BAB V

PENGUJIAN DAN IMPLEMENTASI SISTEM

5.1. Pengujian

Pengujian merupakan suatu investigasi yang dilakukan untuk mendapatkan informasi mengenai kualitas dari produk atau layanan yang sedang diuji (*under test*). Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara obyektif dan independen, yang bermanfaat dalam operasional bisnis untuk memahami tingkat risiko pada implementasinya. Teknik-teknik pengujian mencakup, namun tidak terbatas pada, proses mengeksekusi suatu bagian program atau keseluruhan aplikasi dengan tujuan untuk menemukan bug perangkat lunak.

5.1.1. Rencana Pengujian

Pengujian perangkat lunak yang dilakukan oleh penulis menggunakan metode pengujian *Black Box*. Pengujian *Black Box* merupakan pengujian yang berfokus pada persyaratan fungsional perangkat lunak yang dibuat.

Tabel 5.1 Rencana pengujian.

Kelas Uji	**Butir Uji**	**Tingkat Pengujian**	**Jenis Pengujian**
Login User	Input *User Name*	Modul	*Black Box*
	Input *Password*	Modul	*Black Box*
	Pengecekan *User Name*	Sistem	*Black Box*
	Pengecekan *Password*	Sistem	*Black Box*
Pengujian Pengisian Data	Update data kategori	Modul	*Black Box*
	Pengecekan data kategori	Sistem	*Black Box*
	Input data produk	Modul	*Black Box*
	Pengecekan data produk	Sistem	*Black Box*
	Input data jenis	Modul	*Black Box*
	Pengecekan data jenis	Sistem	*Black Box*
	Input data order	Modul	*Black Box*
	Pengecekan data order	Sistem	*Black Box*

5.1.2. Kasus dan Hasil Pengujian

1. Data User Admin

Tabel 5.2 Pengujian Data User Admin

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Submit	Data yang sudah di isi lalu di simpan ke dalam database.	Dapat menyimpan data baru yang telah di masukan ke dalam database.	[X] diterima [ ] ditolak
Klik tombol Reset	Mengosongkan inputan atau data yang sudah di isi	Dapat mengosongkan data yang sudah di isi dan kembali ke semula kosong.	[X] diterima [ ] ditolak
Klik Tombol Update	Mengubah data admin yang sudah ada.	Dapat mengubah data admin yang sudah ada.	[X] diterima [ ] ditolak
Klik Tombol Delete	Menghapus data yang sudah ada dalam database.	Dapat menghapus data yang sudah ada dalam database.	[X] diterima [ ] ditolak

2. Data Kategori

Tabel 5.3 Pengujian Data kategori

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Submit	Data yang sudah di isi lalu di simpan ke dalam database.	Dapat menyimpan data baru yang telah di masukan ke dalam database.	[X] diterima [ ] ditolak
Klik Tombol Reset	Mengosongkan inputan atau data yang sudah di isi	Dapat mengosongkan data yang sudah di isi dan kembali ke semula kosong	[X] diterima [ ] ditolak
Klik Tombol Update	Mengubah data kategori yang	Dapat mengubah data admin yang	[X] diterima [ ] ditolak

	sudah ada.	sudah ada.	
Klik Tombol Delete	Menghapus data yang sudah ada dalam database.	Dapat menghapus data yang sudah ada dalam database.	[X] diterima [ ] ditolak
Klik Tombol Choose	Menambah gambar kategori ke dalam database.	Dapat menambahkan gambar kategori.	[X] diterima [ ] ditolak

3. Data Produk

Tabel 5.4 Pengujian Data Produk

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Submit	Dapat mengisi data baru kedalam database tbl_barang	Dapat menambah data baru ke dalam database tbl_barang	[X] diterima [ ] ditolak
Klik Tombol Reset	Mengosongkan inputan atau data yang sudah di isi	Dapat mengosongkan data yang sudah di isi dan kembali ke semula kosong	[X] diterima [ ] ditolak
Klik Tombol Choose	Menambah gambar produk ke dalam database.	Dapat menambahkan gambar produk.	[X] diterima [ ] ditolak
Klik Update Barang	Dapat mengubah data yang ada.	Dapat mengubah data yang sudah ada.	[X] diterima [ ] ditolak
Klik Tombol Delete	Dapat menghapus data.	Dapat menghapus data	[X] diterima [ ] ditolak
Klik Tombol Change Jumlah Ukuran	Dapat mengubah jumlah ukuran	Dapat mengubah jumlah ukuran yang ada di bagi perukuran	[X] diterima [ ] ditolak

4. Data Jenis

Tabel 5.5 Pengujian Data Jenis

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Submit	Dapat mengisi data baru kedalam database	Dapat menambah data baru ke dalam database	[X] diterima [ ] ditolak

	tbl_barang	tbl_barang	
Klik Tombol Reset	Mengosongkan inputan atau data yang sudah di isi	Dapat mengosongkan data yang sudah di isi dan kembali ke semula kosong	[X] diterima [ ] ditolak
Klik Tombol Update	Dapat mengubah data yang ada.	Dapat mengubah data yang sudah ada.	[X] diterima [ ] ditolak
Klik Tombol Delete	Dapat menghapus data.	Dapat menghapus data	[X] diterima [ ] ditolak

5. Data Order

Tabel 5.6 Pengujian Data Order

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol Imput Pembayaran	Dapat menampilkan daftar produk yang sudah dipesan dan belum dibayar.	Dapat menampilkan daftar produk sudah dipesan.	[X] diterima [ ] ditolak
Klik Tombol Bayar	Dapat menampilkan daftar produk yang sudah dipesan dan belum dibayar.menjadi sudah dibayar.	Dapat menampilkan daftar produk yang sudah dibayar.	[X] diterima [ ] ditolak

6. Data Penjualan

Tabel 5.7 Pengujian Data Penjualan

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Tombol send kirim	Dapat merubah status kirim yang tadina no menjadi ok..	Dapat merubah status kirim yang tadina no menjadi ok.	[X] diterima [ ] ditolak

5.1.3. 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.

5.2. Implementasi

Implementasi *Prototype* Sistem Informasi Penjualan Berbasis Web pada Distro Vicious Clothing Company dilakukan menggunakan pemrograman DREAMWEAVER CS4 dan bahasa pemrograman PHP, HTML, JavaScript serta basis data yang digunakan adalah XAMPP. Implementasi dan pengujian sepenuhnya dilakukan di perangkat keras PC (*Personal Computer*) dengan sistem operasi Microsoft Windows 7.

5.2.1. Batasan Implementasi

Dalam mengendalikan aplikasi ini ada beberapa hal yang menjadi batasan implementasi, yaitu perangkat lunak ini hanya difokuskan pada proses penjualan dan pemesanan saja, kemudian di tambah dengan pengelolaan data admin berisi data kategori, data produk, jenis dan data order.

5.2.2. Implementasi Perangkat Lunak

Untuk pengembangan perangkat lunak ini digunakan *Dreamweaver CS4, Xampp 1.7.0, PHP 5.2.8, dan MySQL 5.1.30. Dreamweaver CS 4* dipilih sebagai perangkat lunak untuk *content management system*, *Xampp* dipilih sebagai perangkat lunak untuk *web server, PHP* dipilih sebagai perangkat lunak pengembangan karena menyediakan fasilitas yang memadai untuk membuat

perangkat lunak berbasis *web* yang dinamis. Sementara *MySQL* digunakan sebagai perangkat lunak pengembangan dalam pembuatan basis data**.**

5.2.3. Implementasi Perangkat Keras

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

1. Server

a. Menggunakan minimal *Intel Pentium 4* atau sekelasnya.

b. Menggunakan *RAM* minimal 256MB.

c. Tersedia *Hardisk* untuk media penyimpanan, minimal 100 MB untuk *server*, diluar basis data.

d. *Mouse, keyboard,* dan *monitor* sebagai antarmuka.

2. Client

a. Menggunakan minimal *Intel Pentium 4* atau sekelasnya.

b. Menggunakan *RAM* minimal 256MB.

c. *Mouse, keyboard,* dan *monitor* sebagai antarmuka.

5.2.4. Implementasi Basis Data

Berikut implementasi data basis data dalam XAMPP:

```
-- phpMyAdmin SQL Dump
-- version 2.11.4
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Jan 10, 2011 at 06:32 PM
-- Server version: 5.0.51
-- PHP Version: 5.2.5

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
```

```
--
-- Database: `unikom_distro`
--

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

--
-- Table structure for table `admin`
--

CREATE TABLE IF NOT EXISTS `admin` (
  `id_admin` int(11) NOT NULL auto_increment,
  `nama_admin` varchar(25) NOT NULL,
  `alamat_admin` varchar(50) NOT NULL,
  `no_telp_admin` varchar(20) NOT NULL,
  `email_admin` varchar(50) NOT NULL,
  `u_name_admin` varchar(10) NOT NULL,
  `pass_admin` varchar(100) NOT NULL,
  `lev_user` varchar(1) NOT NULL,
  PRIMARY KEY  (`id_admin`),
  UNIQUE KEY `u_name_admin` (`u_name_admin`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;


--
-- Table structure for table `barang`
--

CREATE TABLE IF NOT EXISTS `barang` (
  `id_barang` int(11) NOT NULL auto_increment,
  `id_kategori` int(11) NOT NULL,
  `nama_barang` varchar(50) NOT NULL,
  `jns_brg` int(11) NOT NULL,
  `harga` varchar(10) NOT NULL,
  `gambar` varchar(500) NOT NULL,
  `stok` smallint(3) NOT NULL,
  PRIMARY KEY  (`id_barang`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1
AUTO_INCREMENT=111145 ;

--
-- RELATIONS FOR TABLE `barang`:
--   `id_kategori`
--       `kategori` -> `id_kategori`
--   `jns_brg`
--       `jenis` -> `id_jenis`
```

```
--

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

--
-- Table structure for table `comment`
--

CREATE TABLE IF NOT EXISTS `comment` (
  `no_comment` int(11) NOT NULL auto_increment,
  `id_member` int(11) NOT NULL,
  `id_brg` int(11) NOT NULL,
  `tgl_comment` date NOT NULL,
  `comment` text NOT NULL,
  PRIMARY KEY  (`no_comment`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

--
-- RELATIONS FOR TABLE `comment`:
--   `id_brg`
--       `barang` -> `id_barang`
--   `id_member`
--       `member` -> `id_member`
--

--
-- Table structure for table `data_ongkos`
--

CREATE TABLE IF NOT EXISTS `data_ongkos` (
  `id` int(11) NOT NULL auto_increment,
  `id_provinsi` int(11) NOT NULL,
  `ongkos` double NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

--
-- RELATIONS FOR TABLE `data_ongkos`:
--   `id_provinsi`
--       `provinsi` -> `id_provinsi`
--




--
```

```
-- Table structure for table `detail_jual`
--

CREATE TABLE IF NOT EXISTS `detail_jual` (
  `no_jual` varchar(15) NOT NULL,
  `id_barang` int(11) NOT NULL,
  `jumlah_jual` smallint(2) NOT NULL,
  `idukuran` char(4) NOT NULL,
  `harga_jual` double default NULL,
  PRIMARY KEY  (`no_jual`,`id_barang`,`idukuran`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- RELATIONS FOR TABLE `detail_jual`:
--   `idukuran`
--       `ukuran` -> `id`
--   `id_barang`
--       `barang` -> `id_barang`
--   `no_jual`
--       `penjualan` -> `no_jual`
--

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

--
-- Table structure for table `detail_order`
--

CREATE TABLE IF NOT EXISTS `detail_order` (
  `no_order` varchar(15) NOT NULL,
  `id_barang` int(11) NOT NULL,
  `jumlah_barang` smallint(2) NOT NULL,
  `idukuran` char(4) NOT NULL,
  `harga_jual` double default NULL,
  PRIMARY KEY  (`no_order`,`id_barang`,`idukuran`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- RELATIONS FOR TABLE `detail_order`:
--   `idukuran`
--       `ukuran` -> `id`
--   `id_barang`
--       `barang` -> `id_barang`
--   `no_order`
--       `order` -> `no_order`
--
```

```
-- --------------------------------------------------------
--
-- Table structure for table `det_ukuran_barang`
--

CREATE TABLE IF NOT EXISTS `det_ukuran_barang` (
  `idBarang` int(11) default NULL,
  `idUkuran` char(4) NOT NULL,
  `ukuran_stock` smallint(2) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- RELATIONS FOR TABLE `det_ukuran_barang`:
--   `idBarang`
--       `barang` -> `id_barang`
--   `idUkuran`
--       `ukuran` -> `id`
--

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

--
-- Table structure for table `jenis`
--

CREATE TABLE IF NOT EXISTS `jenis` (
  `id_jenis` int(11) NOT NULL auto_increment,
  `nama_jenis` varchar(20) NOT NULL,
  `deskripsi` text NOT NULL,
  PRIMARY KEY  (`id_jenis`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=10
;

-- --------------------------------------------------------
--
-- Table structure for table `kategori`
--

CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL auto_increment,
  `nama_kategori` varchar(20) NOT NULL,
  `gambar` varchar(50) NOT NULL,
  PRIMARY KEY  (`id_kategori`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=27
;
```

```
-- --------------------------------------------------------
--
-- Table structure for table `member`
--

CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(11) NOT NULL auto_increment,
  `nama_member` varchar(50) NOT NULL,
  `jns_kel` char(1) NOT NULL,
  `tgl_lahir` date NOT NULL,
  `provinsi` int(11) NOT NULL,
  `kota` varchar(30) NOT NULL,
  `alamat` varchar(50) NOT NULL,
  `kode_pos` varchar(10) NOT NULL,
  `no_telp` varchar(25) NOT NULL,
  `email` varchar(30) NOT NULL,
  `u_name` varchar(30) NOT NULL,
  `pass` varchar(500) NOT NULL,
  PRIMARY KEY  (`id_member`),
  UNIQUE KEY `u_name` (`u_name`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=19
;

--
-- RELATIONS FOR TABLE `member`:
--   `provinsi`
--       `provinsi` -> `id_provinsi`
--

-- --------------------------------------------------------
--
-- Table structure for table `order`
--

CREATE TABLE IF NOT EXISTS `order` (
  `no_order` varchar(15) NOT NULL,
  `tgl_order` date NOT NULL,
  `id_member` int(11) NOT NULL,
  `status_beli` char(1) NOT NULL,
  `status_bayar` char(1) NOT NULL,
  `status_kirim` char(1) NOT NULL,
  `total` decimal(15,0) NOT NULL,
  PRIMARY KEY  (`no_order`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

```
--
-- RELATIONS FOR TABLE `order`:
--   `id_member`
--       `member` -> `id_member`
--
-- --------------------------------------------------------

--
-- Table structure for table `pembayaran`
--

CREATE TABLE IF NOT EXISTS `pembayaran` (
  `No_Bayar` varchar(15) NOT NULL,
  `no_order` varchar(15) default NULL,
  `tanggal_bayar` date default NULL,
  `nama_pembayar` varchar(30) NOT NULL,
  `no_rekening` varchar(25) NOT NULL,
  `nama_tujuan` varchar(30) default NULL,
  `alamat_tujuan` varchar(50) NOT NULL,
  `provinsi` varchar(30) NOT NULL,
  `kode_pos` char(6) NOT NULL,
  PRIMARY KEY  (`No_Bayar`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- RELATIONS FOR TABLE `pembayaran`:
--   `no_order`
--       `order` -> `no_order`
-

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

--
-- Table structure for table `penjualan`
--

CREATE TABLE IF NOT EXISTS `penjualan` (
  `no_jual` varchar(15) NOT NULL default '',
  `no_order` varchar(15) default NULL,
  `tanggal_jual` date default NULL,
  `total` double NOT NULL,
  PRIMARY KEY  (`no_jual`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


--
```

```
-- RELATIONS FOR TABLE `penjualan`:
--   `no_order`
--       `order` -> `no_order`
--


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

--
-- Table structure for table `provinsi`
--

CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(11) NOT NULL,
  `nama_provinsi` varchar(30) NOT NULL,
  PRIMARY KEY  (`id_provinsi`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

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

--
-- Stand-in structure for view `stokallbarang`
--
CREATE TABLE IF NOT EXISTS `stokallbarang` (
`total` decimal(27,0)
);
-- --------------------------------------------------------

--
-- Stand-in structure for view `stokallsizeproduk`
--
CREATE TABLE IF NOT EXISTS `stokallsizeproduk` (
`idbarang` int(11)
,`nama_barang` varchar(50)
,`harga` varchar(10)
,`gambar` varchar(500)
,`totalstok` decimal(27,0)
);
-- --------------------------------------------------------
--
-- Stand-in structure for view `totalpenjualan`
--
CREATE TABLE IF NOT EXISTS `totalpenjualan` (
`total` decimal(37,0)
);
-- --------------------------------------------------------
```

```
--
-- Table structure for table `ukuran`
--

CREATE TABLE IF NOT EXISTS `ukuran` (
  `id` char(4) NOT NULL,
  `nama_ukuran` varchar(15) NOT NULL,
  PRIMARY KEY  (`id`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1;


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

--
-- Structure for view `stokallbarang`
--
DROP TABLE IF EXISTS `stokallbarang`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `unikom_distro`.`stokallbarang` AS select
sum(`unikom_distro`.`barang`.`stok`) AS `total` from `unikom_distro`.`barang`;

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

--
-- Structure for view `stokallsizeproduk`
--
DROP TABLE IF EXISTS `stokallsizeproduk`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `unikom_distro`.`stokallsizeproduk` AS select
`unikom_distro`.`det_ukuran_barang`.`idBarang` AS
`idbarang`,`unikom_distro`.`barang`.`nama_barang` AS
`nama_barang`,`unikom_distro`.`barang`.`harga` AS
`harga`,`unikom_distro`.`barang`.`gambar` AS
`gambar`,sum(`unikom_distro`.`det_ukuran_barang`.`ukuran_stock`) AS
`totalstok` from (`unikom_distro`.`det_ukuran_barang` join
`unikom_distro`.`barang` on((`unikom_distro`.`det_ukuran_barang`.`idBarang`
= `unikom_distro`.`barang`.`id_barang`))) group by
`unikom_distro`.`det_ukuran_barang`.`idBarang`;

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


--
```

```
-- Structure for view `totalpenjualan`
--
DROP TABLE IF EXISTS `totalpenjualan`;

CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL
SECURITY DEFINER VIEW `unikom_distro`.`totalpenjualan` AS select
sum(`unikom_distro`.`order`.`total`) AS `total` from `unikom_distro`.`order`;
```

5.2.5. Implementasi Antar Muka

Berikut ini adalah implementasi dari setiap halaman yang dibuat dan dibedakan berdasarkan user.

a. Implementasi Halaman Berdasarkan User Bagian admin

Tabel 5.8 Implementasi Halaman Berdasarkan User Bagian admin

Sub Menu	Deskripsi	Nama Form
Login	otentifikasi dengan memeriksa *user name* dan *password* yang dimasukkan pengguna	Form Utama
Data kategori	Update data kategori sesuai yang di inginkan	Form data kategori
Data Produk	Input data produk, termasuk penambahan, pengubahan dan pembatalan	Form produk
Data Order	Mengubah status order	Form datat order
Data Penjualan	Mengubah status kirim	Form data Penjualan
Data Akun	Mengubah data admin	Form data akun

b. Implementasi Halaman Berdasarkan User Pelanggan

Tabel 5.9 Implementasi Halaman Berdasarkan User Pelanggan

Sub Menu	Deskripsi	Nama Form
Login	otentifikasi dengan memeriksa *user name* dan *password* yang dimasukkan pengguna	Form utama
Data Produk	Melihat data produk dan detail produk yang akan di pilih.	Form Utama
Keranjang belanja	Mengkonfirmasi dan mengubah data produk yang akan di pesan.	Form keranjang belanja
Data Order	Input data Order.	Form Order

Konfirmasi Order	Pengecekan data informasi pemesanan atau order barang.	Form Cek Data Order

5.2.6. Implementasi Penginstalan XAMPP

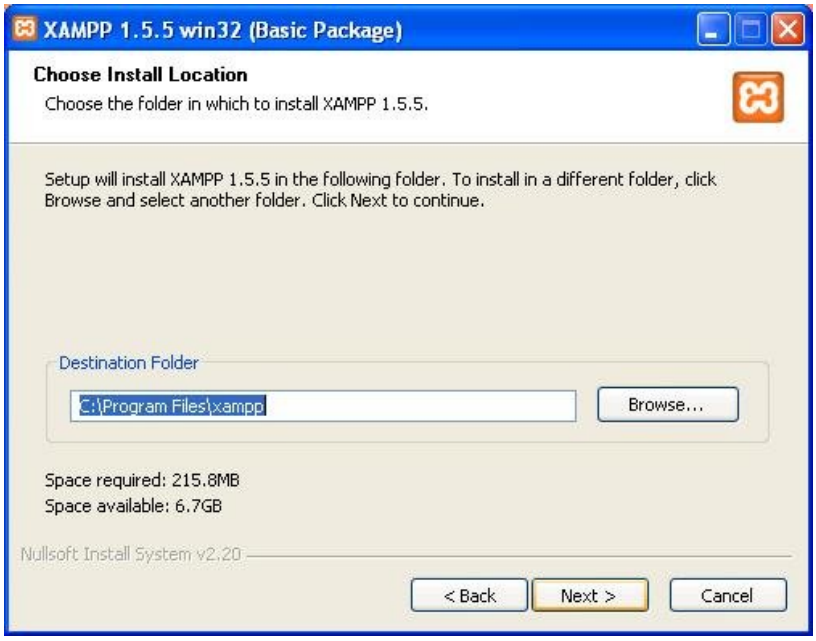
Langkah terakhir adalah memanggil program yang telah di buat menggunakan aplikasi Internet Explorer atau Mozilla Firefox.

1. Buka folder yang terdapat XAMPP.EXE instalasi yang telah dibuat untuk menginstal program.
2. Klik dua kali untuk menjalankan program SETUP.EXE tersebut
3. Setelah program installer dijalankan maka akan tampak seperti gambar di bawah ini. Klik Next untuk melanjutkan.



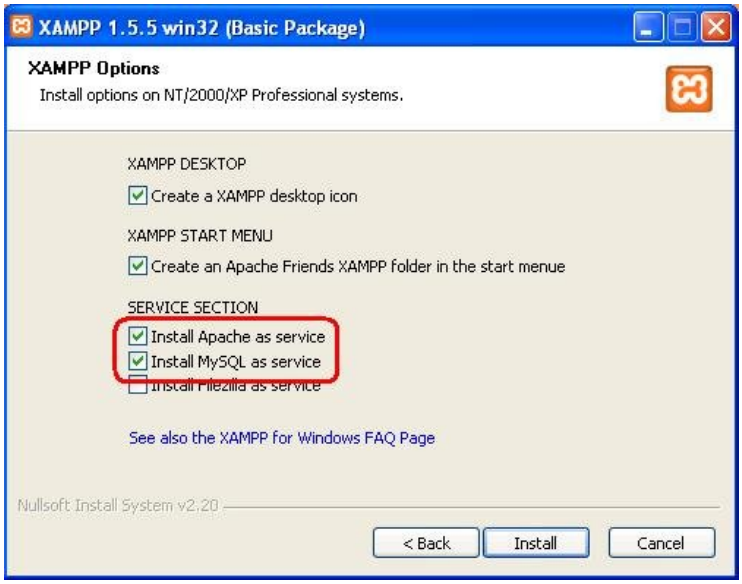
Gambar 5.1 Install xampp

4. Dan akan tampil seperti gambar di bawah ini, kemudian klik next.



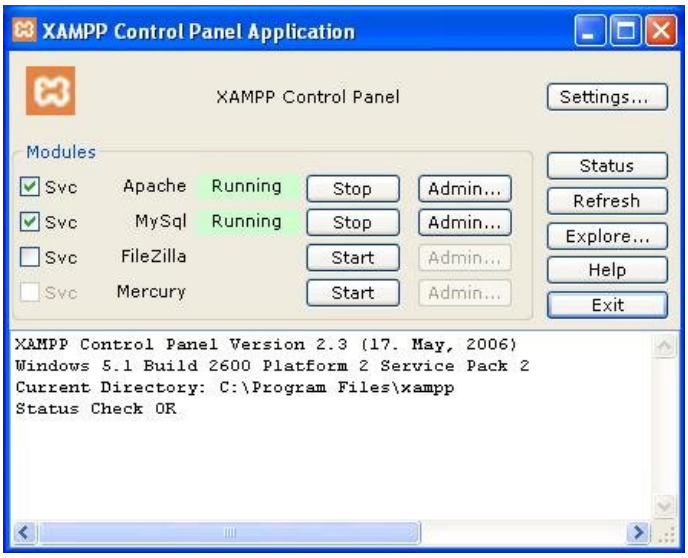
Gambar 5.2 Destination Folder xampp

5. Berikan tanda check pada check box seperti yang terlihat pada gambar di bawah.



Gambar 5.3. Check box installer

6. Klik tombol “Install”.
7. Jika sudah terinstal dengan baik maka akan nampak aplikasi windows seperti pada gambar di bawah ini.



Gambar 5.4 Running XAMPP

8. Sekarang jalankan browser anda pada firefox atau ie. Ketikkan url http://localhost/. Kemudian pilih bahasa yang anda bisa. Maka akan tampak seperti gambar di bawah.



Gambar 5.5 Pemanggilan localhost/xampp

5.2.7. Penggunaan program

1. Form Login Admin

Implementasi tampilan form validasi user merupakan tampilan pertama pada saat program dijalankan. Form ini digunakan sebagai saran keamanan bagi sistem untuk memberikan hak akses bagi yang berhak saja.



Gambar 5.6 Tampilan Form Login Admin

User yang terdapat dalam sistem ini yaitu admin. Pada saat masuk ke form login yang harus dilakukan adalah:

1. Input nama user
2. Input password
3. Kemudian tekan OK

Jika terjadi kesalahan dalam pengisian password, maka akan muncul pesan.



Gambar 5.7 Form pesan jika *password* salah.

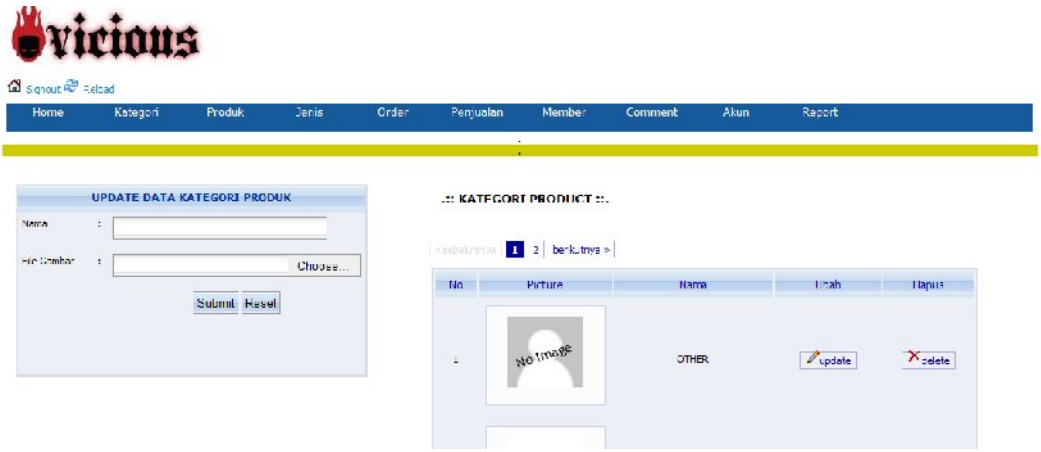
2. Form Utama



Gambar 5.8 Tampilan Form Utama Admin

3. Form Data Kategori

Form data Kategori yaitu penambahan jenis atau update data Kategori. Berikut ini gambar 5.9 merupakan tampilan form data Kategori :



Gambar 5.9 Form Data Kategori Admin

a. **submit**

Tombol submit ini berfungsi untuk menyimpan data produk yang sudah di inputkan ke dalam database.

b. reset

Tombol Reset berfungsi untuk mengosongkan data yang telah di input balik ke semula.

c. update

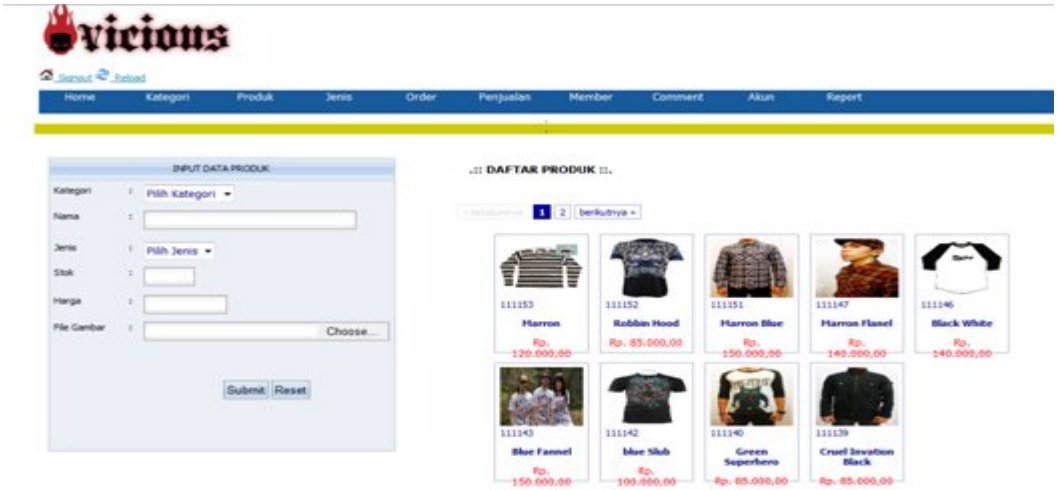
Tombol update berfungsi untuk mengubah data kategori yang telah ada.

d. delete

Tombol delete berfungsi untuk menghapus data kategori yang telah ada.

4. Form Data Produk

Form data Produk merupakan form input data dan juga berfungsi untuk menampilkan data. Berikut ini gambar 5.9 merupakan tampilan form data Kategori :



Gambar 5.10 Form Data Produk Admin

a. **Submit**

Tombol submit ini berfungsi untuk menyimpan data produk yang sudah di inputkan ke dalam database.

b. reset

Tombol reset berfungsi untuk mengosongkan data yang telah di input balik ke semula.

c. Update

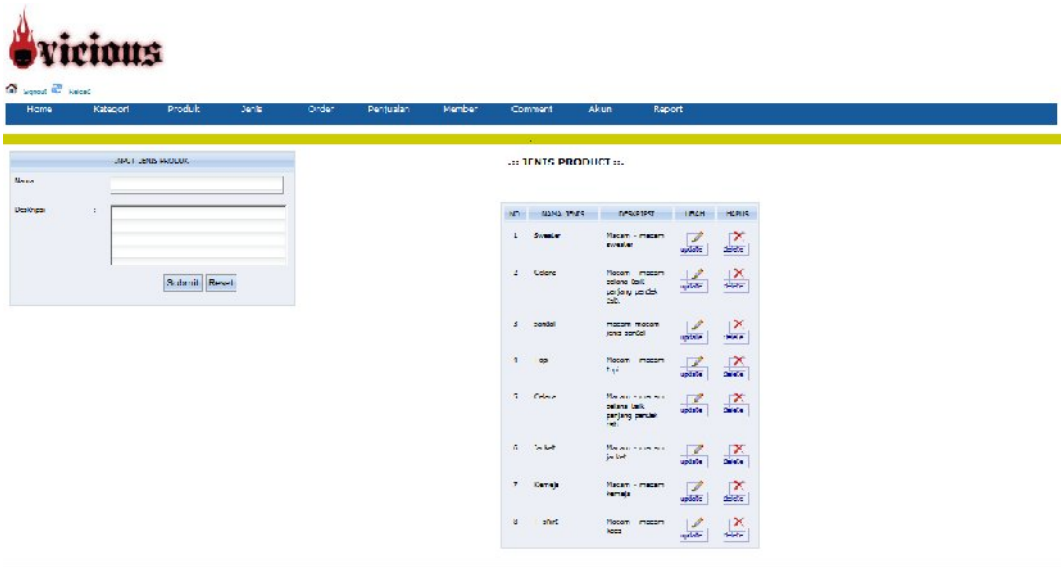
Tombol update berfungsi untuk mengubah data produk yang telah ada.

d. delete

Tombol delete berfungsi untuk menghapus data produk yang telah ada.

5. Form Data Jenis

Form data Jenis yaitu form input jenis barang yang ada pada web.



Gambar 5.11 Form Data Jenis Admin

a. **Submit**

Tombol submit ini berfungsi untuk menyimpan data produk yang sudah di inputkan ke dalam database.

b. reset

Tombol reset berfungsi untuk mengosongkan data yang telah di input balik ke semula.

c. Update

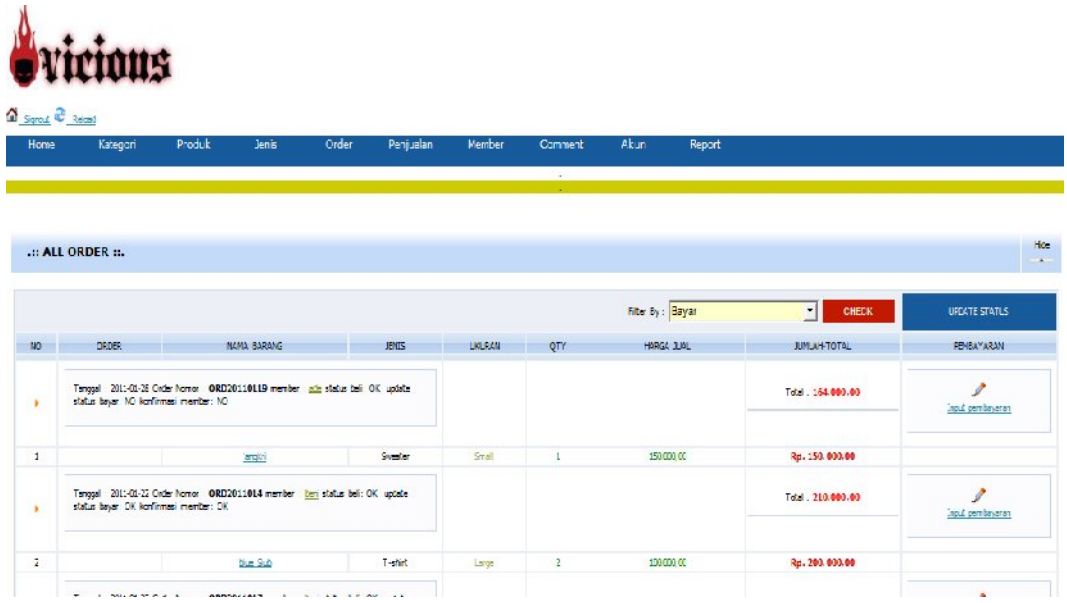
Tombol update berfungsi untuk mengubah data produk yang telah ada.

d. delete

Tombol delete berfungsi untuk menghapus data produk yang telah ada.

6. Form data Order.

Form data order berfungsi untuk melihat data order konsumen, baik dari data. Admin dapat mengubah status order.



Gambar 5.12 Form Data Order Admin

a. **check**

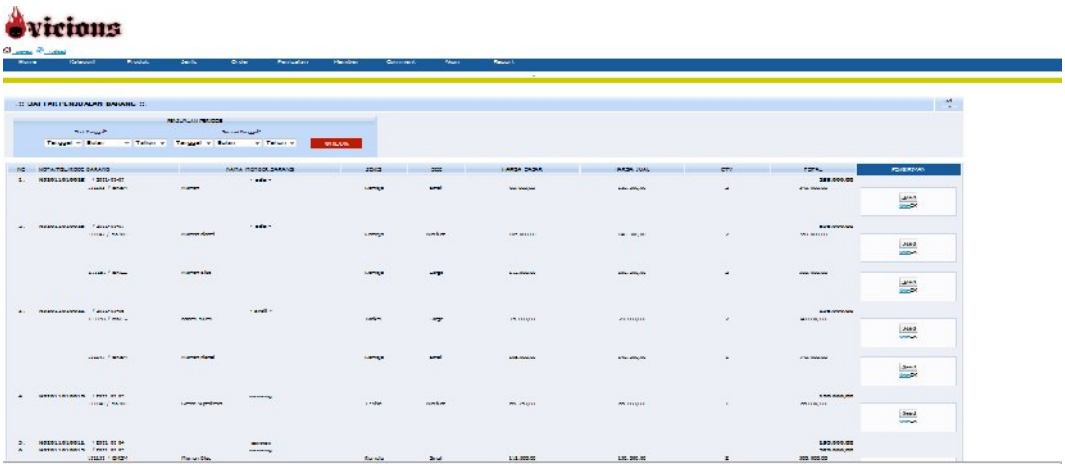
Tombol check ini berfungsi untuk memfilter data yang sudah melakukan konfirmasi pembayaran atau belum.

b. Input pembayaran

Tombol input pembayaran berfungsi untuk menginput data yang sudah melakukan konfirmasi pembayaran.

7. Form data Penjualan

Form data penjualan berfungsi untuk melihat data penjualan yang udah dikirim atau belum. Admin dapat mengubah status kirim menjadi ok.



Gambar 5.13 Form Data Penjualan Admin

a. **check**

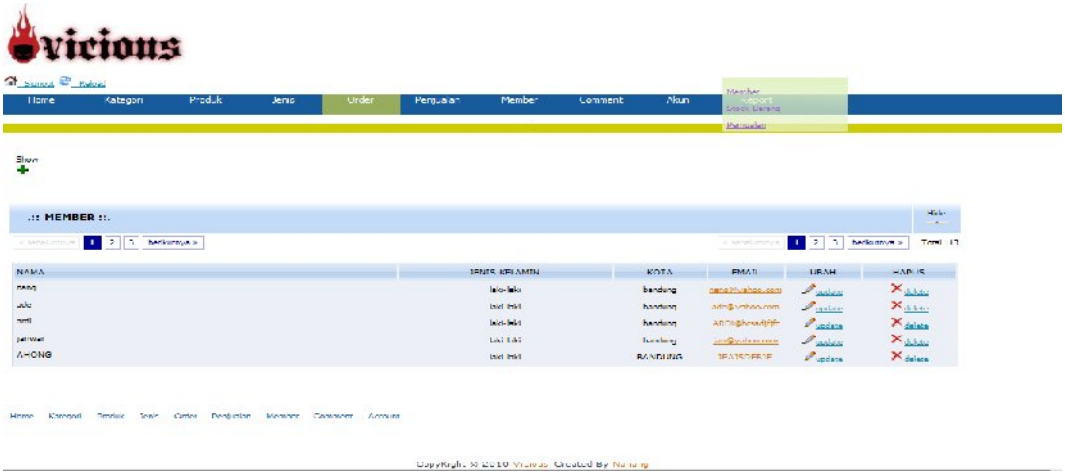
Tombol check ini berfungsi untuk memfilter data penjualan perhari atau perbulan.

b. ketik sent

Tombol sent berfungsi untuk merubah data yang belum dikirim menjadi sudah dikirim.

8. Form Data Member

Form data member merupakan form tampilan semua user yang sudah jadi member.



Gambar 5.14 Form Data member di Admin

a. **Submit**

Tombol submit ini berfungsi untuk menyimpan data member.

b. reset

Tombol reset berfungsi untuk mengosongkan data yang telah di input balik ke semula.

c. Update

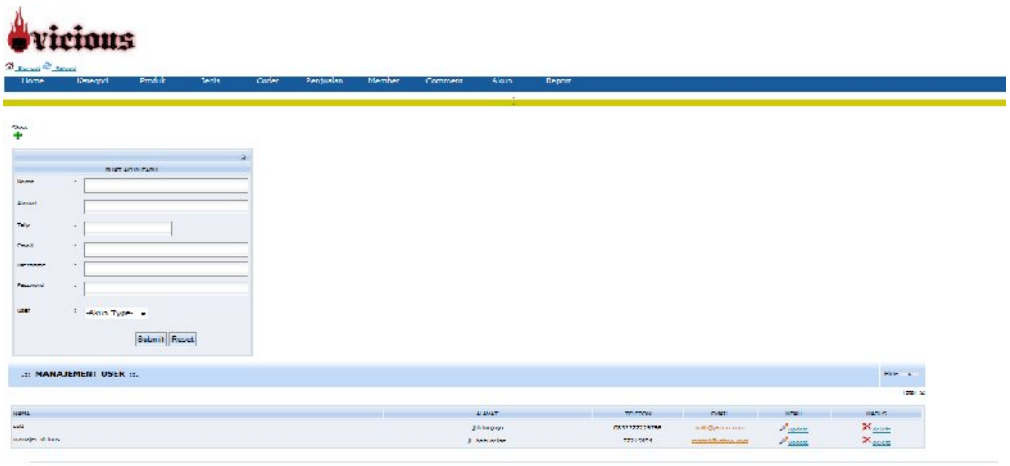
Tombol update berfungsi untuk mengubah data member yang sudah ada.

d. delete

Tombol delete berfungsi untuk menghapus data member yang sudah ada.

9. Form Data Akun

Form data akun merupakan form pengaturan data admin yang sudah jadi pegawai.



Gambar 5.15 Form Data Akun di Admin

a. **Submit**

Tombol submit ini berfungsi untuk menyimpan data baru admin.

b. reset

Tombol reset berfungsi untuk mengosongkan data yang telah di input balik ke semula.

c. Update

Tombol update berfungsi untuk mengubah data admin yang telah ada.

d. delete

Tombol delete berfungsi untuk menghapus data admin yang telah ada.

10. Form Data Report Stock Barang



LAPORAN STOK BARANG

Distro Vicious Clothing Company
Address : Jl. Bahureksa No.1 Bandung
Contact Person : (022)6018943

NO	KODE BARANG	NAMA	SIZE	HARGA BELI	HARGA JUAL	STOK BARANG
1.	111112	- Angry Mickey / T-shirt				
	BKSM		Small	125.000,00	150.000,00	1
	BKMD		Medium	125.000,00	150.000,00	4
	BKLG		LARGE	125.000,00	150.000,00	2
2	111114	- Black White / T-shirt				
	BKSM		Small	105.000,00	140.000,00	3
	BKMD		Medium	105.000,00	140.000,00	8
	BKLG		Large	105.000,00	140.000,00	2
	BKXL		Xtra Large	105.000,00	140.000,00	6
3	111113	- Blue Angel / T-shirt				
	BKSM		Small	112.500,00	150.000,00	0
	BKMD		Medium	112.500,00	150.000,00	5
	BKLG		Large	112.500,00	150.000,00	0
4.	111112	- Blue Skull / T-shirt				
	BKSM		Small	75.000,00	100.000,00	0
	BKMD		Medium	75.000,00	100.000,00	1
	BKLG		Large	75.000,00	100.000,00	3
	BKXL		Xtra Large	75.000,00	100.000,00	2
5	111119	- Cruel Invation Black / Jacket				
	BKSM		Small	63.750,00	85.000,00	1
	BKLG		Large	63.750,00	85.000,00	5
6.	111113	Fairies Blues / Jacket				
	BKMD		Medium	15.000,00	20.000,00	2
	BKLG		Large	15.000,00	20.000,00	3

Gambar 5.16 Form Data Report stock barang di Admin

11. Form Report Data Member



LAPORAN DATA MEMBER

Distro Vicious Clothing Company
Address : Jl. Bahureksa No.1 Bandung
Contact Person : (022)6018943

Total : 13

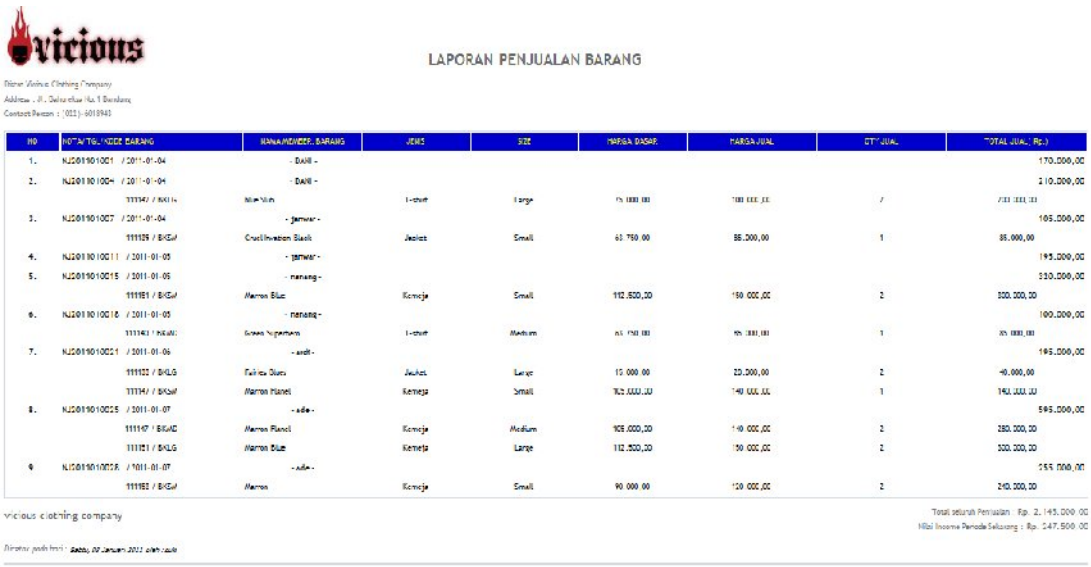
NO	NAMA	JENIS KELAMIN	TGL LAHIR	ALAMAT	KOTA	KODE POS	TELP	EMAIL
1	nang	laki-laki	0000-00-00	jl. margahayu selatan	bandung	b do hdbdh	878798 u908908	nang@yahoo.com
2	ace	laki-laki	0000-00-00	jl. sukasari	bandung	7668687	87676876	ace@yahoo.com
3	ardi	laki-laki	0000-00-00	pelesiran	bandung	765768	5436457658645	ARDI@bcsadjfjfo
4	janwar	laki-laki	1986-08-14	kasang setra	bandung	654664	435445466	jan@yahoo.com
5	AHONG	laki-laki	0000-00-00	JL.TEGAL	BANDUNG	1432454	8686775676GR	JFAJSDFBJE
6	DANI	laki-laki	1978-09-09	jhjk	jkj	454646	76797797	nannag@gmail.com
7	narang	laki-laki	1987-02-09	jl. ...	jambi	40977	C9C8C8C5	narang@yahoo.com
8	hilman	laki-laki	0000-00-00	tikukur	bandung	7264542	97867567S2	hilman@yahoo.com
9	janwar	laki-laki	1987-12-15	kasangsetra	bandung	4017221	0857954412221	janwar@yahoo.com
10	a	laki-laki	2010-02-11	a	h	40222	4444	e@gmail.com
11	narang	laki-laki	1988-12-12	duka	subang	402145	2897753454	duka@yahoo.com
12	TES"	laki-laki	0000-00-00	FA	BANDUNG	54678	45878	E@RE.COM
13	echo	laki-laki	0000-00-00	kcf	ba	4905656	2343	facfcdf

vicious clothing company

Dicetak pada hari : Selasa, 08 Januari 2013 Jam : 21:16

Gambar 5.17 Form Report Data Member

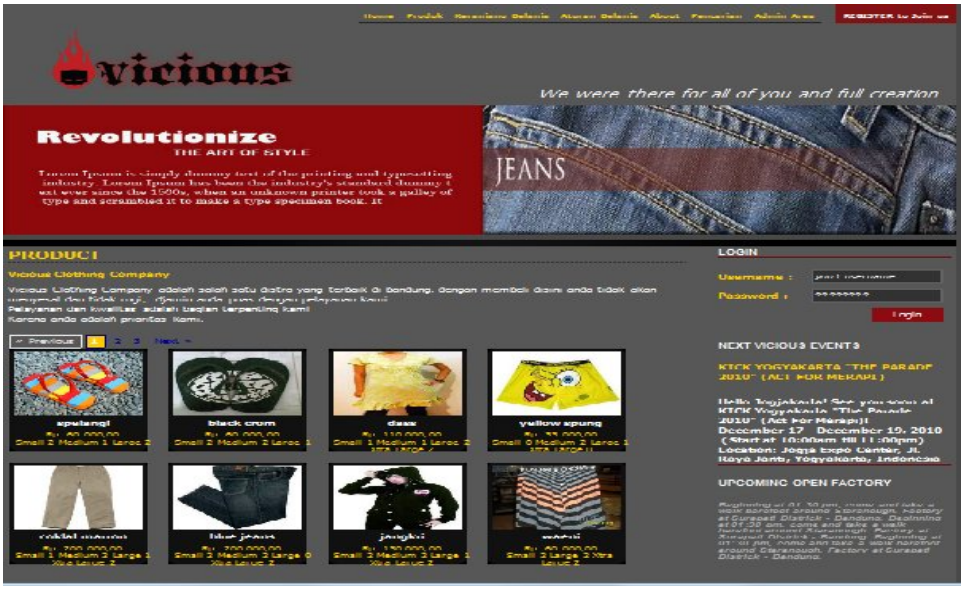
12. Form Report Data Penjualan



Gambar 5.18 Form Report Data Penjualan

13. Form Halaman User

Form halaman user ini merupakan halaman utama untuk user pelanggan.



Gambar 5.19 Form Halaman User

a. User bisa melihat – lihat produk terlebih dahulu jika ada yang cocok dan kemudian mau pesen harus daftar menjadi member dahulu di atas sebelah kanan klik register.
b. Setelah masuk form register user menginput data diri kemudian klik tombol save.
c. Setelah ada pemberitahuan sukses berarti user sukses menjadi member dan bisa langsung melakukan login terlebih dahulu jika ingin melakukan pemesanan.

14. Form Produk dan detail Produk

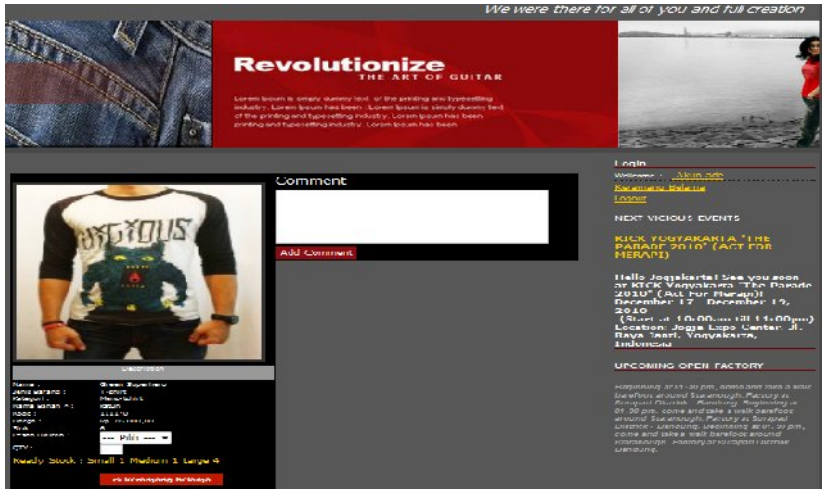
Form data produk dan detail produk ini merupakan gambaran produk yang akan di pilih oleh user atau pelanggan.



Gambar 5.20 Form Data Produk

a. Setelah melakukan login member bisa langsung memilih produk yang di inginkan jika ada yang cocok member bisa langsung klik gambar produk yang di inginkan.

b. Kemudian muncul detail barang seperti di bawah ini.



Gambar 5.21 Form Detail Produk

a. Kemudian member memilih ukuran yang berada dalam tombol combobox

b. Setelah itu member menentukan jumlah yang akan dibeli

c. Kalau sudah selesai member bisa langsung klik tombol keranjang belanja.

d. Jika simpan data sukses member harus cek barang yang ada di dalam keranjang belanja

15. Form Keranjang Belanja

Form keranjang belanja ini berfungsi untuk menyimpan produk sementara yang akan di pesan oleh user.



Gambar 5.22 Form Keranjang Belanja

a. delete

Tombol delete berfungsi untuk menghapus data yang telah di input.

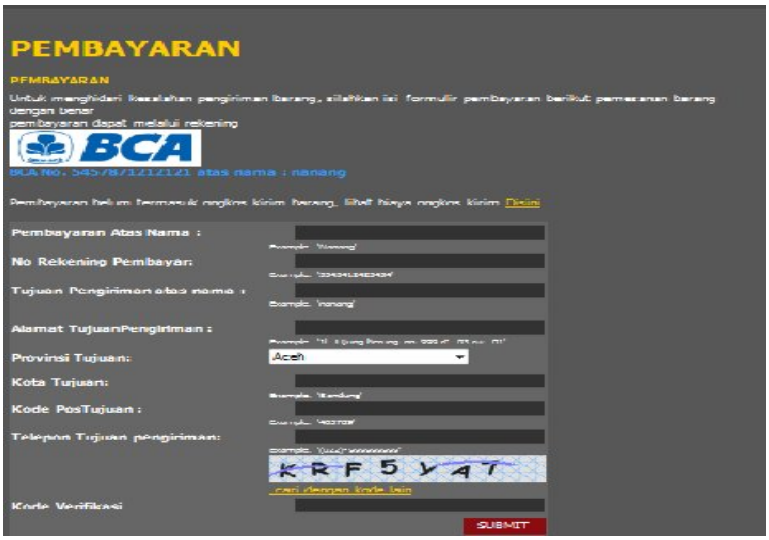
b. batal

Tombol batal berfungsi untuk membatalkan data pemesanan.

jika sudah benar barang yang tadi dipesan ada member bisa langsung klik tombol data keranjang belanja yang berwarna merah kemudian muncul form pembayaran untuk di isi oleh member.

16. Form Data Pembayaran

Form data pembayaran ini berisi tentang data pembayaran yang akan di input oleh oleh pelanggan. Pertama pelanggan mengisi data informasi pengiriman, kemudian pelanggan mengisi data informasi pembayaran.



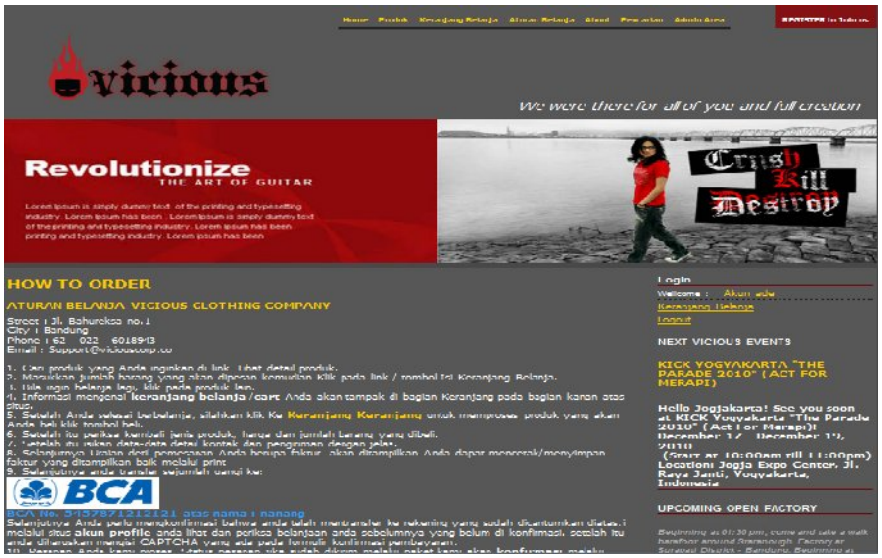
Gambar 5.23 Form Data Pembayaran

a. Kemudian member mengisi data form pembayaran.

b. Setelah itu member klik tombol data submit untuk melakukan proses terakhir pemesanan kemudian muncul bukti pemesanan untuk konsumen.

17. Form Aturan Belanja

Form aturan belanja yaitu aturan atau cara memesan produk



Gambar 5.24 Form Data Aturan Belanj

18. Form Akhir Laporan Data Pemesanan untuk Pelanggan



BUKTI PEMESANAN BARANG

Order Nomor : ORD20110119 member : ade
status beli: OK
status bayar: NO
status kirim: NO

NO	BARANG	JENIS	UKURAN	HARGA	QTY	JUMLAH
1	jangkri	Sweater	Small	150.000,00	1	150.000,00

jumlah item : 1
jumlah barang : 2
berat : 1 Kg
TOTAL. 150.000,00
Biaya Kirim : 14000
Total yang harus dibayar ke rekening sebesar :164.000,00

pembayaran dapat melalui rekening

BCA No. 5457871212121 atas nama : nanang

dicetak pada hari : Sabtu, 29 Januari 2011

Gambar 5.25 Form Akhir Bukti Data Pemesanan User