BAB V

IMPLEMENTASI DAN PENGUJIAN PROGRAM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem yang baru. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman. Setelah implementasi maka dilakukan pengujian terhadap sistem yang baru dan akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk pengembangan sistem selanjutnya.

5.1. Implementasi

Implementasi adalah suatu bentuk kegiatan yang merupakan rangkaian lanjutan dari kegiatan perancangan perangkat lunak. Implementasi dimaksudkan sebagai usaha untuk mewujudkan hasil dari perancangan perangkat lunak. Hasil ini berarti hasil dari kegiatan rangkaian implementasi perancangan perangkat lunak adalah proses aplikasi yang telah dibuat dan sudah dapat digunakan serta berfungsi dengan baik.

Jadi secara garis besar implementasi merupakan proses penerapan rancangan program yang telah dibuat pada bab sebelumnya atau aplikasi dalam melaksanakan sisten informasi pemograman yang dibuat.

5.1.1. Batasan Implementasi (*Optional*)

Dalam mengimplementasikan perangkat lunak *Website* Penjualan di Distro Snooby ini dibatasi oleh penggunaan *Software* (perangkat lunak) dan *Hardware* (perangkat keras) minimum.

5.1.2. Implementasi Perangkat Lunak

Untuk implementasi perangkat lunak dibagi menjadi 3 sudut pandang, yaitu :

1. Implementasi Perangkat Lunak untuk pengembangan, antara lain :
    a. Apache (2.2.4), sebagai *web server*.
    b. PHP (5.2.3), sebagai bahasa pemograman.
    c. MySQL(5.0.37), untuk pengembangan dalam pembuatan *database*.
    d. Adobe Dreamweaver CS4, sebagai editor dalam pengembangan desain *web*.
2. Implementasi Perangkat Lunak untuk *admin*, antara lain :
    a. Apache, sebagai *web server* dalam keadaaan *offline.*
    b. *Web hosting* dan *domain*, dalam keadaan *online*.
    c. *Browser*, hasil terbaik menggunakan Google Chrome.
    d. PDF Reader.

5.1.3. Implementasi Perangkat Keras

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

1. Untuk *admin*.
    a. *Processor* dengan kecepatan minimal 2.0 GHz.
    b. *RAM* minimal 1 Gb.
    c. *Hard Drive*, minimal 160 Gb sebagai media penyimpanan data.
    d. *Mouse, Keyboard*, dan *Monitor* sebagai peralatan antarmuka.
    e. *Modem*, sebagai media untuk koneksi ke internet.
2. Untuk *user*.
    a. *Processor* dengan kecepatan minimal 1.2 GHz.
    b. *RAM* minimal 1 Gb.
    c. *Hard Drive*, minimal 80 Gb sebagai media penyimpanan data.
    d. *Mouse, Keybord*, dan *Monitor* sebagai peralatan antarmuka.
    e. Modem, sebagai media untuk koneksi ke internet.

5.1.4. Implementasi Basis Data

-- phpMyAdmin SQL Dump

-- version 2.10.1

-- http://www.phpmyadmin.net

--

```
-- Host: localhost
-- Generation Time: Jun 18, 2012 at 07:05 PM
-- Server version: 5.0.41
-- PHP Version: 5.2.3
SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
--
-- Database: `anggi`
--
-- --------------------------------------------------------
--
-- Table structure for table `katagori`
--
CREATE TABLE `katagori` (
  `id` int(11) NOT NULL auto_increment,
  `kode_katagori` varchar(10) NOT NULL,
  `nama_katagori` varchar(60) NOT NULL,
  PRIMARY KEY  (`id`)
)ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=14
;
--
-- --------------------------------------------------------
--
-- Table structure for table `konfirmasi_pembayaran`
```

```
--

CREATE TABLE `konfirmasi_pembayaran` (
  `id` int(11) NOT NULL auto_increment,
  `tgl_transfer` date NOT NULL,
  `nama_bank` varchar(30) NOT NULL,
  `nama_nasabah` varchar(50) NOT NULL,
  `nominal` int(11) default NULL,
  `tujuan_transfer` varchar(30) NOT NULL,
  `id_pemesanan` varchar(30) NOT NULL,
  `id_member` int(11) default NULL,
  `status` tinyint(1) default NULL,
  PRIMARY KEY  (`id`),
  KEY `id_pemesanan` (`id_pemesanan`),
  KEY `id_member` (`id_member`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3;

--

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

--

-- Table structure for table `kota`

--

CREATE TABLE `kota` (
  `id` int(11) NOT NULL auto_increment,
```

```
  `nama_kota` varchar(50) default NULL,
  `status` tinyint(1) default NULL,
  `id_provinsi` int(11) default NULL,
  PRIMARY KEY  (`id`),
  KEY `id_provinsi` (`id_provinsi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=7;
--
-- ------------------------------------------------------
--
-- Table structure for table `member`
--
CREATE TABLE `member` (
  `id` int(11) NOT NULL auto_increment,
  `email` varchar(50) NOT NULL,
  `password` varchar(150) NOT NULL,
  `nama_depan` varchar(100) NOT NULL,
  `nama_belakang` varchar(100) default NULL,
  `id_provinsi` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `alamat` varchar(150) NOT NULL,
  `kode_pos` varchar(7) NOT NULL,
  `telp` varchar(15) default NULL,
```

```
  `status` tinyint(1) default NULL,
  PRIMARY KEY  (`id`),
  UNIQUE KEY `email` (`email`),
  KEY `id_provinsi` (`id_provinsi`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2;
--
-- --------------------------------------------------------
--
-- Table structure for table `mitra_pengiriman`
--
CREATE TABLE `mitra_pengiriman` (
  `id` int(11) NOT NULL auto_increment,
  `kode_mitra` varchar(10) NOT NULL,
  `nama_mitra` varchar(50) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5;
--
-- --------------------------------------------------------
--
-- Table structure for table `pemesanan`
--
```

```
CREATE TABLE `pemesanan` (
  `id` varchar(30) NOT NULL,
  `id_member` int(11) NOT NULL,
  `nama_depan` varchar(50) default NULL,
  `nama_belakang` varchar(50) default NULL,
  `id_provinsi` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `alamat` varchar(150) NOT NULL,
  `kode_pos` varchar(5) NOT NULL,
  `telp` varchar(15) NOT NULL,
  `nama_depan2` varchar(50) default NULL,
  `nama_belakang2` varchar(50) default NULL,
  `id_provinsi2` int(11) NOT NULL,
  `id_kota2` int(11) NOT NULL,
  `alamat2` varchar(150) NOT NULL,
  `kode_pos2` varchar(5) NOT NULL,
  `telp2` varchar(15) NOT NULL,
  `id_mitra_kirim` int(11) NOT NULL,
  `sub_total` int(11) default NULL,
  `biaya_kirim` int(11) default NULL,
  `grand_total` int(11) default NULL,
  `jenis_pembayaran` varchar(30) NOT NULL,
```

```
  `komentar` varchar(200) default NULL,
  `date` datetime default NULL,
  `nomer_pengiriman` varchar(30) default NULL,
  `status_pengiriman` varchar(3) default NULL,
  `tgl_pengiriman` date default NULL,
  `status` varchar(3) NOT NULL,
  PRIMARY KEY  (`id`),
  KEY `id_mitra_kirim` (`id_mitra_kirim`),
  KEY `id_provinsi` (`id_provinsi`),
  KEY `id_provinsi2` (`id_provinsi2`),
  KEY `id_kota` (`id_kota`),
  KEY `id_kota2` (`id_kota2`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- --------------------------------------------------------
--
-- Table structure for table `pemesanan_det`
--
CREATE TABLE `pemesanan_det` (
  `id` int(11) NOT NULL auto_increment,
  `id_produk_det` int(11) NOT NULL,
  `harga` int(11) default NULL,
```

```
  `bnyk` int(11) NOT NULL,
  `total` int(11) default NULL,
  `disk` int(11) default NULL,
  `sub_total` int(11) default NULL,
  `status` tinyint(1) default NULL,
  `id_pemesanan` varchar(30) NOT NULL,
  PRIMARY KEY  (`id`),
  KEY `id_pemesanan` (`id_pemesanan`),
  KEY `id_produk_det` (`id_produk_det`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=9;
--
-- --------------------------------------------------------
--
-- Table structure for table `produk`
--
CREATE TABLE `produk` (
  `id` varchar(10) NOT NULL,
  `nama_produk` varchar(50) NOT NULL,
  `warna` varchar(30) NOT NULL,
  `harga` int(11) default NULL,
  `disk` int(11) default NULL,
  `stok` int(11) default NULL,
```

```
  `id_katagori` int(11) default NULL,
  `keterangan` varchar(150) default NULL,
  `tanggal` datetime default NULL,
  `image1` varchar(50) default NULL,
  `image2` varchar(50) default NULL,
  `image3` varchar(50) default NULL,
  PRIMARY KEY  (`id`),
  KEY `id_katagori` (`id_katagori`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- --------------------------------------------------------
--
-- Table structure for table `produk_det`
--
CREATE TABLE `produk_det` (
  `id` int(11) NOT NULL auto_increment,
  `size_produk` varchar(20) NOT NULL,
  `stok` int(11) default NULL,
  `berat` int(11) default NULL,
  `status` tinyint(1) default NULL,
  `id_produk` varchar(10) NOT NULL,
  PRIMARY KEY  (`id`),
```

```
  KEY `id_produk` (`id_produk`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=30 ;
--
-- ------------------------------------------------------
--
-- Table structure for table `provinsi`
--
CREATE TABLE `provinsi` (
  `id` int(11) NOT NULL auto_increment,
  `nama_provinsi` varchar(50) NOT NULL,
  `status` tinyint(1) default NULL,
  PRIMARY KEY  (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=6;
--
-- ------------------------------------------------------
--
-- Table structure for table `retur`
--
CREATE TABLE `retur` (
  `id` varchar(30) NOT NULL,
  `tgl_retur` datetime NOT NULL,
```

```
  `id_transaksi` varchar(30) NOT NULL,
  `status` varchar(3) NOT NULL,
  `status_pengiriman` varchar(3) default NULL,
  `id_mitra_kirim` int(11) default NULL,
  `tgl_pengiriman_retur` date default NULL,
  `nomer_pengiriman` varchar(30) default NULL,
  PRIMARY KEY  (`id`),
  KEY `id_transaksi` (`id_transaksi`),
  KEY `id_mitra_kirim` (`id_mitra_kirim`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- --------------------------------------------------------
--
-- Table structure for table `retur_det`
--
CREATE TABLE `retur_det` (
  `id` int(11) NOT NULL auto_increment,
  `id_pemesanan_det` int(11) NOT NULL,
  `bny` int(11) default NULL,
  `alasan` varchar(250) default NULL,
  `id_retur` varchar(30) NOT NULL,
  `status` varchar(3) NOT NULL,
```

```
  PRIMARY KEY  (`id`),
  KEY `id_pemesanan_det` (`id_pemesanan_det`),
  KEY `id_retur` (`id_retur`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1;
--
-- --------------------------------------------------------
--
-- Table structure for table `tarif_kirim`
--
CREATE TABLE `tarif_kirim` (
  `id` int(11) NOT NULL auto_increment,
  `harga_kirim` int(11) NOT NULL,
  `id_mitra` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  PRIMARY KEY  (`id`),
  KEY `id_mitra` (`id_mitra`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=13 ;
```

5.1.5. Implementasi Antar Muka dan Penggunaan Program

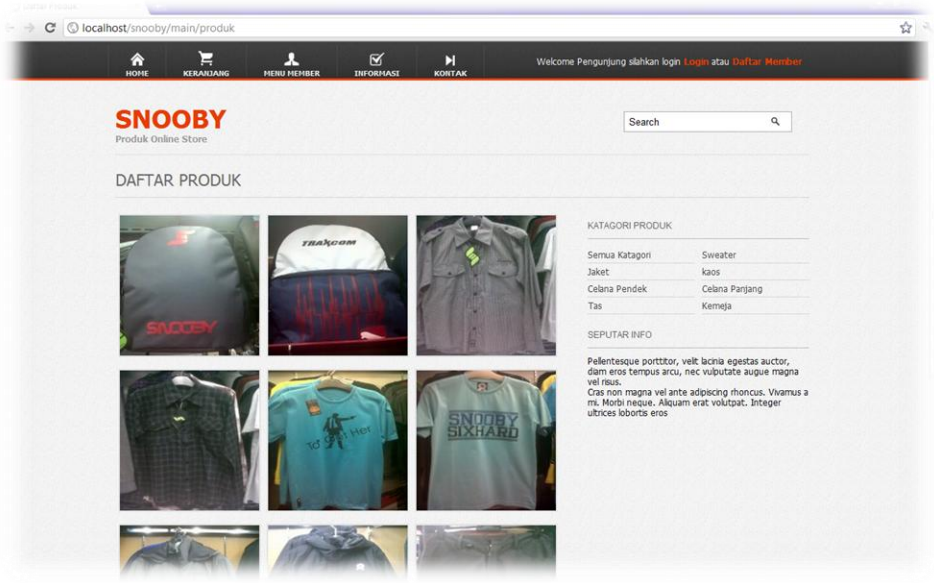
Dalam pembuatan aplikasi apapun, implementasi antar muka juga mempunyai peranan yang penting. Implementasi antar muka dapat memudahkan *user* dalam mengoperasikan aplikasi yang telah dibuat. Oleh karena itu, antar muka pada aplikasi *Website* Penjualan pada Distro Snooby dibuat menarik dan sederhana sehingga *admin* dan juga *user* dapat mengoperasikan aplikasi ini dengan mudah. Untuk lebih jelasnya akan digambarkan sebagai berikut :

1. Tampilan *User*

   Tampilan *user* merupakan tampilan yang digunakan oleh *user* untuk dapat mengakses data produk dan melakukan kegiatan transaksi pemesanan produk.

   1.1. Tampilan Utama

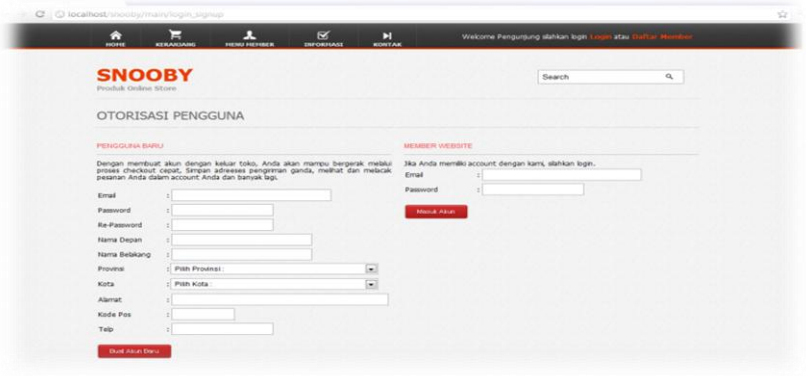
   Pada halaman *index* ini terdapat beberapa fungsi yang dapat membantu para konsumen untuk menjalankan program ini, diantaranya pada *header* terdapat *link* (Home, Keranjang, Menu Member, Informasi, Kontak).



Gambar 5.1 Menu Utama *Website* Distro Snooby

1.2. Halaman *Log in*

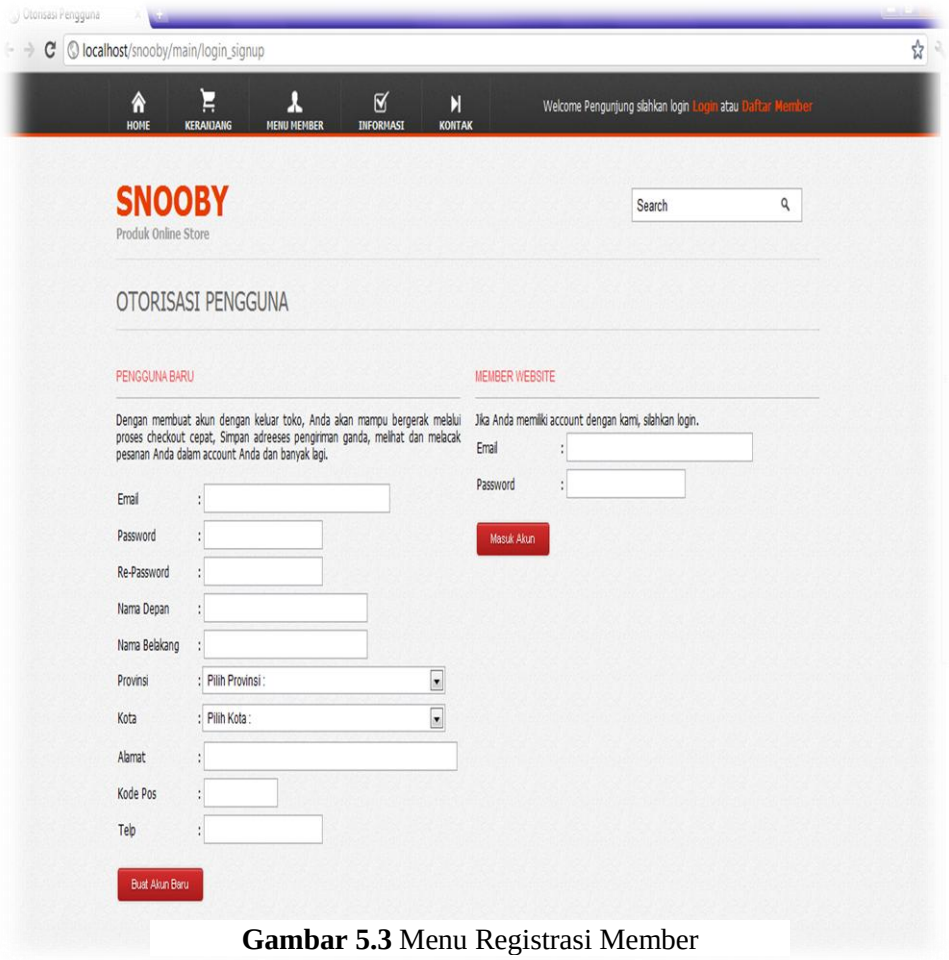
Suatu *form* dimana jika seorang pengunjung ingin melakukan pembelian, maka pengunjung diwajibkan mengisi *login user*, dan jika belum terdaftar maka pengunjung diberikan fasilitas pendaftaran yang ditujukan untuk menjadi member.



Gambar 5.2 Menu *Log in* Member

1.3. Tampilan *Form* Registrasi

Untuk melakukan transaksi, sebelumnya kita sudah terdaftar menjadi member. *Form* pendaftaran bisa dilihat pada gambar di bawah ini.

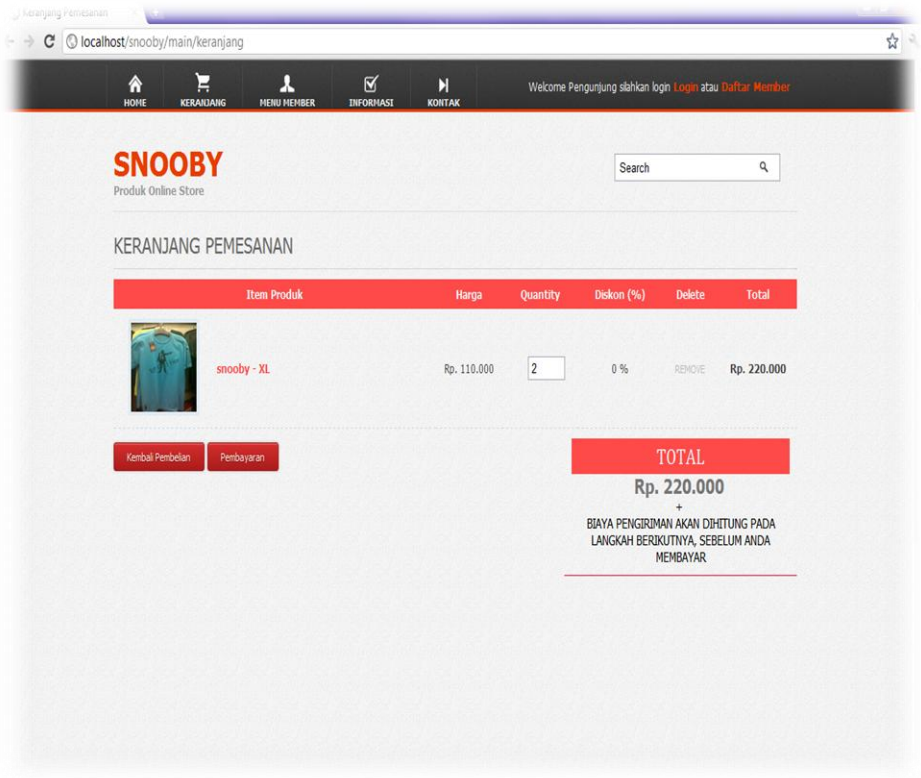


Gambar 5.3 Menu Registrasi Member

1.4. Keranjang Belanja Member

Pada tampilan utama member, konsumen dapat memilih produk yang ingin dipesan, apabila ada produk yang ingin dipesan, maka konsumen dapat menekan tombol “Tambah Pemesanan”. Apabila ingin memilih produk lagi, maka terlebih dahulu menekan tombol “Keranjang”,

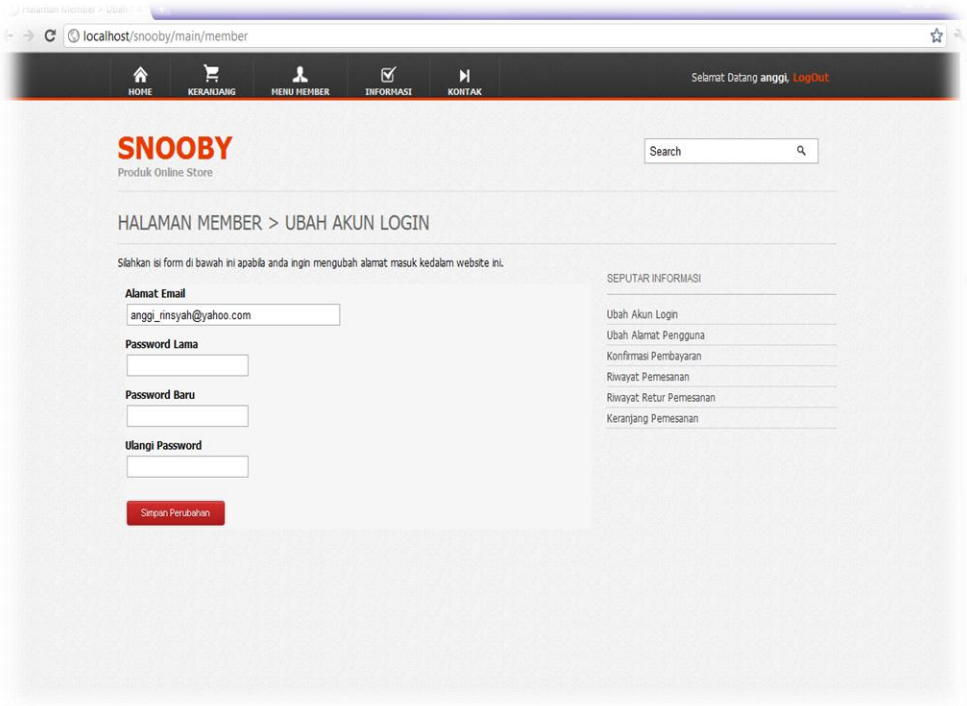
kemudian menekan tombol “Kembali Pembelian”. Untuk menghapus produk yang telah dipesan menekan tombol yang berada di bawah tulisan “Remove”. Apabila telah selesai, maka konsumen dapat menekan tombol “Pembayaran”.



Gambar 5.4 Tampilan Keranjang Belanja Member

1.5. Member

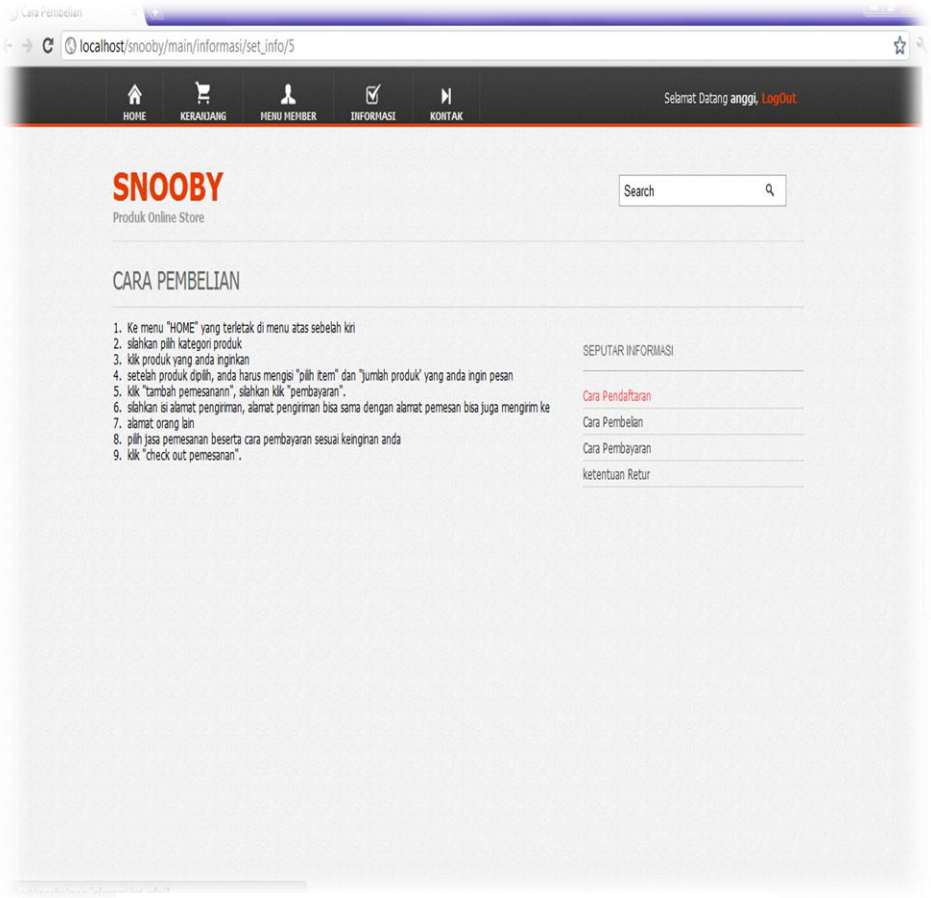
Menu ini merupakan halaman khusus *user* yang berisi mengenai data user. Di dalam menu ini, *user* dapat mengganti data konsumen, mengganti *password*, dan menu konfirmasi pembayaran.



Gambar 5.5 Tampilan Member

1.6. Menu Informasi

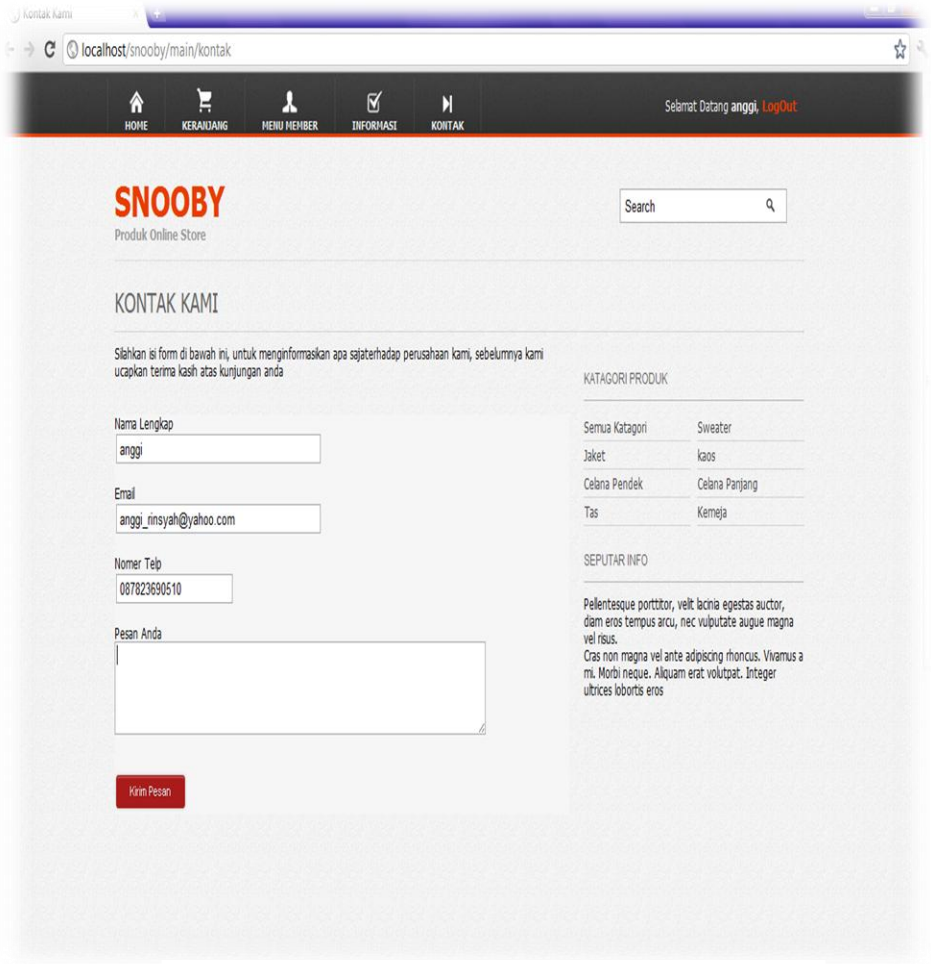
Menu ini berisi mengenai tata cara melakukan transaksi/pemesanan pada *Website* Distro Snooby.



Gambar 5.6 Tampilan Menu Informasi

1.7. Menu Kontak

Menu ini berfungsi untuk memudahkan *user* dalam menyampaikan saran atau kritik kepada Distro Snooby.



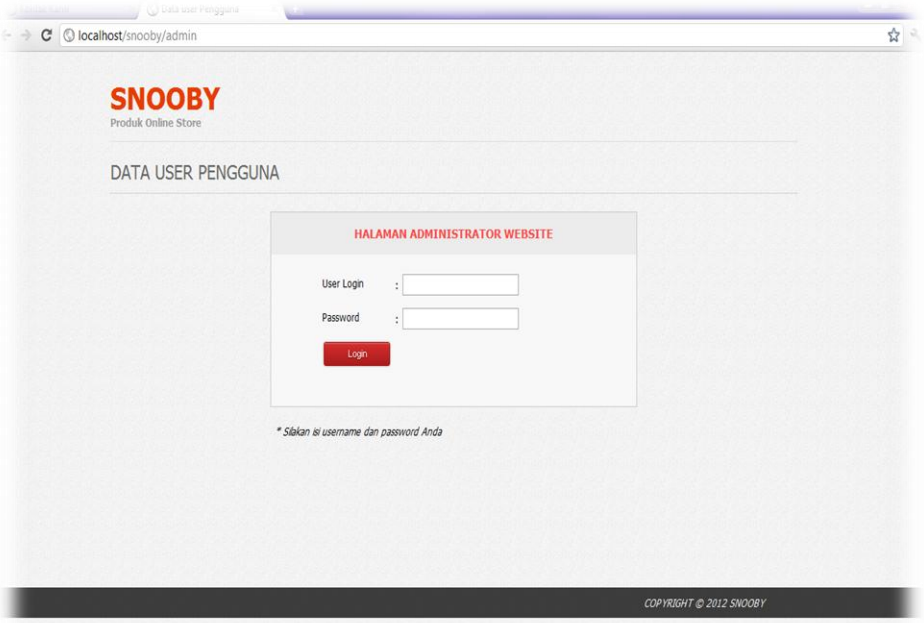
Gambar 5.7 Tampilan Menu Kontak

2. Tampilan *Admin*

Tampilan *admin* merupakan tampilan yang digunakan oleh *admin* untuk melakukan kontrol terhadap data-data yang akan ditampilkan pada *Website* Distro Snooby. Selain itu *admin* juga dapat melakukan kontrol terhadap data produk, data pemesanan, serta mencetak laporan stock barang dan laporan penjualan. Berikut menu-menu yang terdapat di dalam tampilan *admin*.

2.1. *Log in* Administrator

Untuk dapat masuk ke halaman utama administrator, terlebih dahulu harus melakukan proses *login* dengan memasukkan *username* dan *password admin*.



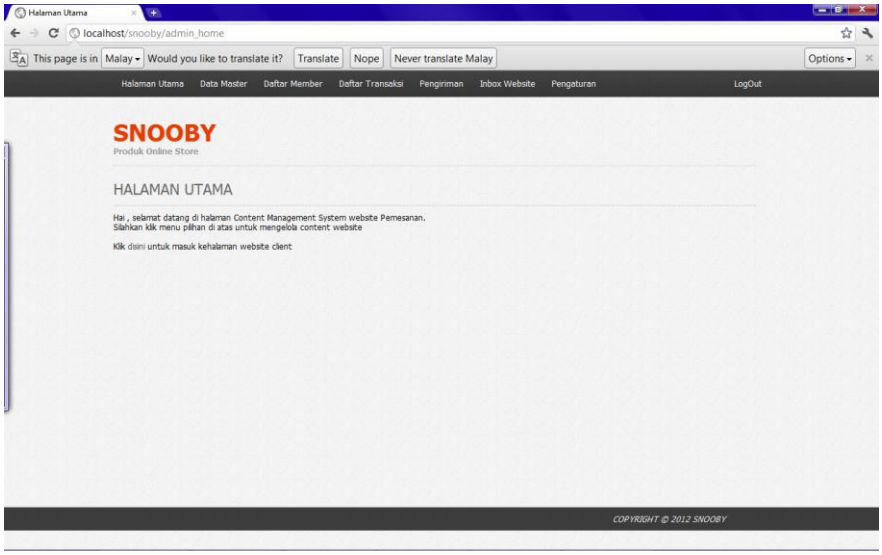
Gambar 5.8 *Log in Admin*

2.2. Tampilan Utama Administrator

Pada halaman utama administrator ini terdapat delapan menu, yaitu: Halaman Utama, Data Master, Daftar Member, Daftar Transaksi, Pengiriman, Inbox Website, Pengaturan, log out.

1. Tampilan Utama

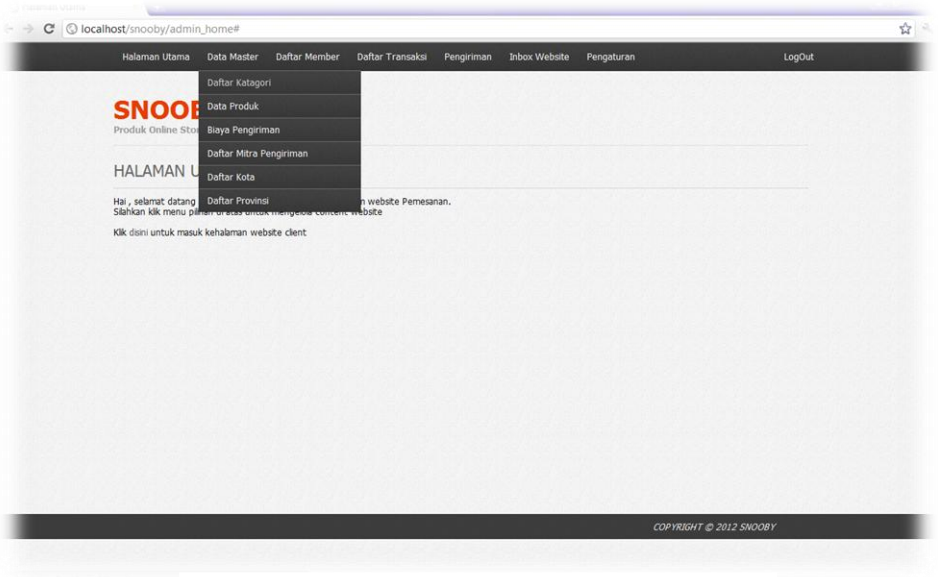
Pada halaman *index* ini terdapat beberapa fungsi yang dapat membantu *admin* untuk menjalankan program ini, diantaranya pada *header* terdapat *link* (Halaman Utama, Data Master, Daftar Member, Daftar Transaksi, Pengiriman, Inbox Website, Pengaturan).



Gambar 5.9 Menu Utama *Admin*

2. Data Master

Dalam data Master terdapat *link* (Daftar Kategori, Daftar Produk, Biaya Pengiriman, Daftar Mitra Pengiriman, Daftar kota, Daftar Propinsi), stiap link Data Master memiliki kegunaan sebagai berikut:



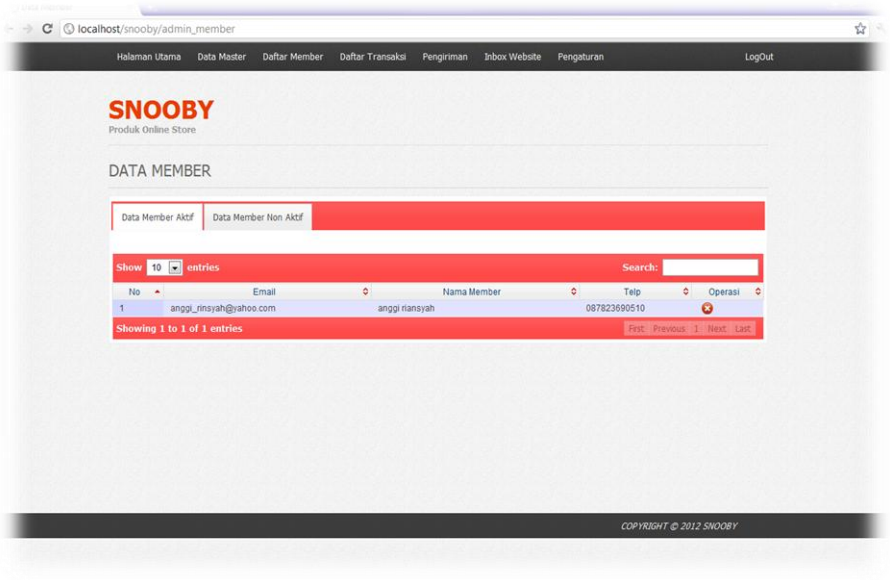
Gambar 5.10 Menu Data Master *admin*

- Daftar Kategori Menu ini berfungsi untuk menampilkan kategori produk. Di dalam menu ini *admin* melakukan tambah, *edit*, dan hapus kategori
- Data Produk Menu ini berfungsi untuk menampilkan data produk. Di dalam menu ini terdapat beberapa fasilitas untuk tambah, hapus, dan *edit* produk.

- Biaya Pengiriman menu ini berfungsi untuk menampilkan daftar biaya pengiriman, di dalam ini terdapat pasilitas edit biaya pengiriman.
- Daftar Mitra Pengiriman Menu ini berfungsi untuk menampilkan daftar Mitra Pengiriman. Di dalam menu ini terdapat beberapa fasilitas untuk tambah, hapus, dan *edit* Daftar Mitra Pengiriman.
- *Daftar Kota* Menu ini berfungsi untuk menampilkan daftar Kota. Di dalam menu ini terdapat beberapa fasilitas untuk tambah, hapus, dan *edit* daftar kota.
- *Daftar Propinsi* Menu ini berfungsi untuk menampilkan daftar propinsi. Di dalam menu ini terdapat beberapa fasilitas untuk tambah, hapus, dan *edit* daftar propinsi.

3. Menu Member

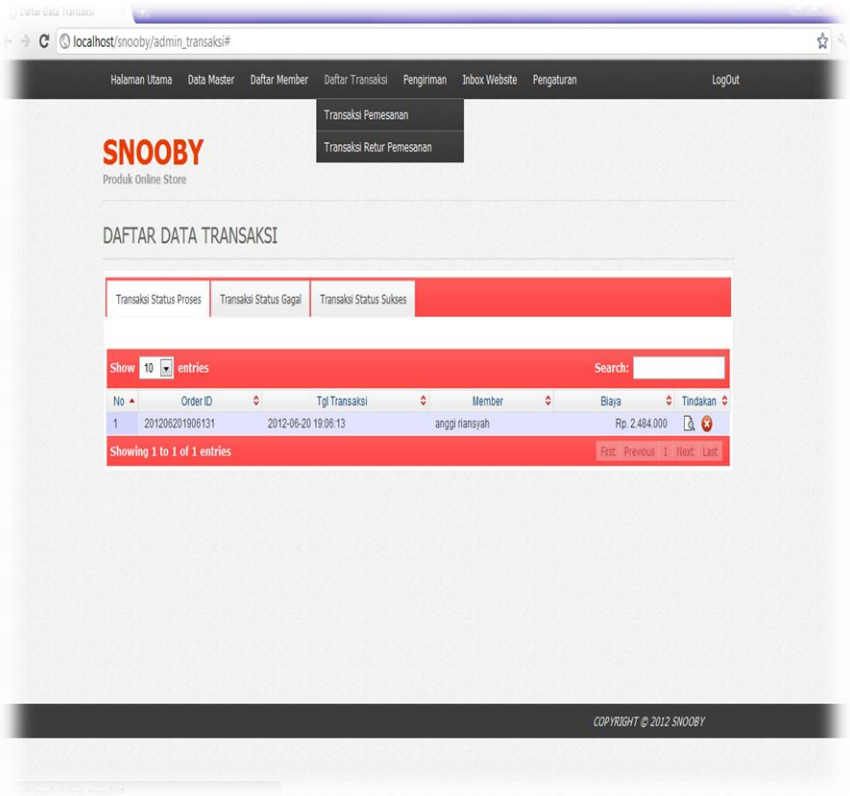
   Menu ini berfungsi untuk melihat detail data member.



Gambar 5.11 Tampilan Menu Member

4. Daftar Transaksi

   Dalam Daftar Transakasi terdapat *link* (Transakasi Pemesanan, Transakasi Retur Pemesanan), stiap link Daftar Transakasi memiliki kegunaan sebagai berikut:

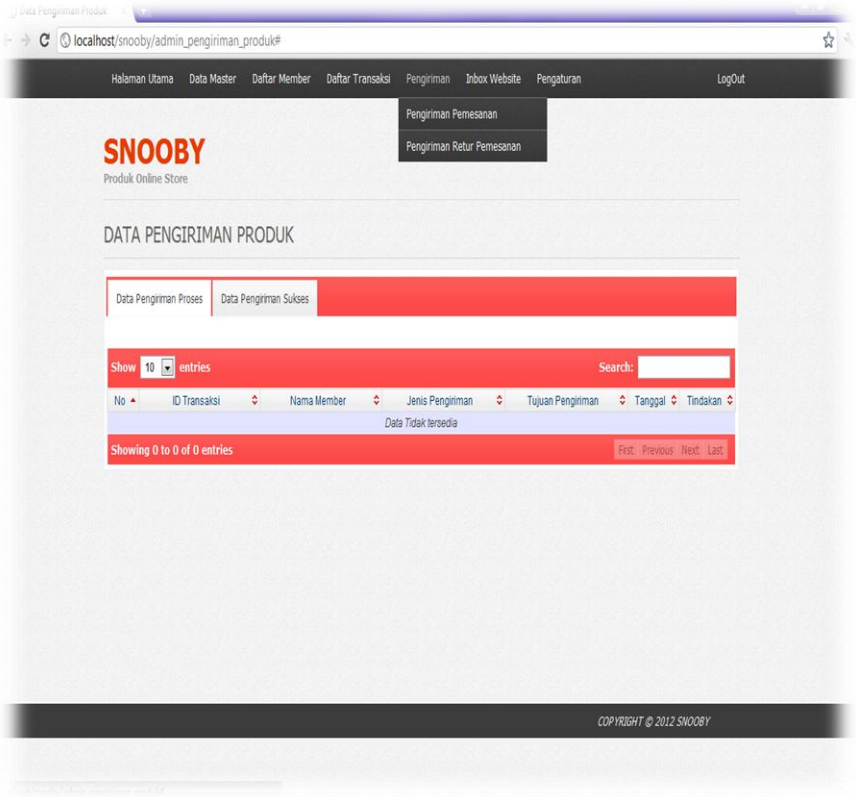


Gambar 5.12 Daftar Transaksi

- *Transaksi Pemesanan* Menu ini berfungsi untuk menampilkan detail Pemesanan member. Di dalam menu ini *admin* dapat melihat status Pemesanan dan dapat melakukan perubahan terhadap status pemesanan.
- *Transaksi retur Pemesanan* Menu ini berfungsi untuk menampilkan detail Retur Pemesanan member. Di dalam menu ini *admin* dapat melihat status retur dan dapat melakukan perubahan terhadap status retur pemesanan.

5. Pengiriman

Dalam menu Pengiriman terdapat *link* (Pengiriman Pemesanan, Pengiriman Retur Pemesanan), stiap link Daftar Transakasi memiliki kegunaan sebagai berikut:



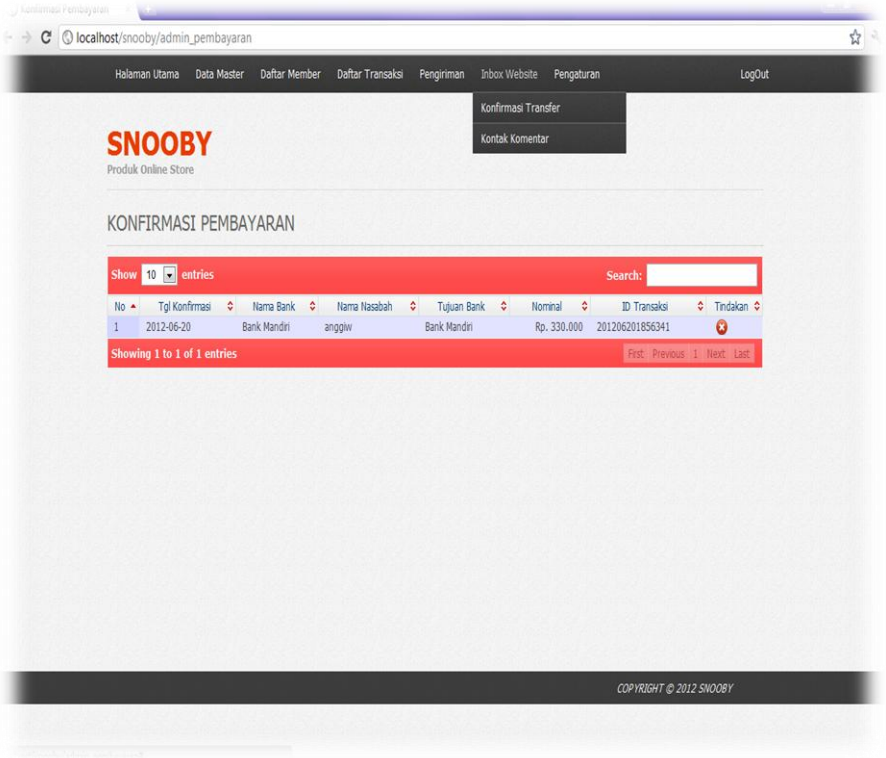
Gambar 5.13 Tampilan Menu Pengiriman

- Pengiriman Pemesanan Menu ini berfungsi untuk menampilkan detail Pengiriman. Di dalam menu ini *admin* dapat melihat status Pengiriman dan dapat melakukan perubahan terhadap status Pengiriman.

- Pengiriman Retur Pemesanan Menu ini berfungsi untuk menampilkan detail Retur Pengiriman. Di dalam menu ini *admin* dapat melihat status retur Pengiriman dan dapat melakukan perubahan terhadap status pengiriman retur pemesanan.

6. Inbox Website

   Dalam menu Inbox Website terdapat *link* (Konfirmasi Tranfer, Kontak Komentar), stiap link Inbox Wbsite memiliki kegunaan sebagai berikut:

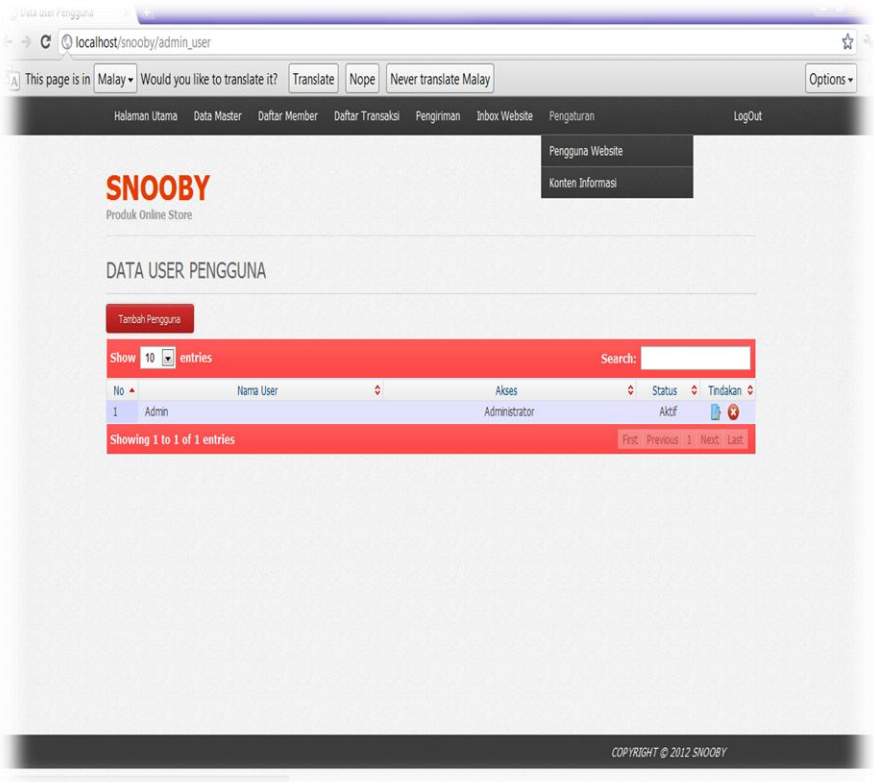


Gambar 5.14 Tampilan Menu Inbox Website

- *Konfirmasi Tranfer* Menu ini berfungsi untuk menampilkan detail konfirmasi tranfer. Di dalam menu ini *admin* dapat melihat status konfirmasi pembayaran dan memiliki fasilitas hapus..
- *Kontak Komentar* Menu ini berfungsi menampilkan saran-saran dari konsumen.

7. Pengaturan

Dalam menu Pengaturan terdapat *link* (Pengguna Website, Konten Informasi), stiap link Pengaturan memiliki kegunaan sebagai berikut:



Gambar 5.15 Tampilan Menu Pengatura

- *Pengguna Website* Menu ini berfungsi untuk menampilkan detail Pengguna Website. Di dalam menu ini terdapat beberapa fasilitas untuk tambah, hapus, dan *edit* Pengguna website.
- *Konten Informasi* Menu ini berfungsi untuk menampilkan detail Data Konten Informasi. Di dalam menu ini terdapat beberapa fasilitas untuk tambah, hapus, dan *edit* data Konten Informasi.

5.1.6. Implementasi Instalasi Program

Implementasi instalasi program merupakan aspek awal untuk menjelaskan bagaimana perancangan aplikasi *Website Penjualan* dapat dijalankan secara utuh didalam sistem komputer. Berikut langkah-langkah yang dilakukan dalam melakukan proses instalasi program agar berjalan pada kondisi *offline*. Berikut merupakan langkah dalam instalasi program :

1. Pastikan sudah memiliki *software* Wampserver untuk servernya.
2. Lakukan konfigurasi *software* Wampserver , dengan mengaktifkan Apache dan MySQL.
3. Konfigurasi *database* yang ada pada PhpMyAdmin, dengan menuju *link*, http://localhost/phpmyadmin/.
4. Konfigurasi penyimpanan file data *web* Distro Snooby secara *offline*, dengan menyimpan file *web* Distro Snooby pada folder *www*.

5. Konfigurasi Koneksi Database.
6. Setelah proses instalasi selesai semua maka untuk mengakses menu utama *Website* Distro Snooby menggunakan alamat : http://localhost/snooby dan untuk *admin* menggunakan alamat http://localhost/admin.

5.2. Pengujian Program

Pengujian program merupakan bagian yang penting dalam *siklus* pembangunan perangkat lunak. Pengujian dilakukan untuk menjamin kualitas dan untuk mengetahui kelemahan dari perangkat lunak. Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang memiliki kualitas yang baik yaitu mampu untuk mempersentasikan kajian pokok dari spesifikasi, analisis, perancangan, dan pengkodean dari perangkat lunak itu sendiri.

5.2.1. Rencana Pengujian

Pengujian perangkat lunak ini menggunakan metode pengujian *blackbox*. Alasannya karena metode *blackbox* dapat mengetahui apakah program dapat berjalan dengan benar dan sesuai dengan yang diharapkan. Kelebihan metode *blackbox* ini adalah kita dapat melakukan pengujian program dengan kondisi nyata saat program berjalan, jadi kita dapat mengetahui langsung apakah program dapat berjalan dan menghasilkan *output* seperti yang diharapkan.

Pengujian Sistem Informasi Penjualan ini menggunakan data uji berupa data masukan dari *user* dan *admin* pada program aplikasi yang telah dibuat.

Tabel di bawah ini akan menjelaskan mengenai rencana pengujian *Website Penjualan* Distro Snooby.

Tabel 5.1 Rencana Pengujian Untuk Tampilan *User*

Kasus Uji	Butir Uji
Registrasi Member	Melakukan Pengisian data pendaftaran *user*.
Login Member	Melakukan proses *login* member.
Ganti password	Melakukan ganti *password* member.
Pesanan barang	Memasukkan pemesanan data barang ke keranjang belanja.
	Melakukan *update* jumlah pemesanan di keranjang belanja.
Input data pemesan	Melakukan penginputan data pemesan.
Input data retur	Melakukan penginputan data retur
Input data pembayaran	Melakukan penginputan data konfirmasi pembayaran

Tabel 5.2 Rencana Pengujian Untuk Tampilan *Admin*

Kasus Uji	Butir Uji	Jenis Pengujian
Login admin	Melakukan proses *login* sebagai *admin*	*Blackbox*
Kategori	Melakukan *input*, *edit*, serta hapus kategori	*Blackbox*
Produk	Melakukan *input* dan *edit* data produk.	*Blackbox*
Biaya Pengiriman	Melakukan update biaya kirim	*Blackbox*
Mitra Pengiriman	Melakukan *input*, *edit*, dan hapus	*Blackbox*
Daftar Kota	Melakukan *input*, *edit*, dan hapus	*Blackbox*
Daftar Provinsi	Melakukan *input*, *edit*, dan hapus	*Blackbox*
Daftar Member	Melakukan hapus	*Blackbox*
Daftar Pemesanan	Melakukan proses pencarian, ubah status pemesanan, dan menghapus.	*Blackbox*
Daftar Retur	Melakukan proses pencarian, ubah, serta hapus.	*Blackbox*
Daftar pengiriman	Melakukan cari, edit dan hapus	*Blackbox*
Dagftar pengiriman retur	Melakukan cari, edit, dan hapus	*Blackbox*
Konfirmasi	Melakukan cari dan hapus	*Blackbox*

5.2.2. Kasus dan Hasil pengujian

Pengujian akan dilakukan satu persatu sesuai dengan rencana pengujian yang telah direncanakan. Berikut ini adalah hasil pengujiannya :

5.2.2.1.Pengujian Untuk Tampilan *User*

Pengujian ini dilakukan pada tampilan *user* dari *Website* Distro Snooby sesuai dengan rencana pengujian yang telah dibuat, antara lain sebagai berikut :

1. Proses Registrasi Member

   Berikut ini adalah hasil pengujian registrasi member.

Tabel 5.3 Pengujian Registrasi Member

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi lengkap data pendaftaran member	Berhasil mendaftar dan memunculkan halaman konfirmasi sukses mendaftar	Berhasil dan masuk ke halaman konfirmasi sukses mendaftar	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi lengkap *form* pendaftaran	Registrasi tidak berhasil dan muncul pesan untuk mengisi lengkap *form* pendaftaran	Muncul pesan “Isi terlebih dahulu “	[X] diterima [ ] ditolak
Mengisi *Form* dengan format data yang tidak *valid*	Muncul validasi data berdasarkan *form* yang diisi	Muncul pesan validasi berdasarkan *form* yang salah diisi	[X] diterima [ ] ditolak

2. Proses *Login User*

Berikut ini adalah hasil pengujian *login user*.

Tabel 5.4 Pengujian *Login User*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi *Username* dan *Password Valid*	Berhasil *login* dan masuk ke menu utama member	*Login* berhasil dan masuk ke halaman menu utama member	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi Kombinasi *Password* dan *Username* dengan data yang tidak *valid*	Gagal *login* dan muncul pesan "silahkan login lagi, password dan username yang anda masukkan salah"	*Login* gagal dan muncu pesan "silahkan login lagi, password dan username yang anda masukkan salah"	[X] diterima [ ] ditolak

3. Proses Ganti *Password*

Berikut ini adalah hasil pengujian ganti *password.*

Tabel 5.5 Pengujian Ganti *Password*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi *field password* lama, *password* baru dan konfirmasi *password*	Berhasil merubah *password* lama *user* menjadi *password* yang baru	Perubahan *password user* berhasil.	[X] diterima [ ] ditolak

dengan data *valid*			
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi lengkap *form* ganti *password*	Gagal mengganti *password* dan muncul pesan "isi data yang lengkap jika ingin merubah password Anda"	Ganti *password user* gagal dan muncul pesan"isi data yang lengkap jika ingin merubah password anda"	[X] diterima [ ] ditolak
Mengisi *password* lama dengan data tidak *valid*	Gagal mengganti *password* dan muncul pesan "password lama anda salah"	Password gagal diganti dan muncul pesan "password lama anda salah"	[X] diterima [ ] ditolak
Mengisi *field password* baru berbeda dengan *field* konfirmasi *password*	Gagal mengganti *password* dan muncul pesan "konfirmasi password baru salah"	*Password* gagal diganti dan muncul pesan "konfirmasi password baru salah"	[X] diterima [ ] ditolak

4. Pesanan Barang

Berikut ini adalah hasil pengujian pemesanan barang.

Tabel 5.6 Pengujian Pemesanan Barang

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukkan data pemesanan ke keranjang belanja dengan menekan tombol "order now"	Data pemesanan berhasil masuk ke keranjang belanja dan muncul halaman keranjang belanja	Data pemesanan berhasil masuk ke keranjang belanja dan muncul halaman keranjang belanja.	[X] diterima [ ] ditolak

Mengupdate jumlah pesanan pada field jumlah di keranjang belanja	Jumlah data dikeranjang belanja berhasil di *update*.	Berhasil mengupdate jumlah pemesanan barang	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Belum *login* dan menekan tombol "order now"	Pemesanan tidak dapat dilakukan dan Muncul pesan "Login dulu untuk melakukan pemesanan barang"	Muncul pesan "Login dulu untuk melakukan pemesanan barang	[X] diterima [ ] ditolak
Memasukkan *field* jumlah bukan angka	*Update* jumlah tidak dapat dilakukan dan *field* jumlah kosong	*Field* jumlah di keranjang belanja kosong	[X] diterima [ ] ditolak

5. Input Data Pemesan

   Berikut ini adalah hasil pengujian penginputan data pemesan setelah melakukan proses pemesanan.

Tabel 5.7 Pengujian *Input* Data Pemesan

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi lengkap data pemesan dengan format yang benar	Data pemesan berhasil dimasukkan dan muncul halaman detail pemesanan.	Muncul halaman detail pesanan dengan data pemesan yang telah diinputkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**

Mengosongkan salah satu *field* pemesan	muncul pesan untuk mengisi lengkap form pendaftaran	Muncul pesan "Isi terlebih dahulu "	[X] diterima [ ] ditolak
Mengisi *Form* dengan format data yang tidak *valid*	Muncul validasi data berdasarkan form yang diisi	Muncul pesan validasi berdasarkan *form* yang salah diisi	[X] diterima [ ] ditolak

6. Input Data Retur

Berikut ini adalah hasil pengujian penginputan data retur.

Tabel 5.8 Pengujian *Input* Data Retur

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi lengkap data retur dengan format yang benar	Data pemesan berhasil dimasukkan dan muncul halaman riwayat retur	Muncul halaman riwayat retur dengan data retur yang di inputkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengosongkan salah satu *field*	muncul pesan untuk mengisi lengkap form pendaftaran	Muncul pesan "Isi terlebih dahulu "	[X] diterima [ ] ditolak
Mengisi *Form* dengan format data yang tidak *valid*	Muncul validasi data berdasarkan form yang diisi	Muncul pesan validasi berdasarkan *form* yang salah diisi	[X] diterima [ ] ditolak

7. Input data pembayaran

Berikut ini adalah hasil pengujian penginputan data pembayaran.

Tabel 5.9 Pengujian *Input* Data Pembayaran

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi lengkap data pembayaran dengan format yang benar	Data Pembayaran berhasil dimasukkan dan muncul halaman Konfirmasi Pembayaran	Muncul halaman konfirmasi pembayaran dengan data pembayaran yang di inputkan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengosongkan salah satu *field*	muncul pesan untuk mengisi lengkap form pendaftaran	Muncul pesan "Isi terlebih dahulu"	[X] diterima [ ] ditolak
Mengisi *Form* dengan format data yang tidak *valid*	Muncul validasi data berdasarkan form yang diisi	Muncul pesan validasi berdasarkan *form* yang salah diisi	[X] diterima [ ] ditolak

5.2.3. Kesimpulan dan Hasil Pengujian

Berdasarkan hasil pengujian yang telah dilakukan dengan kasus di atas dapat ditarik kesimpulan bahwa program *Website Penjualan* Distro Snooby bebas dari kesalahan *syntax* dan secara fungsional telah mengeluarkan hasil yang *valid* karena proses validasi data dapat berfungsi dengan baik.