BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. **Implementasi**

Implementasi *prototype* sistem informasi penjualan dan pembelian Distro Superior dilakukan menggunakan bahasa dan lingkungan pemrograman *PHP*, basis data yang digunakan ialah *MySQL*. Implementasi dan pengujian sepenuhnya hanya dilakukan di perangkat *PC* (*personal computer*) dengan sistem operasi *Microsoft Windows XP*.

5.1.1. **Batasan Implementasi**

Dalam mengimplementasikan perangkat lunak ini ada beberapa hal yang menjadi batasan implementasi, yaitu :

1. Hanya pembeli yang dapat melakukan transaksi, dan memesan barang yang akan dibeli.*user* dapat dilihat di*website*nya tentang informasi-informasi yang disediakan seperti informasi dan kontak perusahaan, promosi produk

2. Administrator melakukan isi data promosi produk, mengirim informasi produk langsung kepelanggan yang ingin membeli, konfirmasi pembayaran, mengelola dan mengolah data yang terdapat pada sistem.

3. Admin melakukan kelola data admin, dan data forum diskusi yang tersedia serta mengolah laporan-laporan yang akan dicetak.

5.1.2. **Implementasi Perangkat Lunak**

Adapun beberapa perangkat lunak yang harus disiapkan dalam pembangunan aplikasi ini yaitu melakukan beberapa proses instalasi beberapa *development tools* yang digunakan, antara lain:

1. *Xampp*

   Penulis menggunakan *xampp web server* untuk mengolah kode *PHP* atau *HTML* yang akan dikirim ke *client.* Menggunakan *Apachefriends Xampp* 3.4.5!

2. *Database Server*

   Database server yang dugunakan oleh penulis adalah *server database MySQL* untuk menyimpan sumber data aplikasi.

3. *Macromedia Dreamweaver 8*

   Untuk membuat aplikasi penjualan dan pembelian penulis menggunakan *Macromedia Dreamweaver 8* sebagai *script*.

4. *Mozzila Firefox*

   Mozzila Firefox, sebagai media untuk menampilkan program penjualan dan pembelian.

5.1.3. **Implementasi Perangkat Keras**

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

Admin :

1. *Processor* yang digunakan adalah *intel pentium 4 CPU 2.60 Ghz.* Rekomendasi minimum adalah processor intel pentium 4 *CPU* 1.60 Ghz.

2. *Harddisk* terpasang 40 GB, rekomendasi minimun adalah 20 GB.

3. *Memory* terpasang 256 MB, rekomendasi minimum adalah 8 MB.

4. *VGA Card* terpasang 32 MB, rekomendasi minimum adalah 1 MB.

5. *LAN Card*

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

User :

1. *Processor* yang digunakan adalah *intel pentium 4 CPU 2.60 Ghz.*

Rekomendasi minimum adalah processor intel pentium 4 *CPU* 1.60 Ghz.

2. *Harddisk* terpasang 40 GB, rekomendasi minimun adalah 10 GB.
3. *Memory* terpasang 256 MB, rekomendasi minimum adalah 8 MB.
4. *VGA Card* terpasang 32 MB, rekomendasi minimum adalah 1 MB.
5. *LAN Card.Mouse, Keyboard,* dan *Monitor* sebagai peralatan antarmuka.

5.1.4. **Implementasi Basis Data**

Pembuatan basis data dilakukan dengan menggunakan *MySQL* . Implementasi basis data dalam bahasa *SQL* adalah sebagai berikut :

```
--

-- Table structure for table `admin`

--


CREATE TABLE IF NOT EXISTS `admin` (
  `username` varchar(25) NOT NULL,
  `password` varchar(35) NOT NULL,
  `nama_lengkap` varchar(35) NOT NULL,
  `email` varchar(30) NOT NULL,
  `no_telp` varchar(15) NOT NULL,
  `blokir` enum('red','yellow','green') NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;


--

-- Dumping data for table `admin`

--
```

```
INSERT INTO `admin` (`username`, `password`, `nama_lengkap`, `email`, `no_telp`, `blokir`) VALUES
('admin', '21232f297a57a5a743894a0e4a801fc3', 'Agus Sumpena', 'admin@jersey.com', '0226000999', 'green');

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

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

CREATE TABLE IF NOT EXISTS `barang` (
  `kode_barang` varchar(6) NOT NULL,
  `nama_barang` varchar(25) NOT NULL,
  `id_kategori` varchar(2) NOT NULL,
  `tipe` enum('pria','wanita') NOT NULL,
  `deskripsi` text NOT NULL,
  `harga_jual` decimal(10,0) NOT NULL,
  `gambar` varchar(225) NOT NULL,
```

```
  PRIMARY KEY (`kode_barang`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;



--

-- Dumping data for table `barang`

--



INSERT INTO `barang` (`kode_barang`, `nama_barang`, `id_kategori`, `tipe`, `deskripsi`, `harga_jual`, `gambar`) VALUES

('TS001', 'Baju Trendy', 'TS', 'pria', 'Baju trendy yang modis sangat nyaman karena terbuat dari bahan berkualitas', 25000, 'TS001.jpg'),

('TS002', 'Baju Trendy', 'TS', 'pria', 'Baju trendy yang modis sangat nyaman karena terbuat dari bahan berkualitas', 750000, 'TS002.jpg'),



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

--

-- Table structure for table `info`

--
```

```
CREATE TABLE IF NOT EXISTS `info` (
  `tabel` varchar(25) NOT NULL,
  `jumlah` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

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

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

CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` varchar(2) NOT NULL,
  `jenis` varchar(25) NOT NULL,
  PRIMARY KEY (`id_kategori`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
```

```
-- Dumping data for table `kategori`

--


INSERT INTO `kategori` (`id_kategori`, `jenis`) VALUES

('TS', 'T-Shirts');


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


--

-- Table structure for table `member`

--


CREATE TABLE IF NOT EXISTS `member` (

  `username` varchar(35) NOT NULL,

  `password` varchar(35) NOT NULL,

  `nama_lengkap` varchar(35) NOT NULL,

  `tanggal_lahir` varchar(20) NOT NULL,

  `email` varchar(25) NOT NULL,
```

```
  `no_telp` varchar(20) NOT NULL,
  `alamat` text NOT NULL,
  `propinsi` varchar(35) NOT NULL,
  `kota` varchar(255) NOT NULL,
  `kodepos` varchar(255) NOT NULL,
  `konfirmasi` enum('KONFIRMASI','BELUM') NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `member`
--

INSERT INTO `member` (`username`, `password`, `nama_lengkap`, `tanggal_lahir`, `email`, `no_telp`, `alamat`, `propinsi`, `kota`, `kodepos`, `konfirmasi`) VALUES
('jojon@bin.ujo', '5cbe716cf4c59cf4b48e8060211be05b', 'jojon bin ujo', '1988-03-05', 'jojon@bin.ujo', '02354584796', 'Alamat palsunya ayu ting ting', 'Bali', 'Denpasar', '41320', 'BELUM'),
```

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

--

-- Table structure for table `pembayaran`

--

CREATE TABLE IF NOT EXISTS `pembayaran` (
  `idpembayaran` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(35) NOT NULL,
  `no_telp` varchar(20) NOT NULL,
  `idtransaksi` varchar(100) NOT NULL,
  `an_transfer` varchar(35) NOT NULL,
  `ke_bank` varchar(200) NOT NULL,
  `tanggal_transfer` datetime NOT NULL,
  `jumlah` decimal(10,0) NOT NULL,
  `status` enum('KONFIRMASI','BELUM') NOT NULL DEFAULT 'BELUM',
  PRIMARY KEY (`idpembayaran`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```

```
--

-- Dumping data for table `pembayaran`

--


INSERT INTO `pembayaran` (`idpembayaran`, `username`, `no_telp`, `idtransaksi`, `an_transfer`, `ke_bank`, `tanggal_transfer`, `jumlah`, `status`) VALUES

(4, 'ceceb@gorbaceb.com', '0228221114', '20130528212828', 'ceceb gorgaceb', 'MANDIRI-143.0010.959.201', '2013-06-12 08:00:00', 345000, 'KONFIRMASI'),


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


--

-- Table structure for table `pembelian`

--


CREATE TABLE IF NOT EXISTS `pembelian` (

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

```
  `username` varchar(25) NOT NULL,
  `alamat_tujuan` text NOT NULL,
  `propinsi` varchar(35) NOT NULL,
  `ongkir` decimal(10,0) NOT NULL,
  `tanggal` date NOT NULL DEFAULT '2012-06-01',
  `bayar` enum('SUDAH','BELUM') NOT NULL,
  `status` enum('SUDAH','BELUM') NOT NULL,
  PRIMARY KEY (`idtransaksi`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

--
-- Dumping data for table `pembelian`
--

INSERT INTO `pembelian` (`idtransaksi`, `username`, `alamat_tujuan`,
`propinsi`, `ongkir`, `tanggal`, `bayar`, `status`) VALUES
('20130528212828', 'ceceb@gorbaceb.com', 'alamat palsu', 'Jawa Barat', 10000,
'2013-05-28', 'SUDAH', 'SUDAH'),
```

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

--

-- Table structure for table `pesan`

--

CREATE TABLE IF NOT EXISTS `pesan` (
  `nomor` int(11) NOT NULL AUTO_INCREMENT,
  `idtransaksi` varchar(100) NOT NULL,
  `idproduk` varchar(6) NOT NULL,
  `jumlah` int(11) NOT NULL,
  `ukuran` enum('S','M','X','XL') NOT NULL,
  `harga` decimal(10,0) NOT NULL,
  PRIMARY KEY (`nomor`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=31 ;

--

-- Dumping data for table `pesan`
```

```
--


INSERT INTO `pesan` (`nomor`, `idtransaksi`, `idproduk`, `jumlah`, `ukuran`,
`harga`) VALUES

(7, '20130528212828', 'TS006', 1, 'S', 85000),


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


--

-- Table structure for table `stok`

--


CREATE TABLE IF NOT EXISTS `stok` (

  `no` int(11) NOT NULL AUTO_INCREMENT,

  `kode_barang` varchar(6) NOT NULL,

  `ukuran` enum('S','M','X','XL') NOT NULL,

  `stok` int(11) NOT NULL DEFAULT '100',

  PRIMARY KEY (`no`)
```

```
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=145
;


--
-- Dumping data for table `stok`
--


INSERT INTO `stok` (`no`, `kode_barang`, `ukuran`, `stok`) VALUES
(1, 'TS001', 'S', 0),
(2, 'TS001', 'M', 30),
(3, 'TS001', 'X', 38),
(4, 'TS001', 'XL', 135),


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


--
-- Table structure for table `tarif`
--
```

```
CREATE TABLE IF NOT EXISTS `tarif` (
  `id_propinsi` int(11) NOT NULL AUTO_INCREMENT,
  `propinsi` varchar(30) NOT NULL,
  `tarif` decimal(10,0) NOT NULL,
  `sampai` int(11) NOT NULL,
  PRIMARY KEY (`id_propinsi`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=28 ;

--
-- Dumping data for table `tarif`
--

INSERT INTO `tarif` (`id_propinsi`, `propinsi`, `tarif`, `sampai`) VALUES
(1, 'D.I Aceh', 26250, 2),
(2, 'Sumatera Utara', 19200, 1),

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

```
--

-- Table structure for table `testimonial`

--


CREATE TABLE IF NOT EXISTS `testimonial` (
  `nomor` int(11) NOT NULL AUTO_INCREMENT,
  `tanggal` varchar(25) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `email` varchar(30) NOT NULL,
  `kota` varchar(30) NOT NULL,
  `testi` text NOT NULL,
  `view` enum('YES','NO') NOT NULL,
  PRIMARY KEY (`nomor`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;


--

-- Dumping data for table `testimonial`
```

```
--


INSERT INTO `testimonial` (`nomor`, `tanggal`, `nama`, `email`, `kota`, `testi`,
`view`) VALUES

 (8, '15 May 2013', 'member', 'member@yahoo.com', 'bandung', 'kirim barangnya
ya mas', 'YES');


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


--

-- Table structure for table `ukuran`

--


CREATE TABLE IF NOT EXISTS `ukuran` (

  `id_kategori` varchar(6) NOT NULL,

  `ukuran` enum('S','M','X','XL') NOT NULL

) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

5.1.5. **Implementasi Antar Muka**

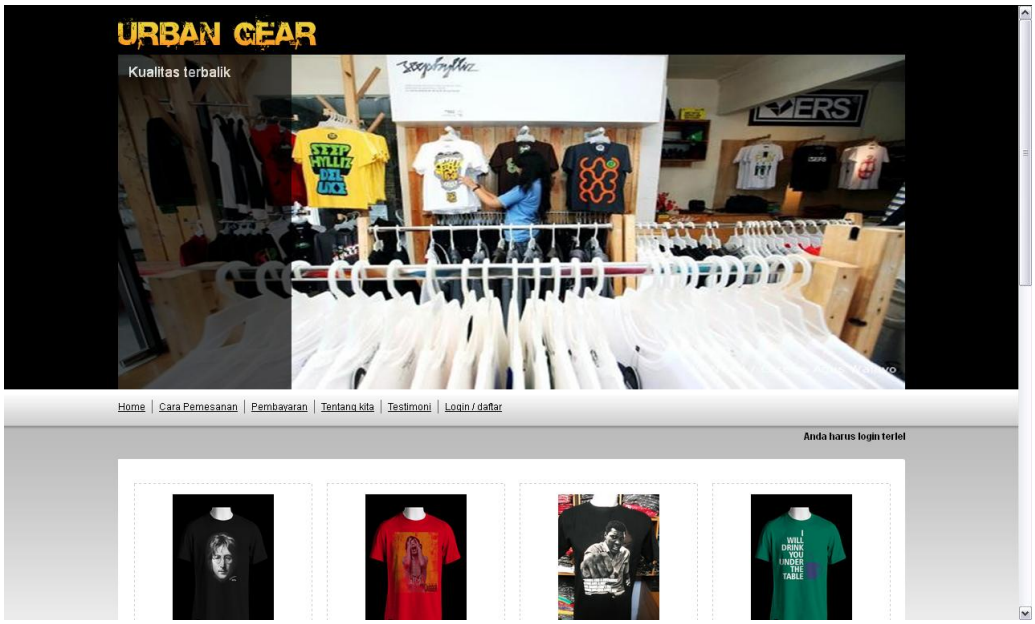
Setelah kita melihat ada kategori apa saja yang dijual,kita bisa melihat data pembeli yang sudah mengorder produk yang dijual,dengan melihat tampilan *Order*.Dalam *PHP* implementasi antarmuka dilakukan dengan sebuah *webpage* yang berekstensi *PHP*. Setiap halaman dalam perangkat lunak dibuat *file* program yang ditulis dengan berekstensi *PHP*. Berikut ini adalah implementasi dari setiap halaman yang dibuat dan dibedakan berdasarkan *use*r atau pengguna dan admin.

5.1.5.1. **Implementasi Halaman Utama**

Implementasi antarmuka dilakukan dengan setiap halaman program yang dibuat dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka yang dibuat :

1. Halaman Utama Tampilan untuk Konsumen

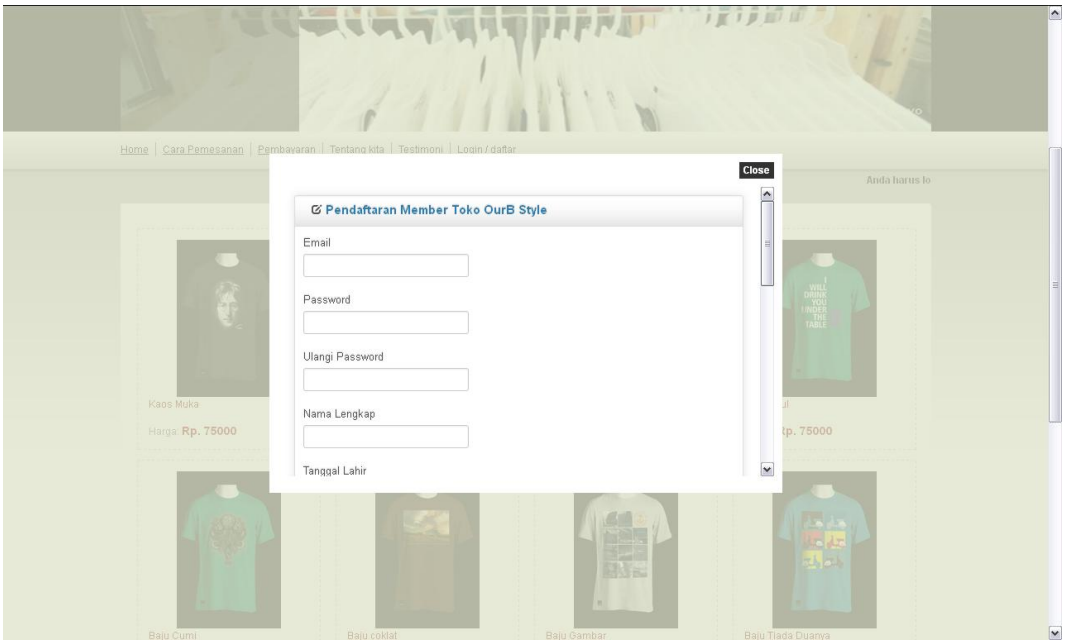
Halaman ini merupakan tampilan utama ketika konsumen membuka aplikasi Toko OurB Style.



Gambar 5.1 Halaman Tampilan Utama untuk Konsumen

2. Halaman Register untuk Konsumen

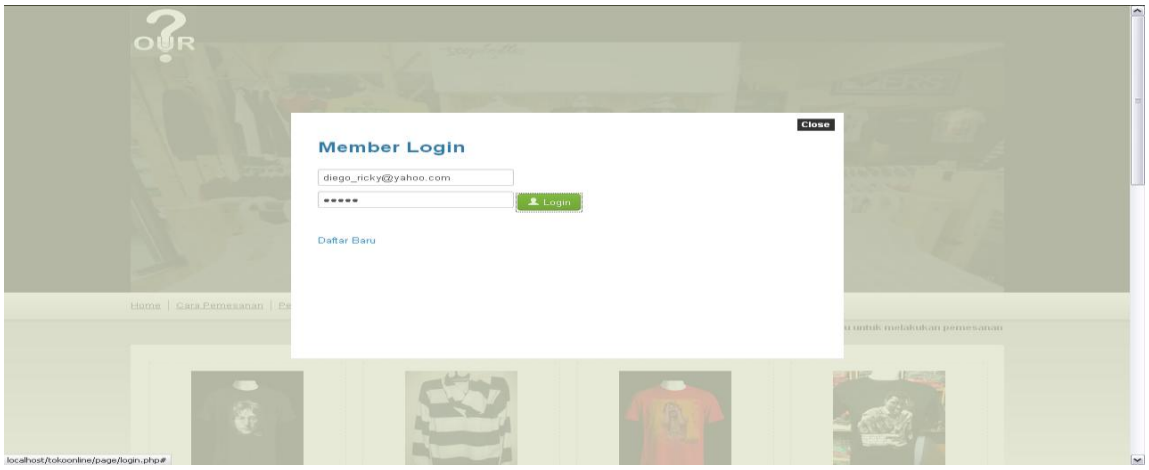
Halaman ini merupakan tampilan form untuk konsumen yang belum terdaftar dalam member konsumen Butik Shop, konsumen perlu menginputkan data-data konsumen sesuai dengan kebutuhan yang telah ditentukan.



Gambar 5.2 Halaman Tampilan Form Registrasi untuk Konsumen

3. Halaman Login untuk Konsumen

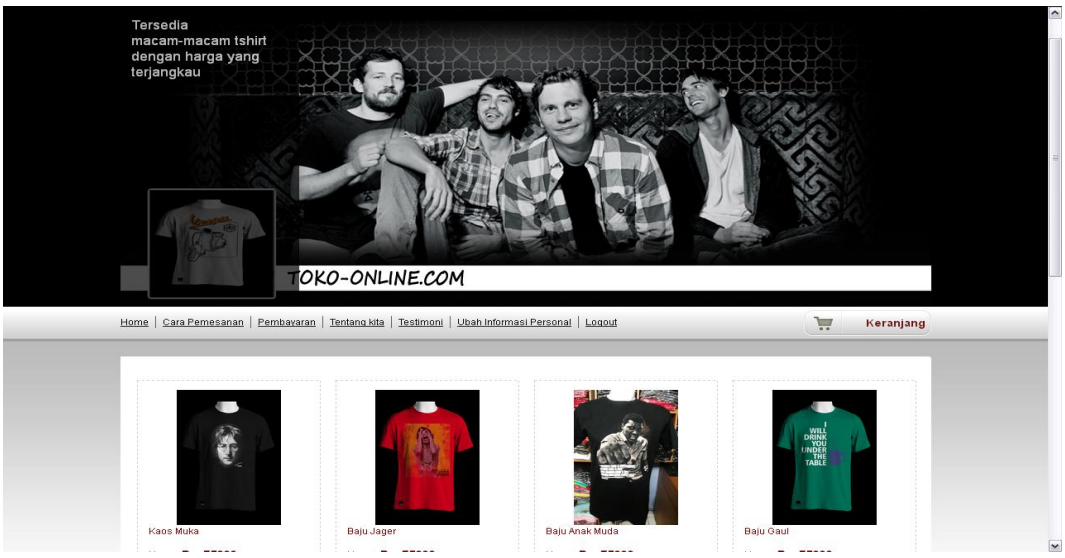
Halaman ini merupakan tampilan ketika konsumen akan melakukan *login* konsumen hanya perlu menginputkan data username dan password yang telah terdaftar dalam database.



Gambar 5.3 Halaman Tampilan Login Konsumen

4. Halaman ketika Konsumen Sukses *Login*

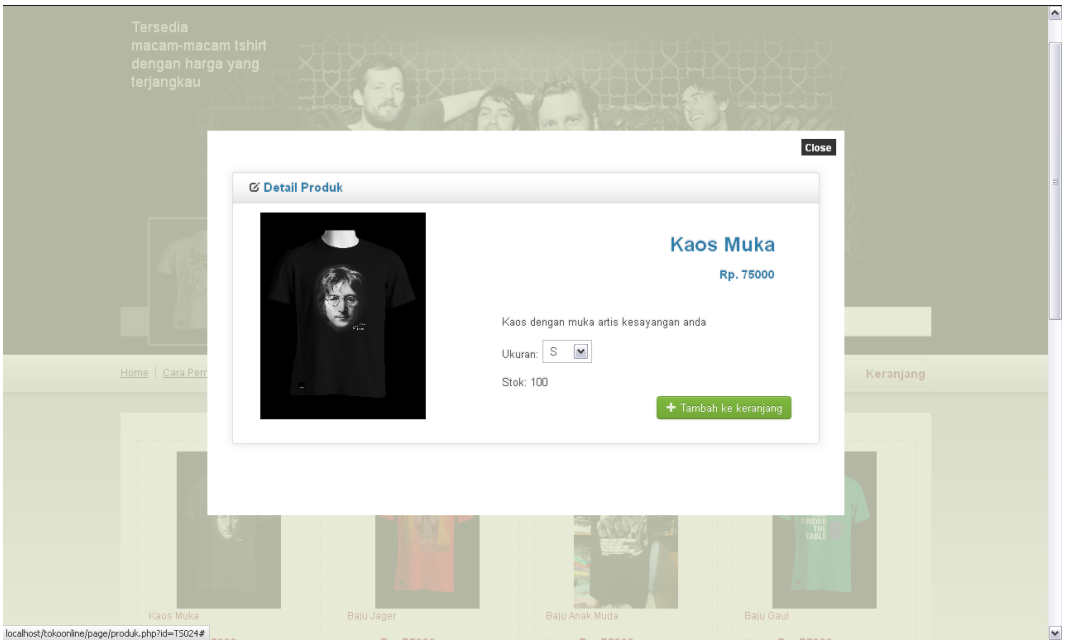
Halaman ini merupakan tampilan ketika konsumen berhasil melakukan proses *login*, maka akan muncul tampilan seperti berikut



Ga**mbar 5.4 Halaman Tampilan Sukses *Login* Konsumen**

5. Halaman Detail Produk

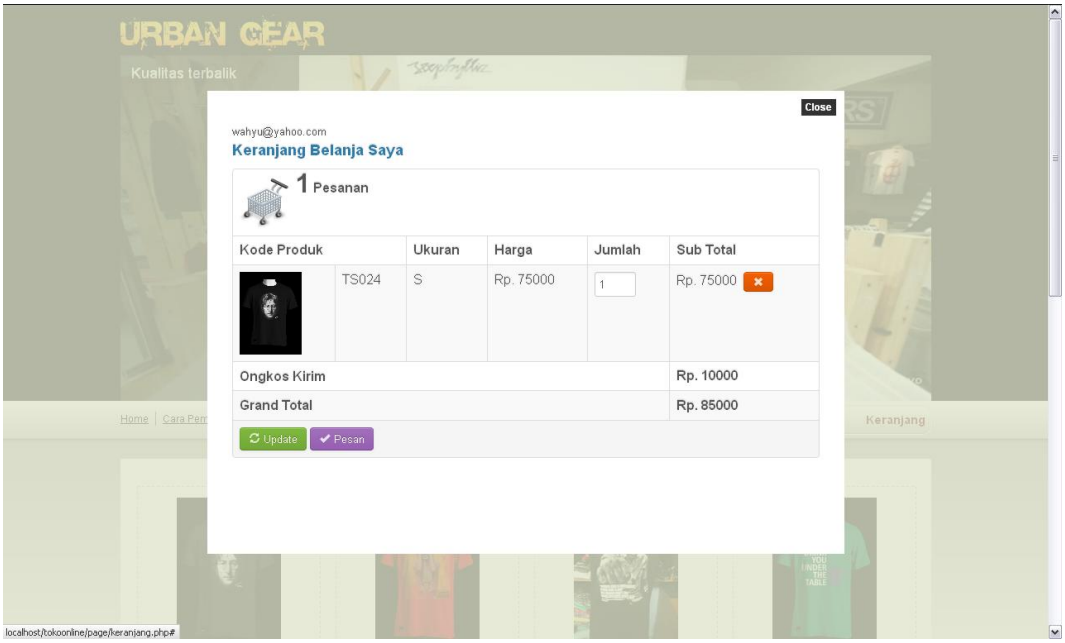
Halaman ini merupakan tampilan ketika konsumen ingin melihat detail produk



Ga**mbar 5.5 Halaman Tampilan Detail Produk**

6. Halaman keranjang belanja untuk Konsumen

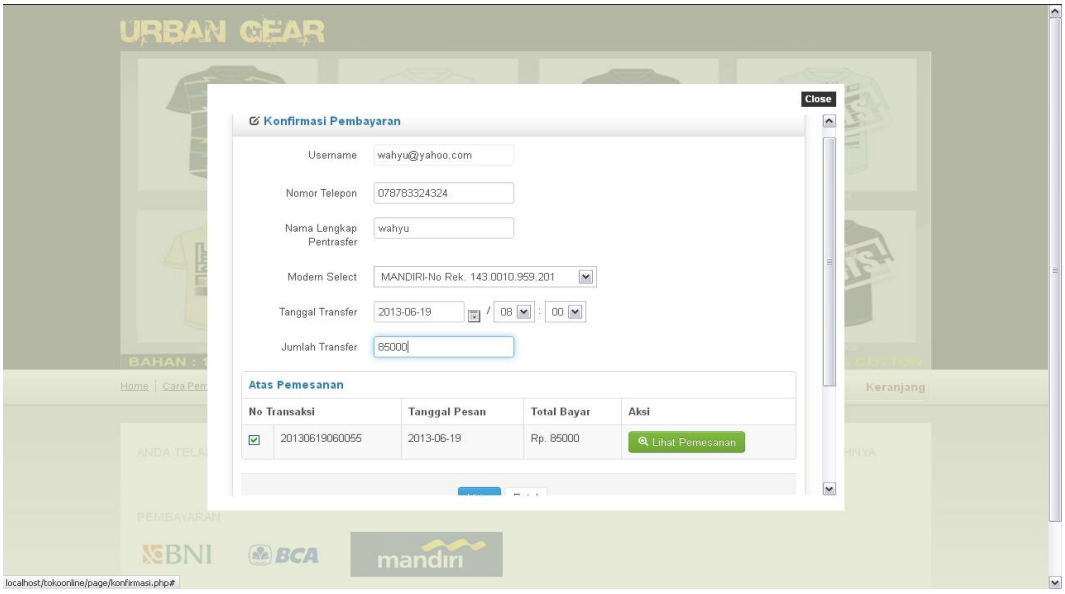
Halaman ini merupakan tampilan ketika konsumen telah memilih dan memesan produk, maka akan muncul tampilan seperti berikut



Gambar 5.6 Halaman Tampilan Keranjang Belanja

7. Halaman Konfirmasi Pembayaran untuk *Member*

Halaman ini merupakan halaman untuk *member* yang akan melakukan konfirmasi pembayaran.



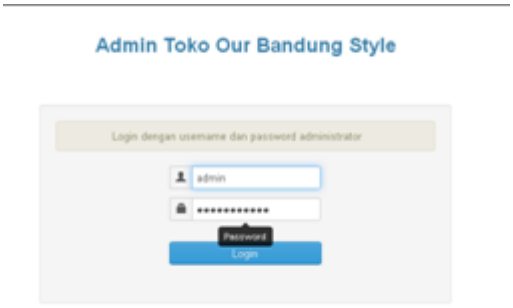
Gambar 5.7 Halaman Tampilan Konfirmasi Pembayaran

5.1.5.2. Penggunaan Program untuk Admin

Berikut ini langkah-langkah penggunaan program aplikasi *Butik shop* untuk admin.

1. **Halaman Login Admin**

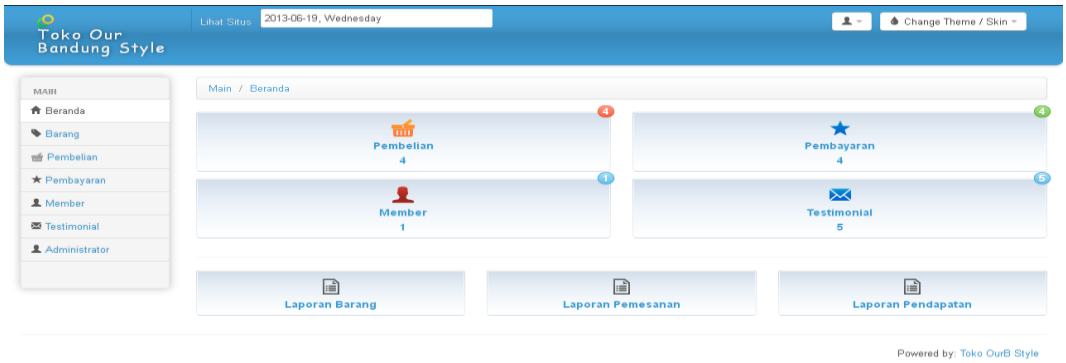
Admin hanya perlu menginputkan data username dan password yang telah tersimpan dalam database dengan benar.



Gambar 5.8 Halaman Login Admin

2. **Halaman Utama Admin setelah *Login***

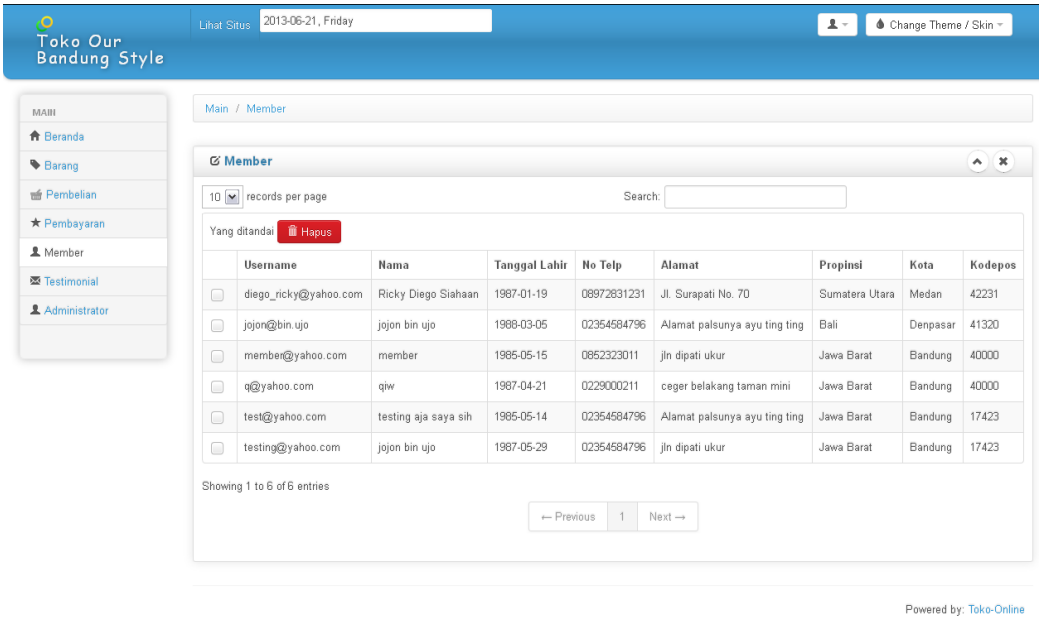
Halaman ini merupakan tampilan utama untuk admin ketika telah sukses melakukan *login*



Gambar 5.9 Halaman Tampilan Utama Admin

3. Halaman Tampilan Menu Member

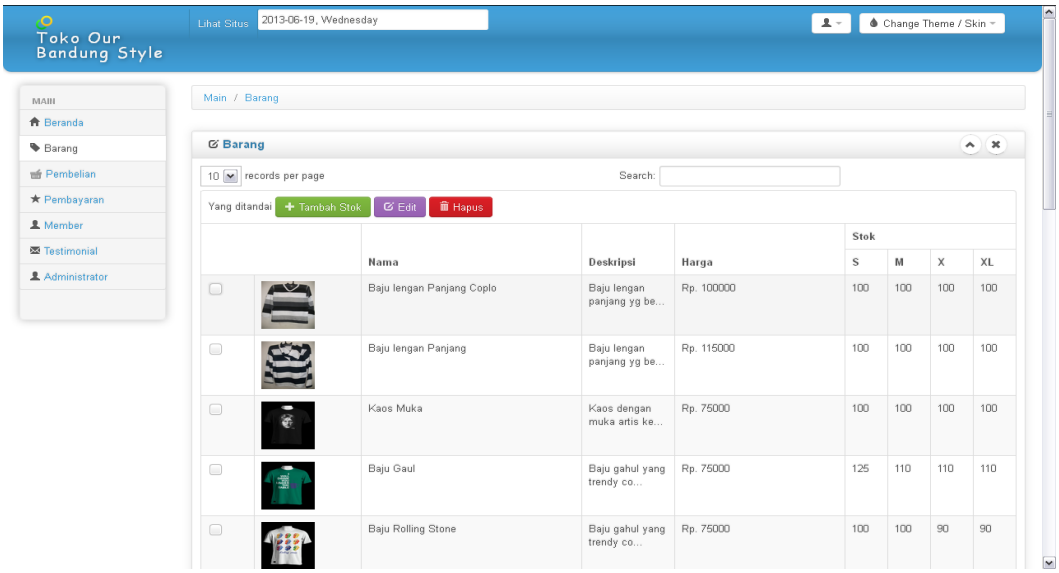
Menu ini berfungsi untuk melihat detail data member.



Gambar 5.10 Halaman Tampilan Utama Member

4. Halaman Tampilan Menu Kategori Produk / Barang

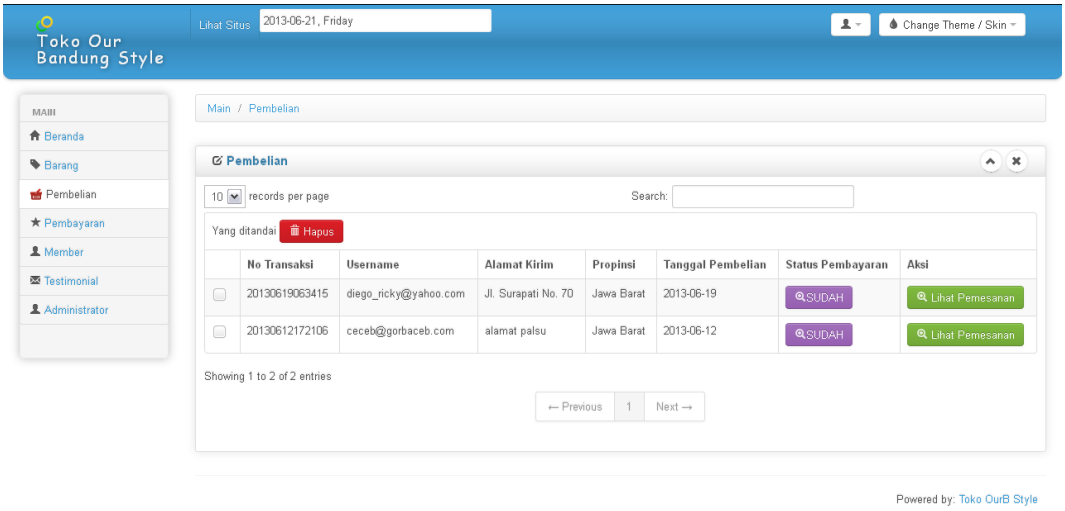
Kategori Menu ini berfungsi untuk menampilkan kategori produk. Di dalam menu ini *admin* melakukan tambah, *edit*, dan hapus kategori



Gambar 5.11 Halaman Tampilan Menu Kategori Produk

5. Halaman Tampilan Menu Pembelian Barang

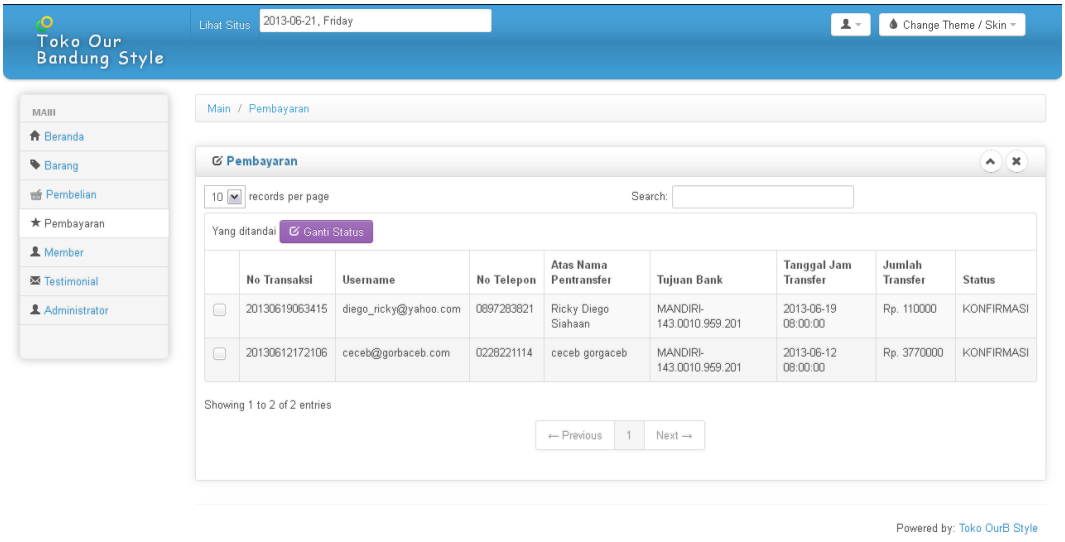
Transaksi Pembelian Menu ini berfungsi untuk menampilkan detail Pemesanan member. Di dalam menu ini *admin* dapat melihat Pemesanan Barang konsumen.



Gambar 5.12 Halaman Tampilan Menu Pembelian Barang

6. Halaman Tampilan Menu Pembayaran

Transaksi Pembayaran Menu ini berfungsi untuk menampilkan detail Pembayaran member. Di dalam menu ini *admin* dapat melihat status Pembayaran dan dapat melakukan perubahan terhadap status pembayaran



Gambar 5.13 Halaman Tampilan Menu Pembayaran Barang

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. Install program *web server* yang anda miliki, seperti appserv , Xampp, dll .
2. Copy folder program pada C:\XAMPP\htdocs untuk programnya.
3. Copy folder dbtravelC:\XAMPP\MySQL\data untuk databasenya
4. Ketikan pada URL *web browser* yang anda miliki yaitu : http://localhost /tokoonline
5. Panggil tampilan program awal sistem pembelian Toko mengetikan pada *web browser* yang anda miliki yaitu : http://localhost/tokoonline dan untuk *admin* menggunakan alamat http://localhost/tokoonline//adm/.

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* Toko OurB Style.

Tabel 5.1 Rencana Pengujian Untuk Tampilan *User*

Kasus Uji	Butir Uji
Register Member	Melakukan pengisian data pendaftaran member.
Login Member	Melakukan proses login member
Ganti password	Melakukan ganti password member
Pesan barang	Memasukan pesanan barang dan melakukan update jumlah di dalam keranjang
Input data pembayaran	Melakukan penginputan konfirmasi pembayaran

Tabel 5.2 Rencana Pengujian Untuk Tampilan *Admin*

Kasus Uji	Butir Uji	Jenis Pengujian
Login Admin	Melakukan proses sebagai admin	Blackbox
Barang	Melakukan *input, edit* serta hapus barang	Blackbox
Pembelian	Melakukan hapus jika member lewat jangka waktu yang di tentukan	Blackbox
Pembayaran	Melakukan ubah status pemesanan.	Blackbox
Member	Melakukan 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* Toko OurB Style 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)			
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)			
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* dengan data valid	Berhasil merubah *password* lama *user* menjadi *password* yang baru	Perubahan *password user* berhasil.	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Tidak mengisi lengkap *form* ganti *password*	Gagal mengganti *password* dan muncul pesan "masukan *password* lama"	Ganti *password user* gagal dan muncul pesan"isi masukan *password* lama"	[X] diterima [ ] ditolak
Mengisi *password* lama dengan data tidak *valid*	Gagal mengganti *password* dan muncul pesan "password lama anda tidak sesuai "	Password gagal diganti dan muncul pesan "password lama anda tidak sesuai"	[X] diterima [ ] ditolak
Mengisi *field password* baru berbeda dengan	Gagal mengganti *password* dan muncul	*Password* gagal diganti dan muncul	[X] diterima

field konfirmasi *password*	pesan "konfirmasi password baru salah"	pesan "konfirmasi password baru salah"	[ ] ditolak

4. Proses Pemesanan 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
1. Pengguna harus berada di halaman pemilihan produk yang akan dipesan 2. Memilih produk yang akan dipesan 3. Klik Tombol Pesan	Akan tersimpan dalam Database dan muncul Detail Pemesanan	Akan tersimpan dalam Database dan muncul Detail Pemesanan	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
1. Pengguna harus berada di halaman Pemilihan produk yang akan dipesan 2. Pengguna harus mengisi biodata lengkap setelah check outmemesan produk 3. Klik Tombol next	Akan terdisplay pesan salah pada sebelah textbox yang tidak sesuai ketentuan yang diberikan	Akan terdisplay pesan salah pada sebelah textbox yang tidak sesuai ketentuan yang diberikan	[X] diterima [ ] ditolak

5. Proses Input Pembayaran

Berikut ini adalah hasil pengujian *input* pembayaran

Tabel 5.7 Pengujian *Input* 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)			
Mengosongkan salah satu *field*	muncul pesan salah satu *field* yang kosong	Muncul pesan "salah satu *field*"	[X] diterima [ ] ditolak
Mengisi jumlah *transfer* yang salah	muncul pesan Jumlah transfer tidak sesuai dengan pemesanan	Muncul pesan "Jumlah transfer tidak sesuai dengan pemesanan"	[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* Toko OurB Style bebas dari kesalahan *syntax* dan secara fungsional telah mengeluarkan hasil yang *valid* karena proses validasi data dapat berfungsi dengan baik.