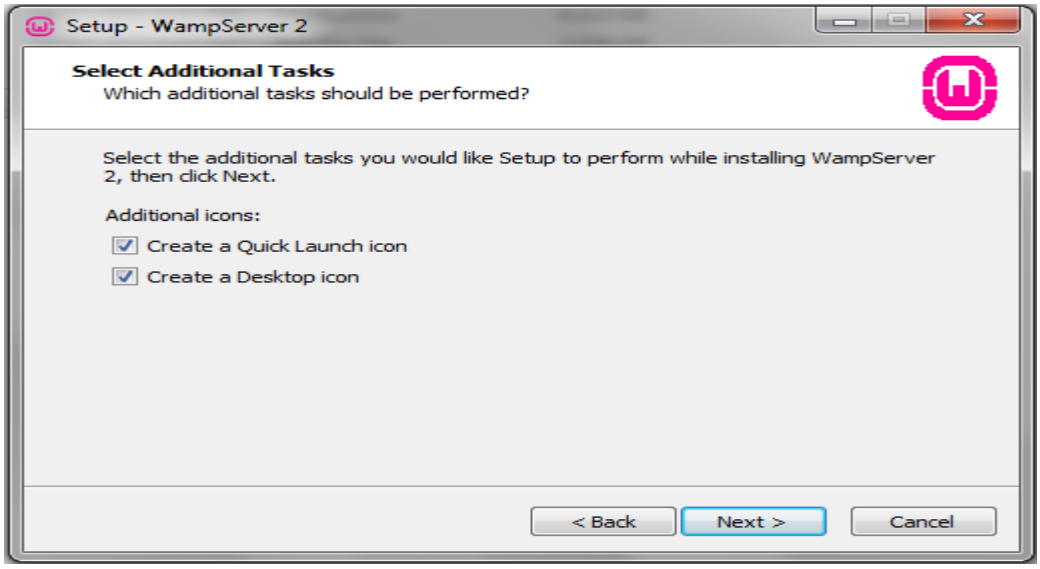
Gambar 4.24 Tampilan Instalasi WampServer Persetujuan

3. Klik next tanpa perlu merubah nama file dan lokasi instal.



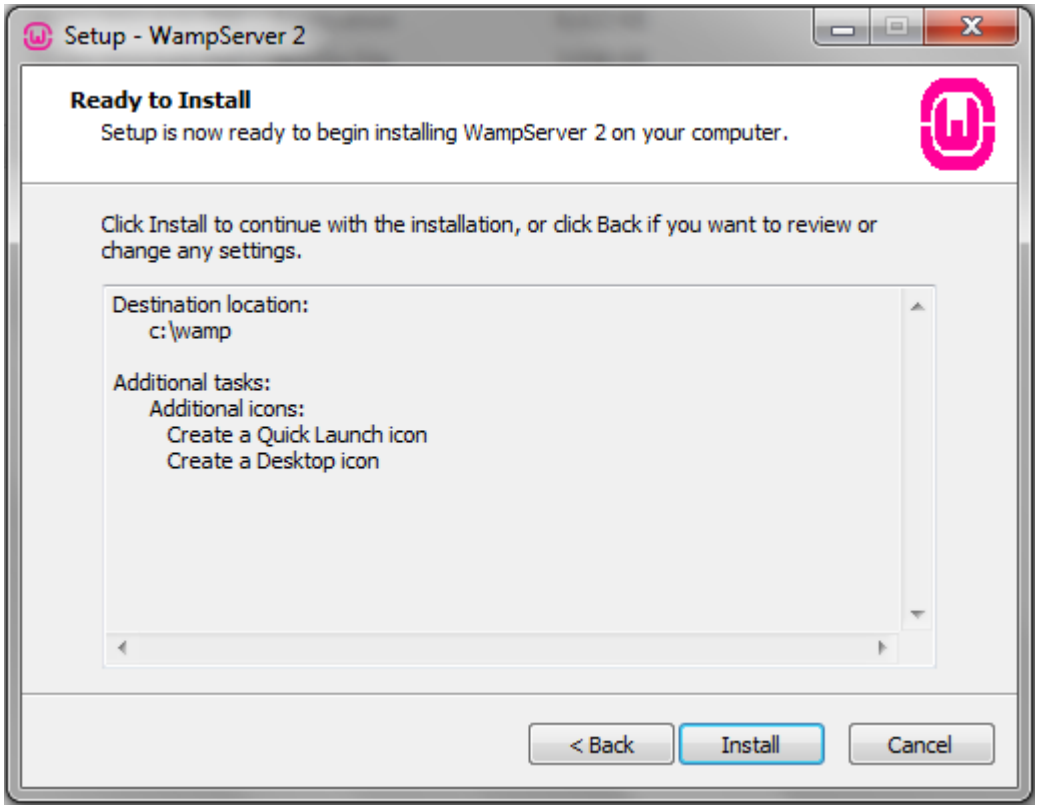
Gambar 4.25 Tampilan Instalasi WampServer Lokasi

4. Centang pada kedua ikon jika ingin menampilkan shortcut dan ikon wamp pada desktop dan task bar komputer, lalu klik next.



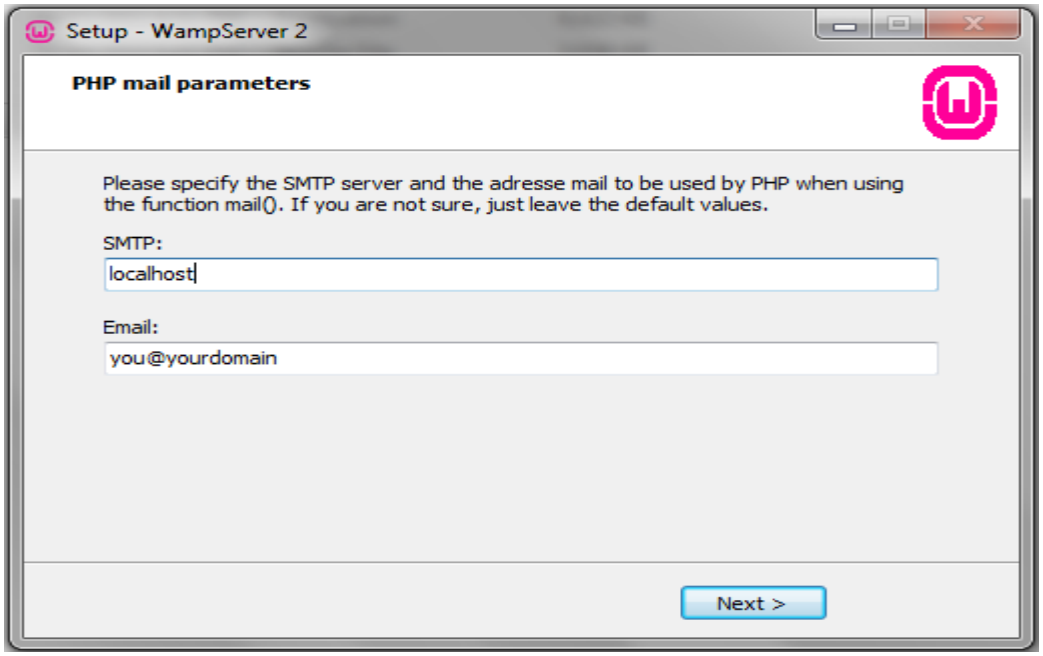
Gambar 4.26 Tampilan Instalasi WampServer Ikon

5. Klik Install dan tunggu beberapa saat selama proses install.



Gambar 4.27 Tampilan Instalasi WampServer Instal

6. Biarkan tetap pada default, lanjutkan dengan klik next.



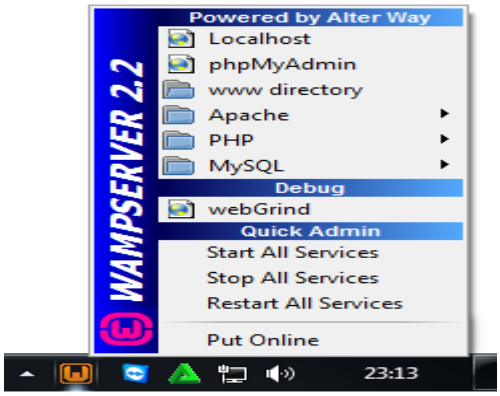
Gambar 4.28 Tampilan Instalasi WampServer Alamat

7. Proses Instal WampServer ke komputer telah selesai dengan baik, klik finish.



Gambar 4.29 Tampilan Akhir Instalasi WampServer

8. Berikut adalah tampilan WampServer setelah selesai diinstal.



Gambar 4.30 Tampilan Aplikasi WampServer

Jalankan Apache dan MySQL dengan menekan tombol start service ketika aplikasi dijalankan ikon WampServer akan berwarna hijau.

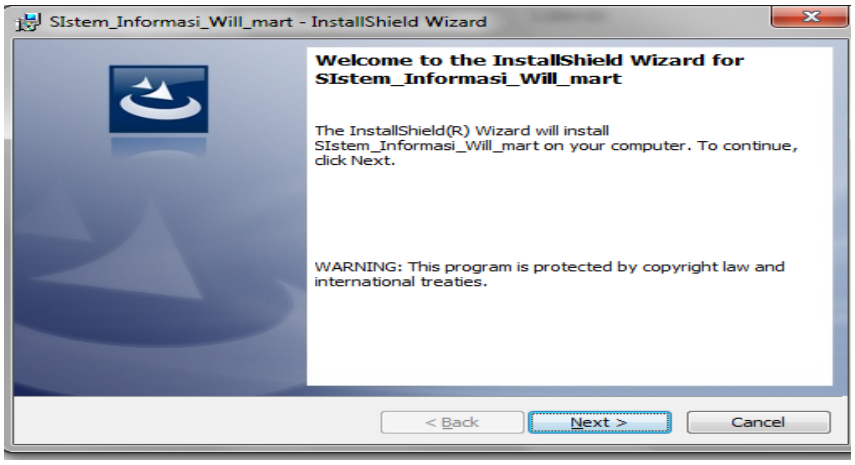
4.3.6.1 Instalasi Program

Untuk menginstal program sistem aplikasi bisnis pada toko willmart pertama klik dua kali pada ikon setup program.



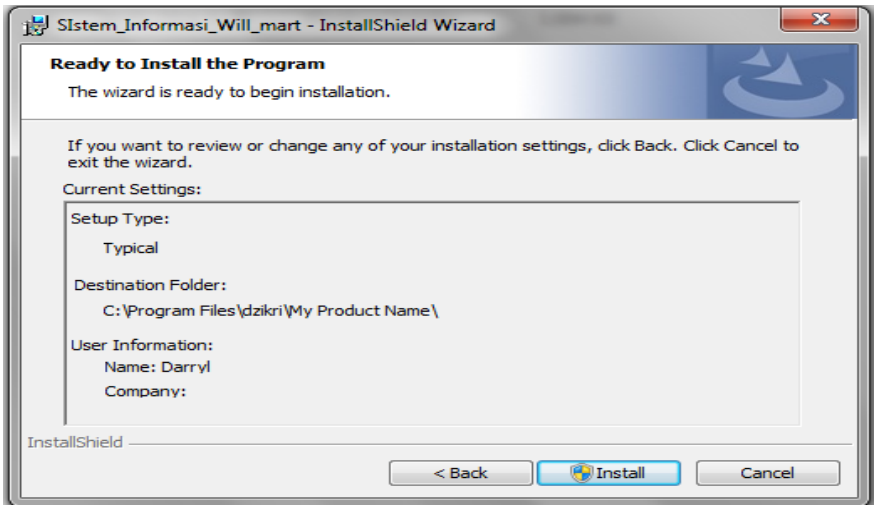
Gambar 4.31 Icon Setup instalasi program

Kemudian akan muncul tampilan awal instalasi, lalu klik next.



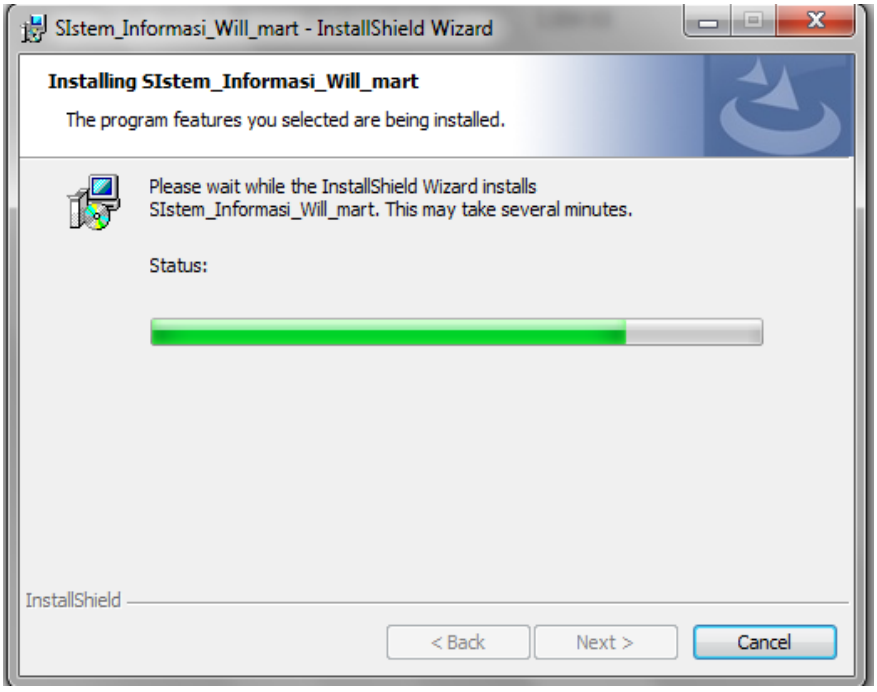
Gambar 4.32 Tampilan awal Instalasi Program

Klik Install



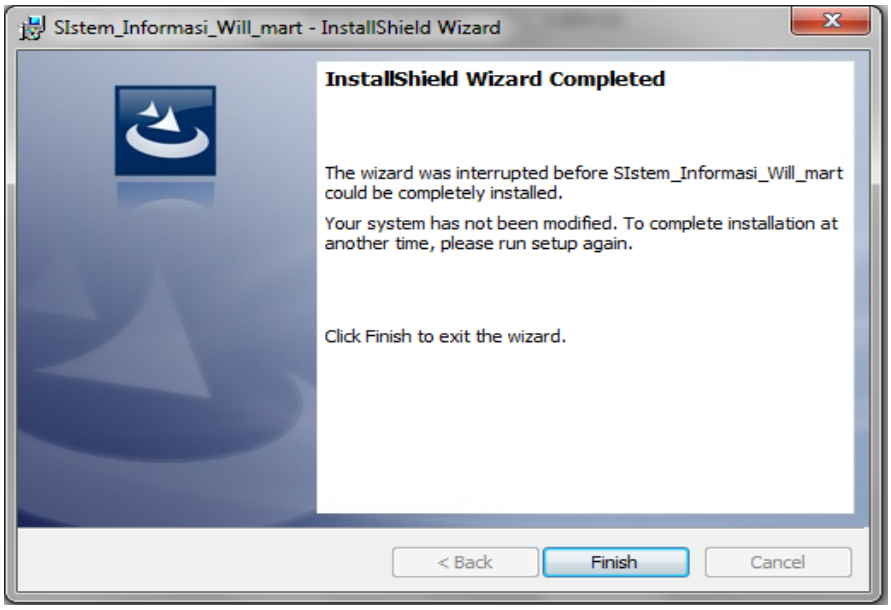
Gambar 4.33 Tampilan Instalasi Program

Sedang dalam proses tunggu beberapa saat.



Gambar 4.34 Tampilan proses Instalasi Program

Proses instalasi selesai kemudian klik finish.



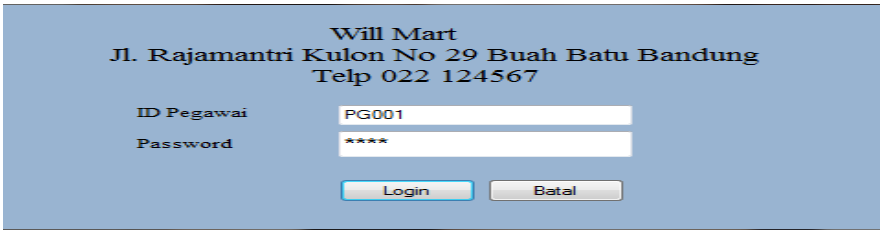
Gambar 4.35 Tampilan proses Instalasi Program selesai.

4.3.7 Penggunaan Program

Penggunaan Program bertujuan untuk menerangkan secara singkat bagaimana program dioperasikan. Adapun tampilan dari program tersebut adalah sebagai berikut:

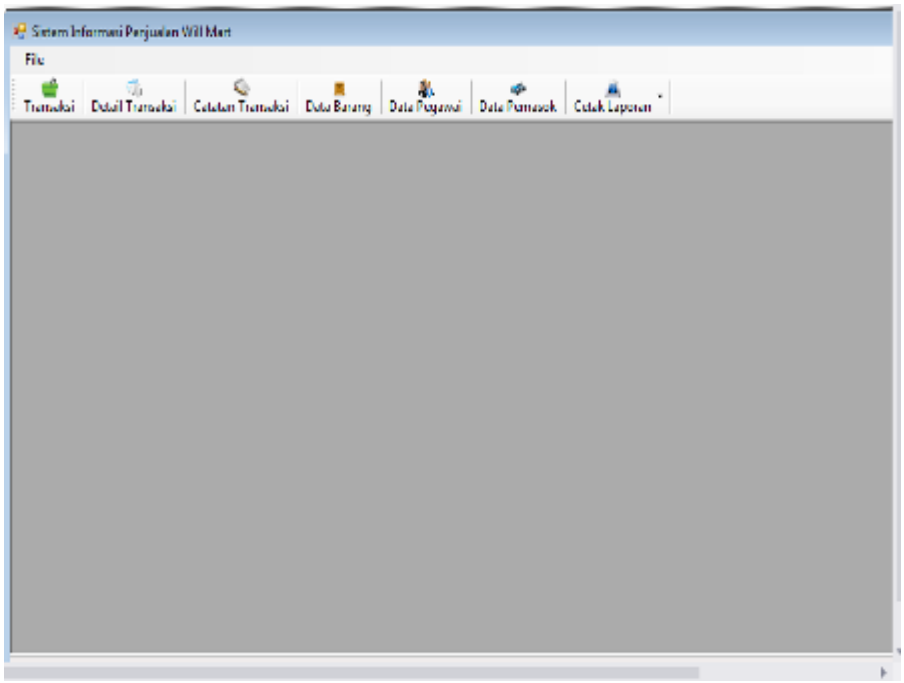
1. Tampilan Form Login

   Form login ini berfungsi sebagai keamanan sistem. Untuk melakukan login pengguna harus isi id pegawai dan password kemudian klik tombol login. Jika id pegawai dan password tidak cocok maka tidak bisa diakses.



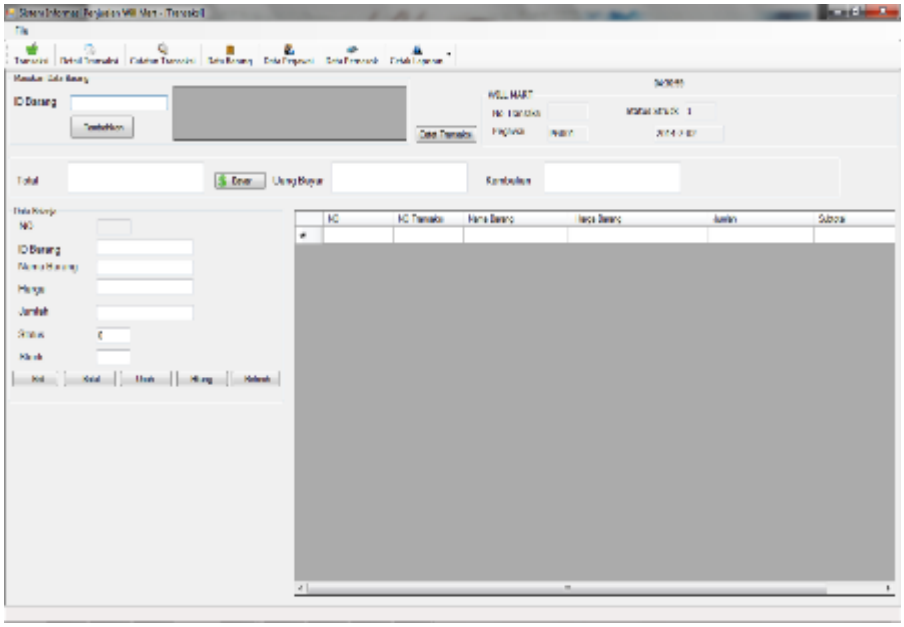
Gambar 4.36 Tampilan Login.

2. Tampilan Menu Utama



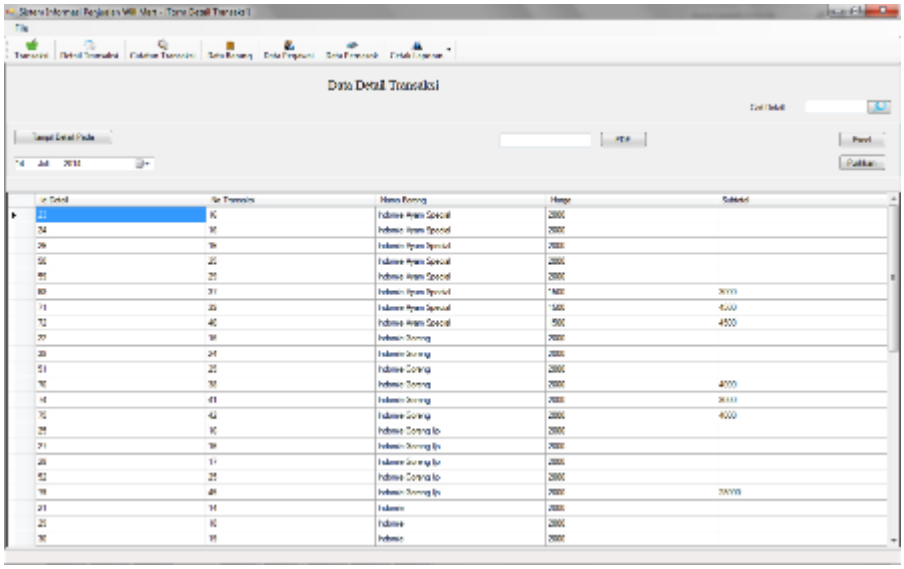
Gambar 4.37 Tampilan Menu Utama

3. Tampilan Transaksi



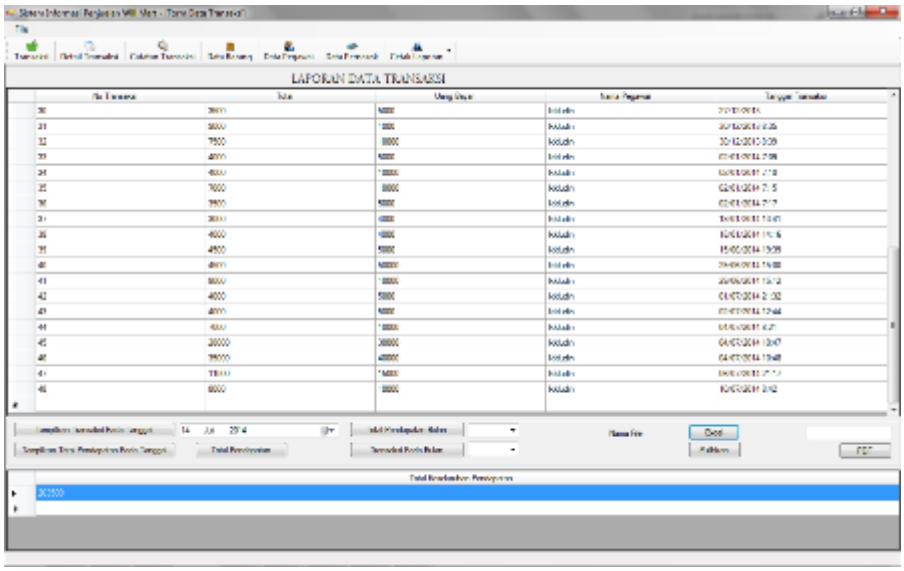
Gambar 4.38 Tampilan Transaksi

4. Tampilan Detail Transaksi



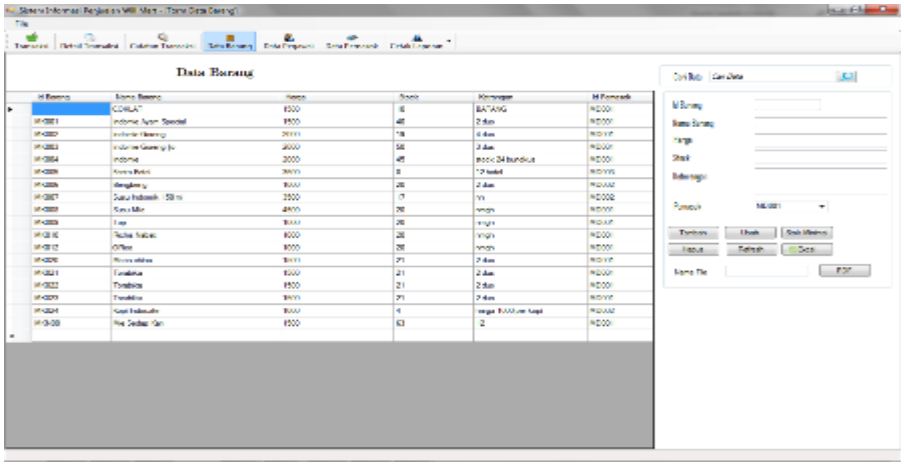
Gambar 4.39 Tampilan Detail Transaksi

5. Tampilan Catatan Transaksi



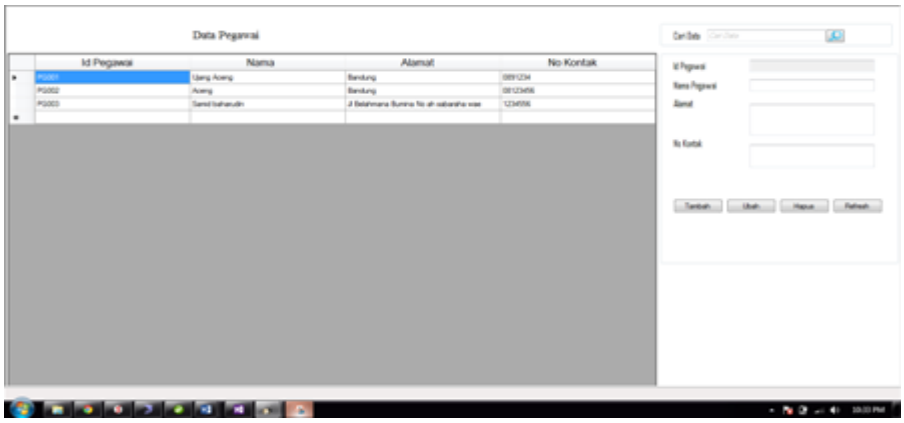
Gambar 4.40 Tampilan Catatan Transaksi

6. Tampilan Data Barang



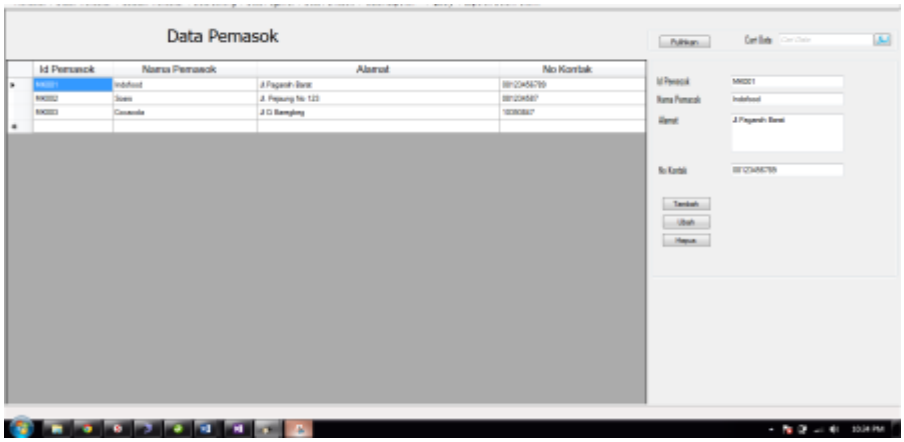
Gambar 4.41 Tampilan Data Barang

7. Tampilan Data Pegawai



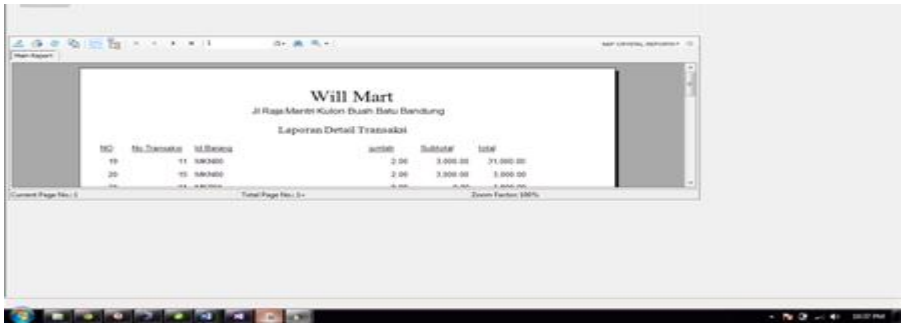
Gambar 4.42 Tampilan Data Pegawai

8. Tampilan Data Pemasok



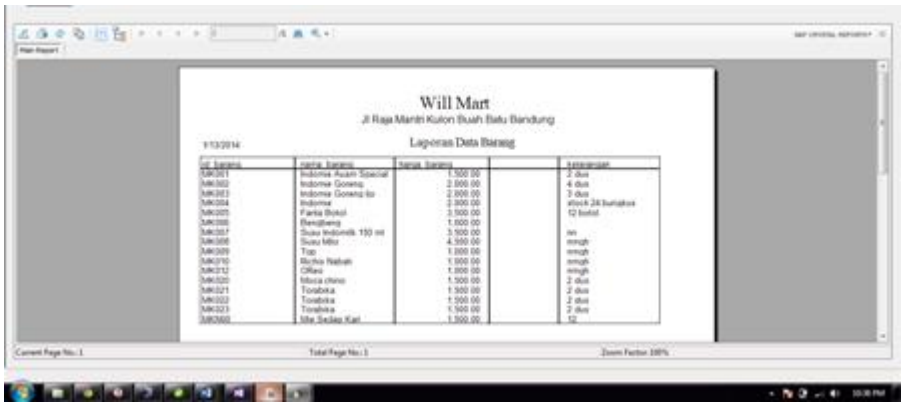
Gambar 4.43 Tampilan Data Pemasok

9. Tampilan Laporan Catatan Transaksi



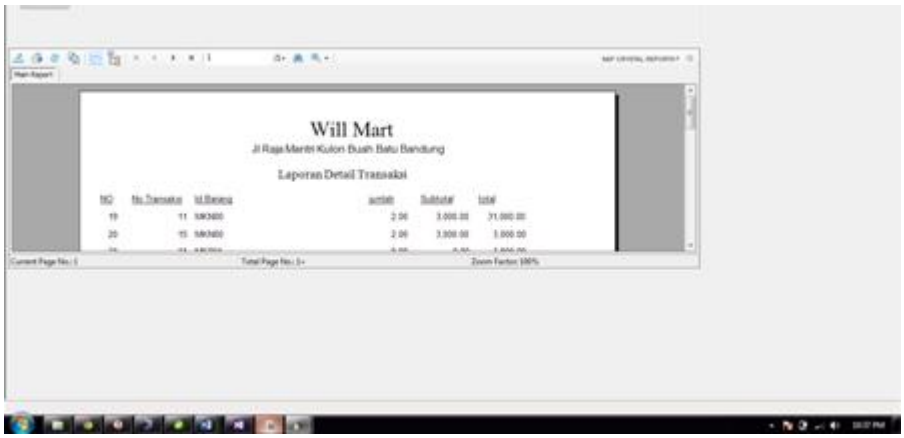
Gambar 4.44 Tampilan Laporan Catatan Transaksi

10. Tampilan Cetak Laporan Data Barang



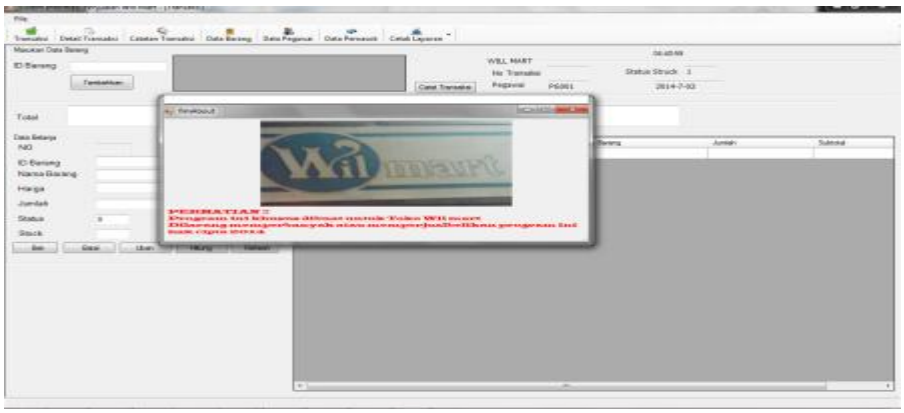
Gambar 4.45 Tampilan Cetak Laporan Data Barang

11. Tampilan Cetak Laporan Transaksi



Gambar 4.46 Tampilan Cetak Laporan Transaksi

12. Tampilan About



Gambar 4.47 Tampilan About

4.4 Pengujian

Bagian yang penting dalam siklus pembangunan perangkat lunak adalah pengujian. Pengujian dilakukan untuk menjamin kualitas dan mengetahui kelemahan dari perangkat lunak tersebut. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibuat adalah perangkat lunak yang memiliki kualitas yang baik, yaitu mampu menerjemahkan setiap proses dari implementasi sistem, dimulai dari spesifikasi, analisis, perancangan, dan pengkodean dari perangkat lunak itu sendiri.

Pengujian perangkat lunak ini menggunakan metode pengujian black box. Dimana pengujian black box berfokus pada fungsional perangkat lunak yang dibuat.

4.4.1 Rencana Pengujian

Proses rencana pengujian meliputi input/output. Proses pengujiannya adalah mencoba program dengan memasukan data kedalam form-form yang telah disediakan. Pada tahap ini merupakan kelanjutan dari tahap implementasi yaitu melakukan pengujian-pengujian terhadap aplikasi yang telah dibangun.

Adapun hal-hal yang akan diuji melalui teknik pengujian Black Box adalah sebagai berikut :

Tabel 4.9 Rencana Pengujian

Kelas Uji	**Butir Uji**	**Tingkat Pengujian**	**Jenis Pengujian**
Login	Mengecek Login	Sistem	Black Box
Pengisian Data	Mengecek Data Transaksi	Modul	
	Mengecek Data Barang		
	Mengecek Data Pegawai		
	Mengecek Data Pemasok		

4.4.2 Kasus dan Hasil Pengujian

Berdasarkan rencana pengujian yang telah diujikan di *Toko Wilmart*, maka dapat dilakukan pengujian sebagai berikut :

1. Pengecekan Login

   Berikut adalah kasus dan hasil pengujian login :

   Tabel 4.10 Kasus dan Hasil Pengujian Pengecekan Login

Kasus dan Hasil Pengecekan Data (Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Id Pegawai : PG001 Password : abcd Id Pegawai : PG002 Password : 12345	Tercantum pada textbox Id Pegawai dan textbox Password.	Mengisi Login seperti yang diharapkan.	[X] Diterima [ ] Ditolak
Klik Tombol Login	Menampilkan login berhasil dan dapat mengakses sub menu yang ada di menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Id Pegawai : PG001 Password : 12345 Id Pegawai : PG002 Password : abcd	Tercantum pada textbox Usename dan textboxt Password.	Login Gagal	[ ] Diterima [X] Ditolak
Klik Tombol Login	Menampilkan pesan	Tombol dapat berfungsi sesuai	[ ] Diterima [X] Ditolak

	“Maaf Username atau Password salah” dan dapat mengakses sub menu yang ada di menu utama	dengan yang diharapkan	

4.4.3 Kesimpulan Hasil Pengujian

Dengan hasil pengujian yang telah dilakukan berdasarkan uji kasus di atas, maka dapat diambil kesimpulan bahwa perangkat lunak yang sebelumnya diuji tidak diketahui terdapat beberapa kesalahan dan setelah diuji dapat diketahui tingkatan kesalahannya dan dapat diperbaiki dari kesalahan sintak dan secara fungsional sudah berjalan dengan baik. Hal ini didukung dengan pengolahan data yang sederhana sehingga pengguna tidak mengalami kesulitan.