BAB V

IMPLEMENTASI DAN PENGUJIAN PROGRAM

Setelah dilakukan perancangan sistem, maka tahap selanjutnya adalah melakukan implementasi dan pengujian sistem. Tujuan dari implementasi sistem adalah untuk menerapkan sistem agar dapat dioperasikan secara optimal sesuai dengan kebutuhan proses. Pengujian merupakan kegiatan yang dilaksanakan untuk mencari atau menemukan kesalahan pada sistem yang telah dibangun.

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 w*ebsite* di Toko Blackberry Mania Bandung 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* 2.2.21, sebagai *web server*.
    b. *PHP* 5.3.10, sebagai bahasa pemograman.
    c. *MySQL* 5.5.20, untuk pengembangan dalam pembuatan *database*.
    d. *phpMyadmin* 3.4.10.1, untuk menangani administrasi *MySQL.*
    e. *Adobe Dreamweaver CS5*, 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*.
3. Implementasi Perangkat Lunak untuk *user*, antara lain :
    a. *Browser,* hasil terbaik menggunakan *Google Chrome*.

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 80 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 40 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

Pembuatan basis data dilakukan dengan menggunakan bahasa *SQL*, dimana *DBMS* yang digunakan adalah *MySQL*. Adapun implementasi basis datanya dalam *SQL* adalah sebagi berikut :

```
-- phpMyAdmin SQL Dump

-- version 3.4.10.1

-- http://www.phpmyadmin.net

--

-- Host: localhost

-- Generation Time: Jul 06, 2012 at 07:44 AM

-- Server version: 5.5.20

-- PHP Version: 5.3.10

--

-- Database: `medio_oktora`

--

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

-- Table structure for table `barang`

--

CREATE TABLE IF NOT EXISTS `barang` (

  `id_brg` int(2) NOT NULL AUTO_INCREMENT,

  `nama_brg` varchar(25) NOT NULL,

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

```
  `warna` varchar(10) NOT NULL,
  `harga` int(8) NOT NULL,
  `spek` varchar(100) NOT NULL,
  `stock` int(4) NOT NULL,
  `grs` varchar(10) NOT NULL,
  `ket` varchar(600) NOT NULL,
  `img` varchar(100) NOT NULL,
  `id_tipe` varchar(1) NOT NULL,
  PRIMARY KEY (`id_brg`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;

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

-- Table structure for table `detail_order`

--

CREATE TABLE IF NOT EXISTS `detail_order` (
  `id_order` varchar(15) NOT NULL,
  `id_brg` varchar(2) NOT NULL,
  `jml` int(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

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

-- Table structure for table `konfirmasi`

--

CREATE TABLE IF NOT EXISTS `konfirmasi` (

  `id_konfirm` varchar(15) NOT NULL,

  `id_order` varchar(15) NOT NULL,

  `cara_transfer` varchar(15) NOT NULL,

  `bank` varchar(10) NOT NULL,

  `pemilik` varchar(25) NOT NULL,

  `no_rek` varchar(15) NOT NULL,

  `ke_rek` varchar(15) NOT NULL,

  `jml_transfer` varchar(8) NOT NULL,

  `no_record` varchar(15) NOT NULL,

  `status_konfirm` enum('Belum Dikirim','Sudah Dikirim') NOT NULL,

  PRIMARY KEY (`id_konfirm`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

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

```
-- Table structure for table `kota`

--

CREATE TABLE IF NOT EXISTS `kota` (

  `id_kota` int(3) NOT NULL AUTO_INCREMENT,

  `nama_kota` varchar(25) NOT NULL,

  `id_prov` int(2) NOT NULL,

  PRIMARY KEY (`id_kota`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=498 ;

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

-- Table structure for table `kurir`

--

CREATE TABLE IF NOT EXISTS `kurir` (

  `id_kurir` int(1) NOT NULL,

  `nama_kurir` varchar(15) NOT NULL,

  PRIMARY KEY (`id_kurir`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

--

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

```
-- Table structure for table `ongkir`

--

CREATE TABLE IF NOT EXISTS `ongkir` (

  `id_ongkir` int(4) NOT NULL AUTO_INCREMENT,

  `id_kurir` int(1) NOT NULL,

  `id_kota` int(3) NOT NULL,

  `ongkir` int(6) NOT NULL,

  PRIMARY KEY (`id_ongkir`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=995 ;

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

-- Table structure for table `order`

--

CREATE TABLE IF NOT EXISTS `order` (

  `id_order` varchar(15) NOT NULL,

  `email_order` varchar(25) NOT NULL,

  `id_user` varchar(3) NOT NULL,

  `id_ongkir` varchar(4) NOT NULL,

  `id_kurir` int(1) NOT NULL,
```

```
  `total` int(10) NOT NULL,
  `status_order` enum('Belum Dibayar','Lunas') NOT NULL,
  PRIMARY KEY (`id_order`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- --------------------------------------------------------
-- Table structure for table `provinsi`
--
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_prov` int(2) NOT NULL AUTO_INCREMENT,
  `prov` varchar(25) NOT NULL,
  PRIMARY KEY (`id_prov`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=34 ;
-- --------------------------------------------------------
-- Table structure for table `retur`
--
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(3) NOT NULL AUTO_INCREMENT,
  `tgl_retur` varchar(15) NOT NULL,
```

```
  `no_resi` int(10) NOT NULL,
  `status_retur` enum('Unproses','Proses','Accepted','Rejected') NOT NULL,
  `alasan` varchar(300) NOT NULL,
  `id_order` varchar(15) NOT NULL,
  `nama_kurir` varchar(15) NOT NULL,
  PRIMARY KEY (`id_retur`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
-- --------------------------------------------------------
-- Table structure for table `tipe_barang`
--
CREATE TABLE IF NOT EXISTS `tipe_barang` (
  `id_tipe` int(1) NOT NULL AUTO_INCREMENT,
  `nama_tipe` varchar(15) NOT NULL,
  PRIMARY KEY (`id_tipe`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
--
-- --------------------------------------------------------
--
```

```
-- Table structure for table `tujuan`

--

CREATE TABLE IF NOT EXISTS `tujuan` (

  `id_tujuan` int(3) NOT NULL AUTO_INCREMENT,

  `id_order` varchar(15) NOT NULL,

  `nama_tujuan` varchar(50) NOT NULL,

  `alamat_tujuan` varchar(150) NOT NULL,

  `kodepos_tujuan` int(5) NOT NULL,

  `tlp_tujuan` varchar(12) NOT NULL,

  `id_kota` int(3) NOT NULL,

  `id_prov` int(2) NOT NULL,

  PRIMARY KEY (`id_tujuan`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

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

-- Table structure for table `user`

--

CREATE TABLE IF NOT EXISTS `user` (

  `id_user` int(3) NOT NULL AUTO_INCREMENT,
```

```
  `nama_user` varchar(50) NOT NULL,
  `alamat_user` varchar(150) NOT NULL,
  `tlp` varchar(12) NOT NULL,
  `id_prov` int(2) NOT NULL,
  `id_kota` int(3) NOT NULL,
  `kode_pos` int(5) NOT NULL,
  `email_user` varchar(25) NOT NULL,
  `pass` varchar(30) NOT NULL,
  `status_user` enum('aktif','nonaktif','diblok') CHARACTER SET latin1
COLLATE latin1_spanish_ci NOT NULL,
  PRIMARY KEY (`id_user`),
  UNIQUE KEY `email_user` (`email_user`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=102 ;
--
-- ------------------------------------------------------
-- ------------------------------------------------------
--
```

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* pada Toko Blackberry Mania 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, Katalog, Profil, Cara Belanja, Kontak Kami). Dan pada *footer* terdapat *link* yang sama (Home, Katalog, Profil, Cara Belanja, Kontak Kami, Syarat dan Ketentuan, FAQ, Peta Situs, BNI, Mandiri, BCA, JNE, Tiki)



Gambar 5.1 Menu Utama *Website* Toko Blackberry Mania

1.2. *Form Login*

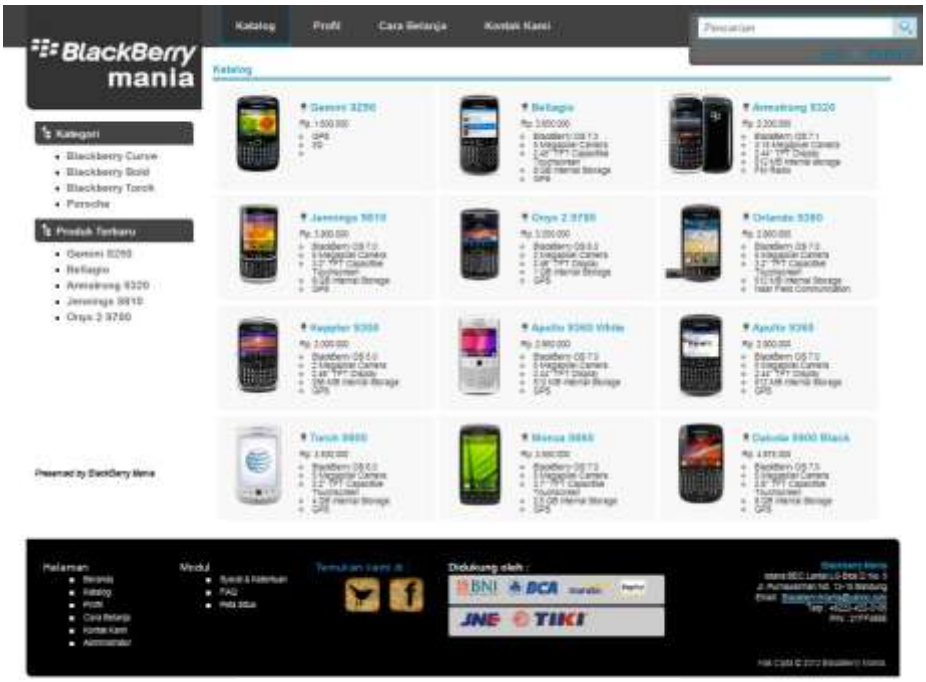
Suatu *form* dimana jika seorang pengunjung ingin melakukan pembelian, maka pengunjung diwajibkan mengisi *login user*, dan jika belum terdaftar maka pengunjung diberikan fasilitas pendaftaran yang ditujukan untuk menjadi member.



Gambar 5.2 Menu *Log in* Member

1.3. Halaman Katalog

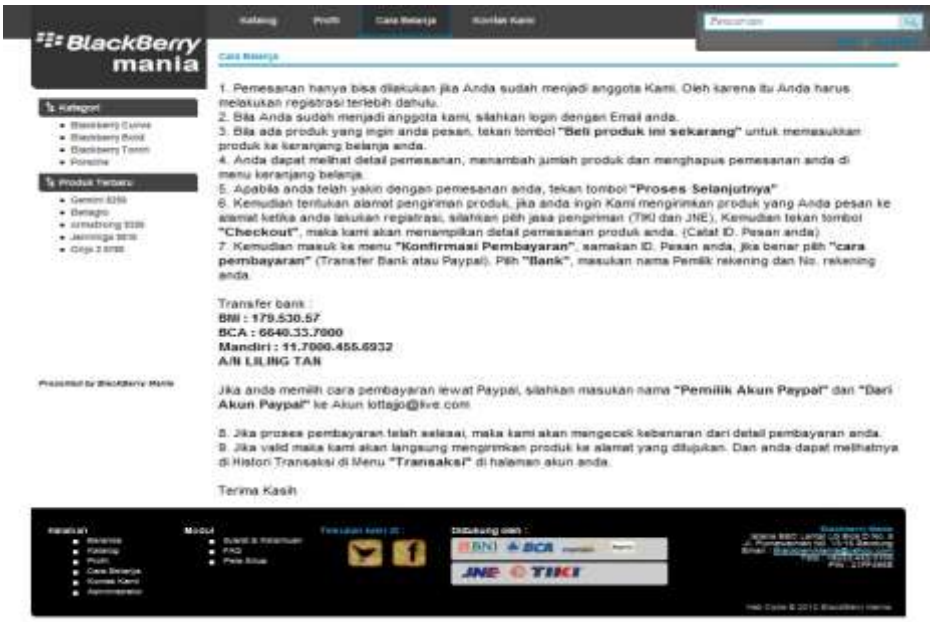
Pada halaman katalog ini berisi semua tipe produk blackberry yang tersedia pada *website* Toko Blackberry Mania Bandung. Lengkap dengan gambar, spesifikasi dan harga produk. Pada halaman ini juga bisa dilihat kategori/tipe serta produk terbaru dari Blackberry.



Gambar 5.3 Tampilan Katalog

1.4. Tampilan Cara Belanja

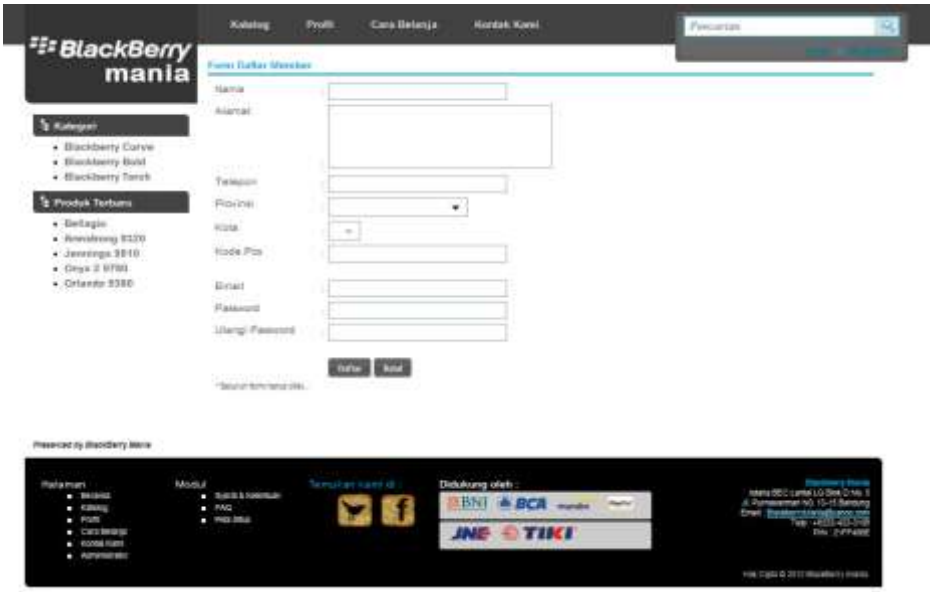
Pada halaman ini berisi detail *step by step* cara belanja pada *website* Toko Blackberry Mania Bandung mulai dari cara pendaftaran, pemesanan, dan pembayaran.



Gambar 5.4 Tampilan Cara Belanja

1.5. Tampilan *Form* Registrasi

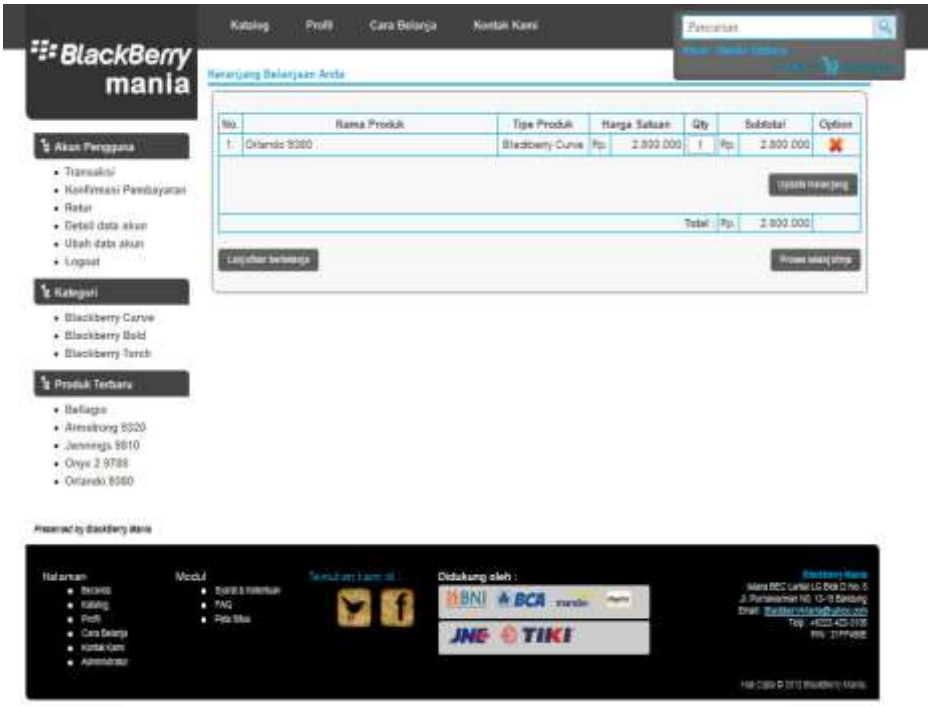
Untuk melakukan transaksi, sebelumnya kita sudah terdaftar menjadi member. *Form* pendaftaran bisa dilihat pada gambar di bawah ini.



Gambar 5.5 Menu Registrasi Member

1.6. Keranjang Belanja Member

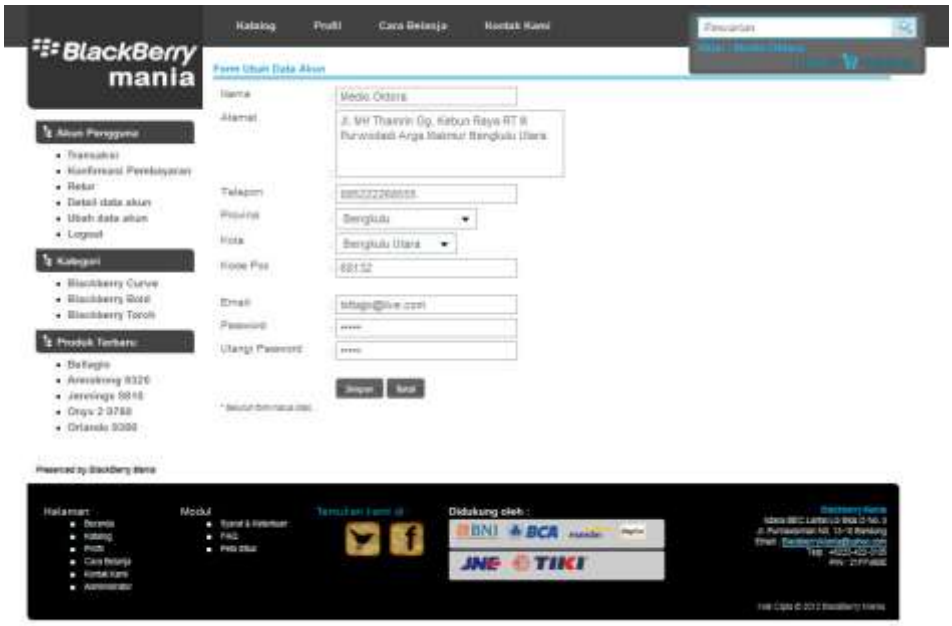
Pada tampilan utama member, konsumen dapat memilih produk yang ingin dipesan, apabila ada produk yang ingin dipesan, maka konsumen dapat menekan tombol "Beli Produk Ini Sekarang!" dan order member akan ditampilkan pada *shopping cart*. Apabila ingin memilih produk lagi, maka terlebih dahulu menekan tombol "Update Keranjang", kemudian menekan tombol "Lanjutkan Belanja". Untuk menghapus produk yang telah dipesan menekan tombol yang berada di bawah tulisan "Option". Apabila telah selesai, maka konsumen dapat menekan tombol "Proses Selanjutnya".



Gambar 5.6 Tampilan Keranjang Belanja Member

1.7. Member Area

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.

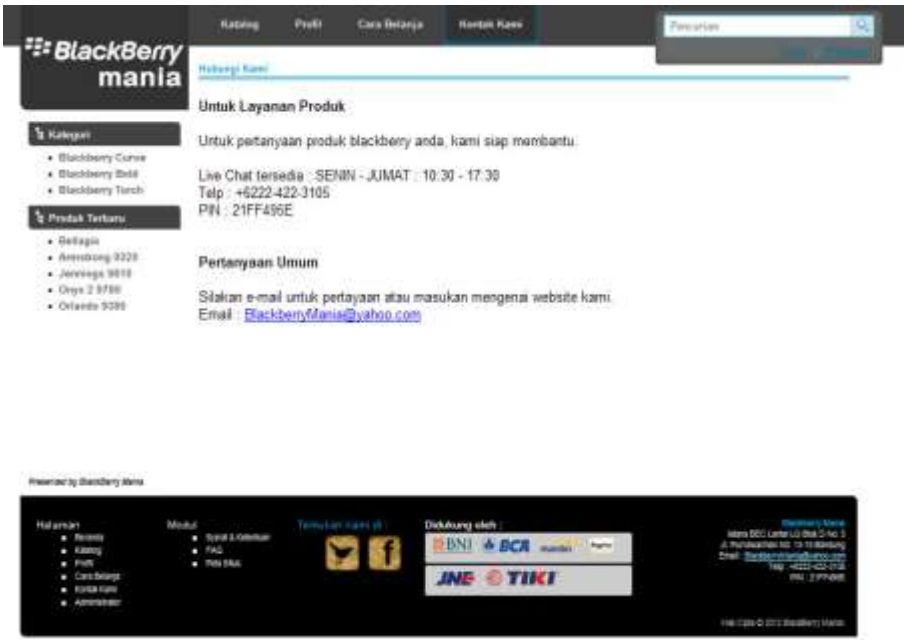


Gambar 5.7 Tampilan *Form* Ubah Data Akun

1.8. Menu Kontak Kami

Menu kontak ini berfungsi untuk memudahkan *user* dalam menyampaikan pertanyaan seputar produk blackberry, seperti harga dan spesifikasi.

Di halaman ini juga tersedia *email* Toko Blackberry Mania Bandung yang digunakan sebagai sarana untuk pertanyaan atau masukan seputar *website* Toko Blackberry Mania Bandung.



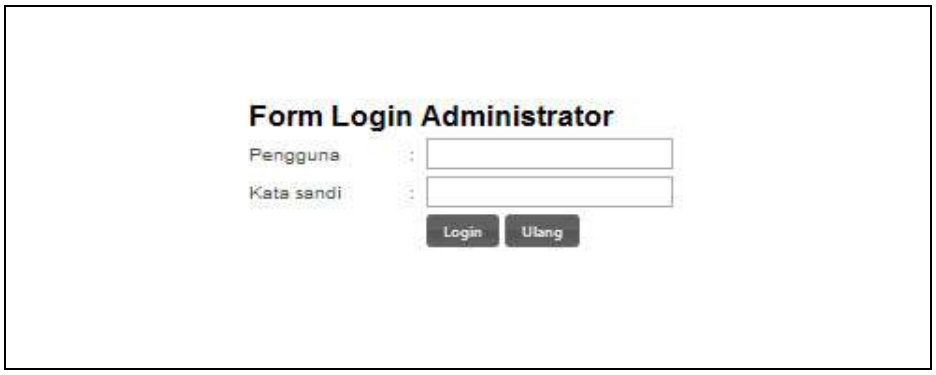
Gambar 5.8 Tampilan Kontak Kami

2. Tampilan *Admin*

Tampilan *admin* merupakan tampilan yang digunakan oleh *admin* untuk melakukan kontrol terhadap data-data yang akan ditampilkan pada *Website* Toko Blackberry Mania. 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. *Login* Administrator

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



Gambar 5.9 Tampilan Menu *Login* Admin

2.2. Tampilan Utama Administrator

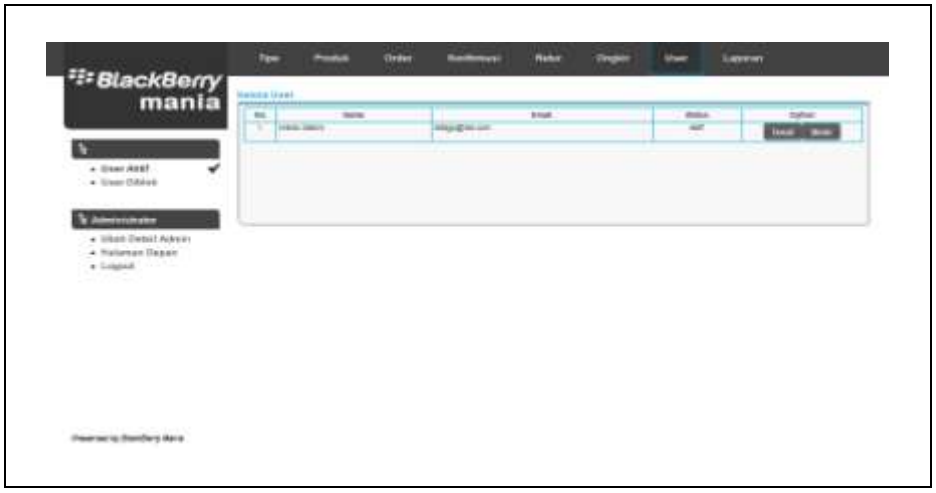
Pada halaman utama administrator ini terdapat sembilan menu, yaitu : Home, Tipe, Produk, Order, Konfirmasi, Retur, Ongkir, Laporan, dan *Logout*.



Gambar 5.10 Menu Utama Admin

2.3. Menu Member

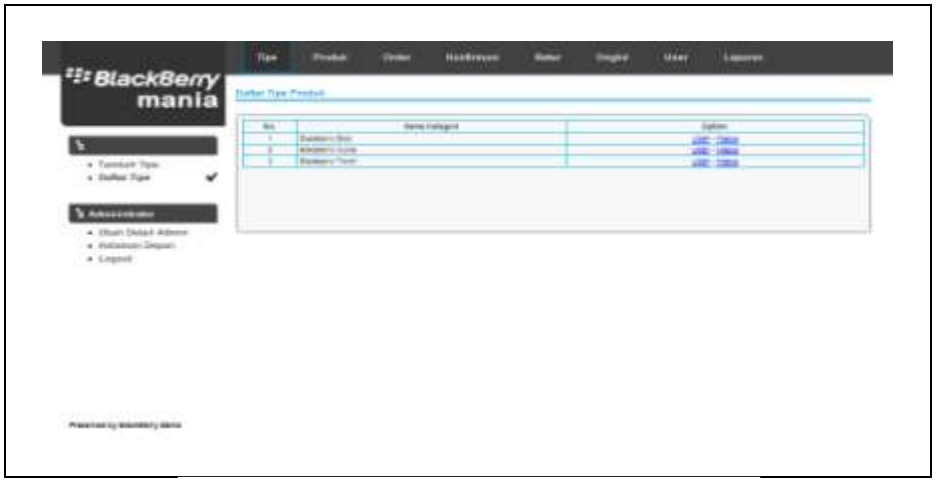
Menu ini berfungsi untuk melihat detail konsumen yang sudah menjadi member.



Gambar 5.11 Tampilan Menu Member

2.4. Menu Tipe Produk

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



Gambar 5.12 Tampilan Menu Tipe Produk

2.5. Menu Produk

Menu ini berfungsi untuk menampilkan data produk. Di dalam menu ini terdapat beberapa fasilitas untuk tambah, hapus, dan *edit* produk.



Gambar 5.13 Tampilan Menu Produk

2.6. Menu Order

Menu ini berfungsi untuk menampilkan detail order member. Di dalam menu ini admin dapat melihat status order dan dapat melakukan perubahan terhadap status order.



Gambar 5.14 Tampilan Menu Order

3. Menu Laporan

   Menu ini berfungsi untuk melihat laporan barang dan laporan penjualan.



Gambar 5.15 Tampilan Menu Laporan

5.1.6. Implementasi Instalasi Program

Implementasi instalasi program merupakan aspek awal untuk menjelaskan bagaimana perancangan aplikasi *website* 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 Wamp Server* untuk servernya.
2. Lakukan konfigurasi *software Wamp Server* (*wamp-control.exe*), dengan mengaktifkan *Apache* dan *MySQL*.
3. Konfigurasi *database* yang ada pada *phpMyAdmin*, dengan menuju *link*, *http://localhost/phpmyadmin/*.

4. Konfigurasi penyimpanan *file* data *web* Toko Blackberry Mania secara *offline*, dengan menyimpan *file web* Toko Blackberry Mania pada folder *htdocs*.
5. Konfigurasi Koneksi Database, pastikan *file database* yang ada pada folder *website* Toko Blackberry Mania dengan nama data koneksinya adalah aksesdb.php, harus memiliki sintak *database* yang sama.
6. Setelah proses instalasi selesai semua maka untuk mengakses menu utama *website* Toko Blackberry Mania Bandung menggunakan alamat : *http://localhost/medio_oktora* dan untuk *admin* menggunakan alamat *http://localhost/ medio_oktora /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* Toko Blackberry Mania.

Tabel 5.1 Rencana Pengujian Untuk Tampilan *User*

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

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*
Order	Melakukan proses perubahan detail dan hapus order member	*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 Blackberry Mania 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	[ ] 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 "Harap isi semua *form* pendaftaran dengan lengkap"	[ ] 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	[ ] 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	[ ] 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"	[ ] 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.	[ ] diterima [ ] ditolak
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"	[ ] 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"	[ ] 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"	[ ] 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.	[ ] diterima [ ] ditolak
Mengupdate jumlah pesanan pada field jumlah di keranjang belanja	Jumlah data dikeranjang belanja berhasil di *update*.	Berhasil mengupdate jumlah pemesanan barang	[ ] 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	[ ] diterima [ ] ditolak
Memasukkan *field* jumlah bukan angka	*Update* jumlah tidak dapat dilakukan dan *field* jumlah kosong	*Field* jumlah di keranjang belanja kosong	[ ] 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	[ ] 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 "Harap isi semua form pemesan dengan lengkap"	[ ] 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	[ ] 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* Toko Blackberry Mania bebas dari kesalahan *syntax* dan secara fungsional telah mengeluarkan hasil yang *valid* karena proses validasi data dapat berfungsi dengan baik.