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 *Website E-Commerce* di Toko Soneta 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. WampServer (2.2), sebagai *web server*.
    b. PHP (5.5.20), sebagai bahasa pemograman.
    c. MySQL(5.0.67), untuk pengembangan dalam pembuatan *database*.
    d. 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. *Website* dapat di akses melalui alamat

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 sebagai berikut :

```
-- phpMyAdmin SQL Dump
-- version 3.4.10.1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Jul 02, 2012 at 07:58 PM
-- Server version: 5.5.20
-- PHP Version: 5.3.10

SET SQL_MODE="NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";


/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
```

```
-- Database: `1`

--


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


--

-- Table structure for table `customer`

--


CREATE TABLE IF NOT EXISTS `customer` (

  `id_customer` int(5) NOT NULL AUTO_INCREMENT,

  `email` varchar(100) NOT NULL,

  `nama` varchar(50) NOT NULL,

  `alamat` varchar(150) NOT NULL,

  `id_ongkir` int(5) NOT NULL,

  `kode_pos` int(5) NOT NULL,

  `telp` varchar(20) NOT NULL,

  `sandi` varchar(50) NOT NULL,

  `status_customer` enum('active','inactive','blocked') NOT NULL,

  PRIMARY KEY (`id_customer`),

  UNIQUE KEY `email` (`email`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;


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

```
--

-- Table structure for table `kategori_pd`

--


CREATE TABLE IF NOT EXISTS `kategori_pd` (

  `id_kategori` int(11) NOT NULL AUTO_INCREMENT,

  `nama_kategori` varchar(30) NOT NULL,

  PRIMARY KEY (`id_kategori`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;


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


--

-- Table structure for table `konfirmasi`

--


CREATE TABLE IF NOT EXISTS `konfirmasi` (

  `id_konfirm` varchar(20) NOT NULL,

  `id_order` varchar(20) NOT NULL,

  `cara_bayar` varchar(25) NOT NULL,

  `bank` enum('BNI','BCA','Mandiri') NOT NULL,

  `pemilik` varchar(50) NOT NULL,

  `no_rek` decimal(30,0) NOT NULL,
```

```
  `jml_bayar` int(15) NOT NULL,
  `ke_rek` varchar(30) NOT NULL,
  `status_konfirm` enum('Belum Dikirim','Sudah Dikirim') NOT NULL,
  `no_resi` varchar(30) NOT NULL,
  PRIMARY KEY (`id_konfirm`),
  UNIQUE KEY `id_transaksi` (`id_order`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

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

CREATE TABLE IF NOT EXISTS `ongkir` (
  `id_ongkir` int(5) NOT NULL AUTO_INCREMENT,
  `provinsi` varchar(30) NOT NULL,
  `kota` varchar(30) NOT NULL,
  `ongkir` int(20) NOT NULL,
  PRIMARY KEY (`id_ongkir`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=82 ;

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

```
--

-- Table structure for table `order`

--


CREATE TABLE IF NOT EXISTS `order` (

  `id_order` varchar(50) NOT NULL,

  `id_customer` int(11) NOT NULL,

  `email_order` varchar(100) DEFAULT NULL,

  `id_ongkir` varchar(11) NOT NULL,

  `total` int(11) NOT NULL,

  `status_order` enum('Belum Dibayar','Lunas') NOT NULL,

  PRIMARY KEY (`id_order`)

) ENGINE=InnoDB DEFAULT CHARSET=latin1;


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


--

-- Table structure for table `order_det`

--


CREATE TABLE IF NOT EXISTS `order_det` (

  `id_order` varchar(50) NOT NULL,

  `id_pd` int(5) NOT NULL,

  `jumlah` int(5) NOT NULL
```

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

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

--
-- Table structure for table `penerima`
--

CREATE TABLE IF NOT EXISTS `penerima` (
  `id_penerima` varchar(40) NOT NULL,
  `id_order` varchar(20) NOT NULL,
  `nama2` varchar(50) NOT NULL,
  `alamat2` varchar(100) NOT NULL,
  `id_ongkir` int(5) NOT NULL,
  `kode_pos2` int(6) NOT NULL,
  `telp2` varchar(15) NOT NULL,
  PRIMARY KEY (`id_penerima`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

--
-- Table structure for table `produk`
--
```

```
CREATE TABLE IF NOT EXISTS `produk` (
  `id_pd` int(5) NOT NULL AUTO_INCREMENT,
  `nama_pd` varchar(50) NOT NULL,
  `id_kategori` int(11) NOT NULL,
  `harga_pd` int(15) NOT NULL,
  `stock_pd` int(11) NOT NULL,
  `img_pd` varchar(150) NOT NULL,
  `ket_pd` varchar(200) NOT NULL,
  PRIMARY KEY (`id_pd`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=47 ;

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

--
-- Table structure for table `retur`
--

CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(11) NOT NULL AUTO_INCREMENT,
  `id_order` varchar(30) NOT NULL,
  `tgl_retur` varchar(30) NOT NULL,
  `jasa_pengiriman` varchar(50) NOT NULL,
  `no_resi` varchar(50) NOT NULL,
```

```
  `alasan` varchar(200) NOT NULL,
  `status_retur` enum('Diajukan','Diterima','Ditolak') NOT NULL,
  PRIMARY KEY (`id_retur`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

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 E-Commerce* pada Toko Soneta dibuat menarik dan sederhana sehingga *admin* dan juga *user* dapat mengoperasikan aplikasi ini dengan mudah. Untuk lebih jelasnya akan digambarkan sebagai berikut :

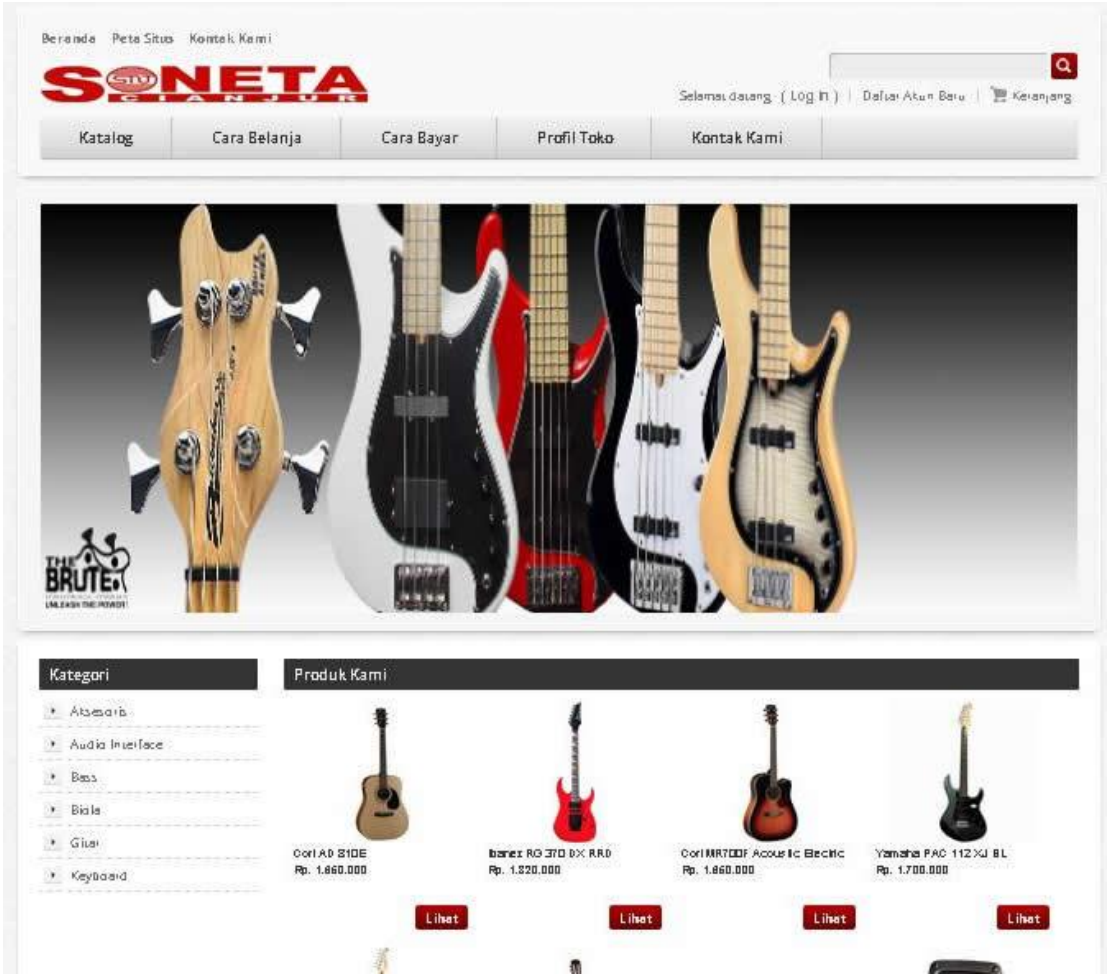
1. Tampilan *User*

   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* (Beranda, Katalog, Cara Belanja, Cara Bayar, Profil Toko, Kontak Kami).



Gambar 5.1 Menu Utama *Website* Toko Soneta

1.2. Halaman *Log in*

Suatu *form* dimana jika seorang pengunjung ingin melakukan pembelian, maka pengunjung diwajibkan mengisi *login user*, dan jika

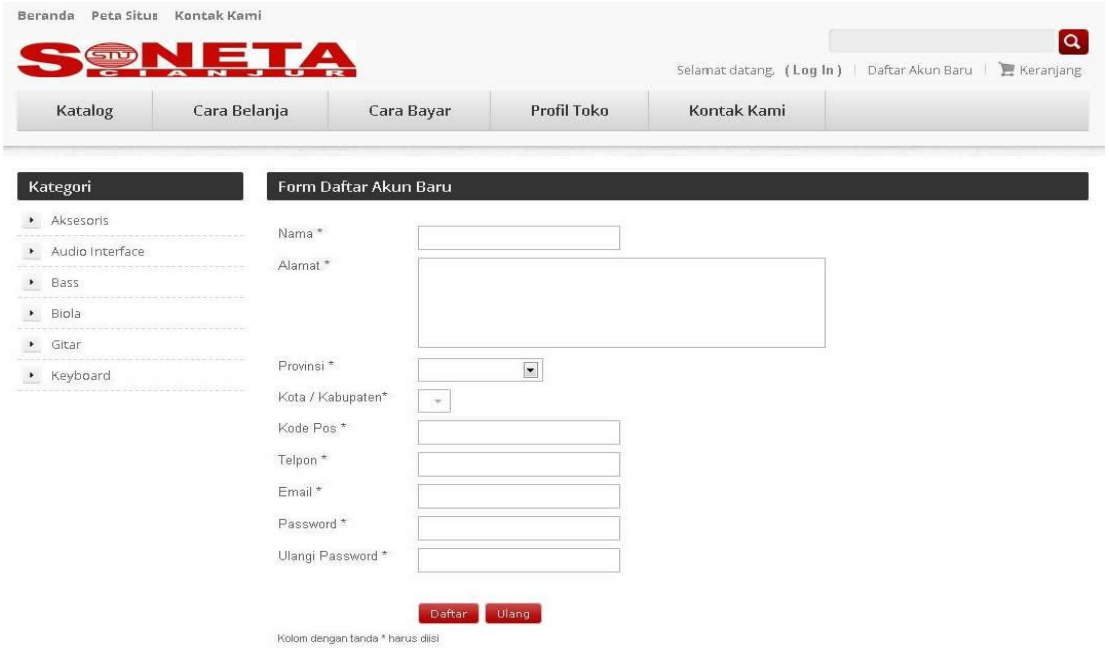
belum terdaftar maka pengunjung diberikan fasilitas pendaftaran yang ditujukan untuk menjadi member.



Gambar 5.2 Menu *Log in* Member

1.3. Tampilan *Form* Registrasi

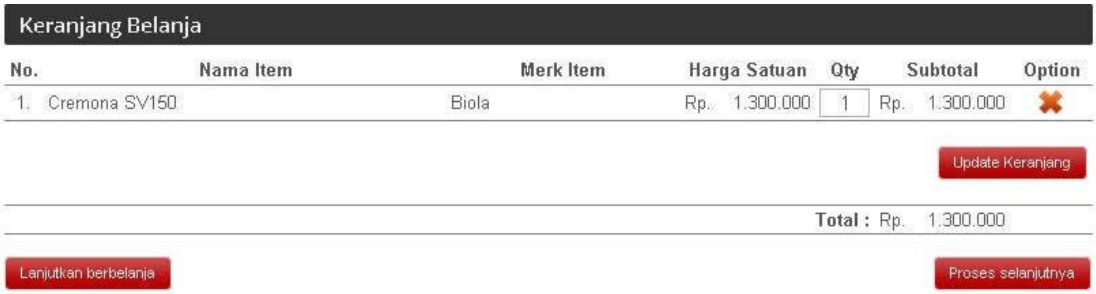
Untuk melakukan transaksi, sebelumnya kita sudah terdaftar menjadi member. *Form* pendaftaran bisa dilihat pada gambar di bawah ini.



Gambar 5.3 Menu Registrasi Member

1.4. Keranjang Belanja *Member*

Pada tampilan utama *member*, *customer* dapat memilih produk yang ingin dipesan, apabila ada produk yang ingin dipesan, maka konsumen dapat menekan tombol "Beli Sekarang" dan pesanan *member* akan ditampilkan pada Keranjang Belanja. Apabila ingin menambah banyak 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" untuk alamat pengiriman.

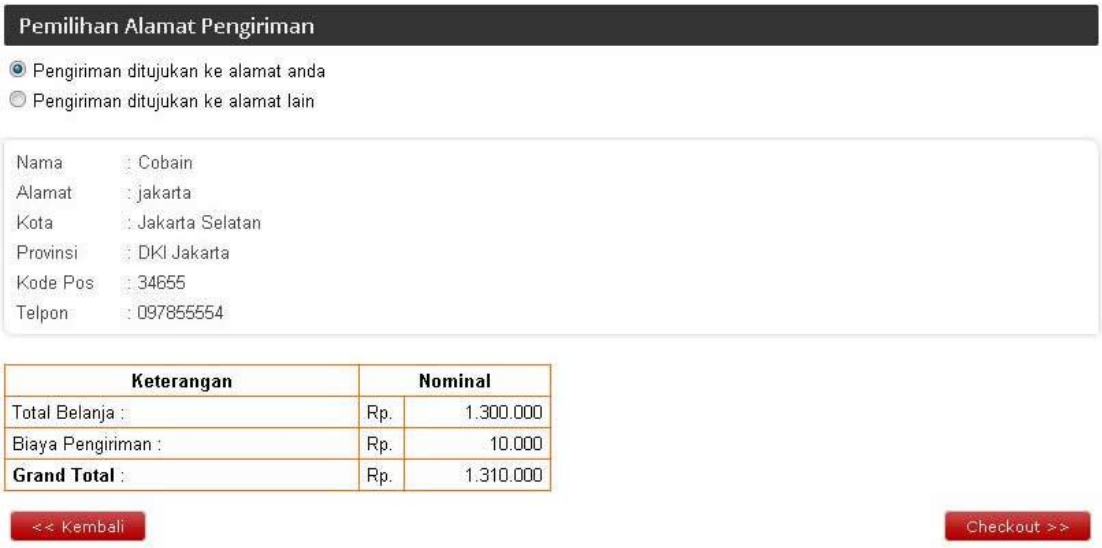


Gambar 5.4 Tampilan Keranjang Belanja Member

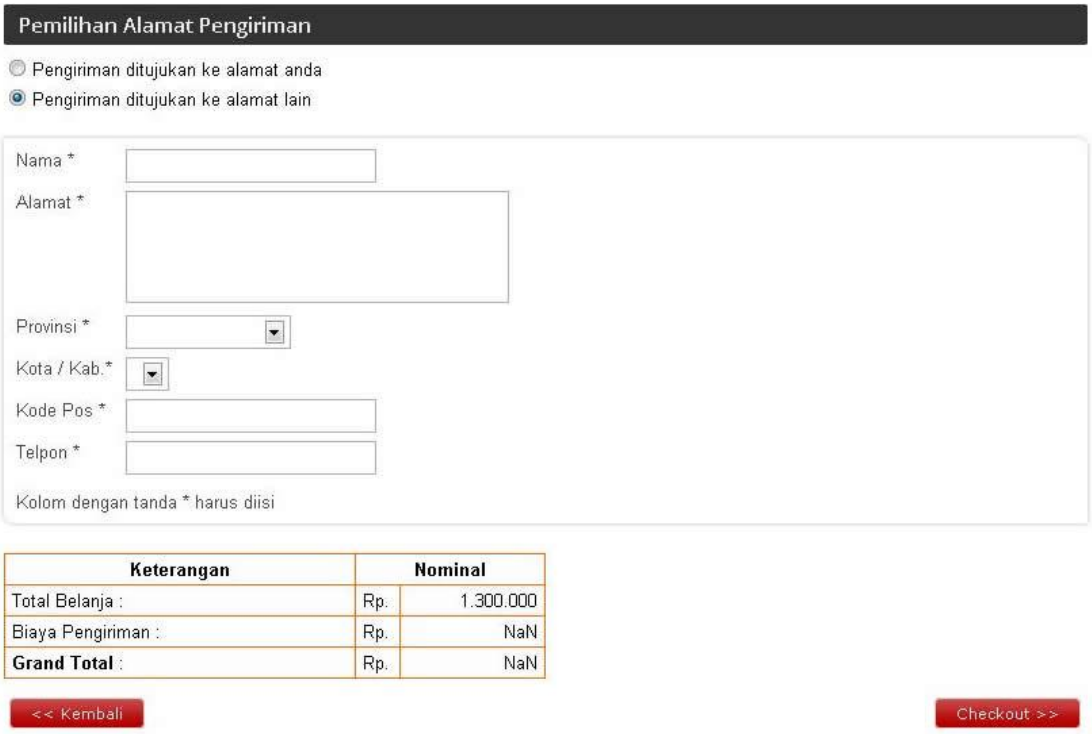
1.5. Pemilihan Alamat Pemesanan

Menu ini merupakan halaman lanjutan dari 'Proses Selanjutnya' khusus *user* yang berisi mengenai pemilihan alamat pengiriman. Di dalam menu ini, *user* dapat menentukan alamat pengiriman. Jika barang dikirim untuk *user*, maka akan dikirim sesuai alamat *user* pada saat registrasi. Tetapi jika barang dikirimkan untuk alamat lain, *user* dapat

mengganti alamat pengiriman dengan mengisi form yang telah disediakan. Bila telah sesuai, selanjutnya menekan tombol ‘Checkout’ untuk proses penerimaan ID Order dan konfirmasi pembayaran.



Gambar 5.5 Tampilan Pemilihan Alamat untuk *user*



Gambar 5.6 Tampilan Menu Pemilihan Alamat Lain

1.6. Menu Bukti Pemesanan

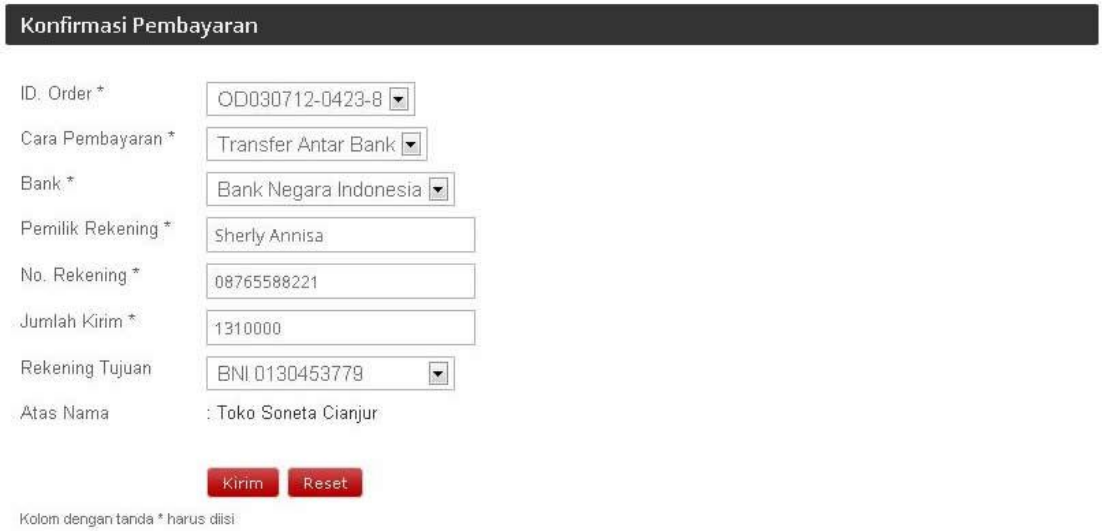
Menu ini berisi mengenai Bukti Pemesanan pada *Website* Toko Soneta berupa ID Order, jumlah banyak barang yang dipesan, harga pengiriman, dan total jumlah yang harus dibayar. Jika sesuai pesanan, maka user dapat melanjutkan pada proses Konfirmasi Pembayaran dengan menekan tombol ‘di sini’ yang tercetak berwarna biru.



Gambar 5.7 Tampilan Menu Bukti Pemesanan

1.7. Menu Konfirmasi Pembayaran

Menu ini berfungsi untuk *user* dalam melakukan Konfirmasi Pembayaran produk yang telah dipesan. Dalam menu ini, user diharuskan mengisi form yang telah disediakan dengan benar. Bila ada kesalahan *input, user* bisa menekan tombol 'Reset', jika telah sesuai tekan tombol 'Kirim'.



Gambar 5.8 Tampilan Menu Konfirmasi Pembayaran

2. Tampilan *Admin*

Tampilan *admin* merupakan tampilan yang digunakan oleh *admin* untuk melakukan kontrol terhadap data-data yang akan ditampilkan pada *Website* Toko Soneta. Selain itu *admin* juga dapat melakukan kontrol terhadap data produk, data pemesanan, serta mencetak laporan stok 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 Pengguna dan Kata Sandi *admin*.



Gambar 5.9 *Log in Admin*

2.2. Tampilan Utama Administrator

Pada halaman utama administrator ini terdapat sembilan menu, yaitu: Kategori, Produk, Pemesanan, Konfirmasi, Retur, Biaya Kirim, Pesan, Laporan, dan Logout.



Gambar 5.10 Menu Utama *Admin*

1. Menu Kategori

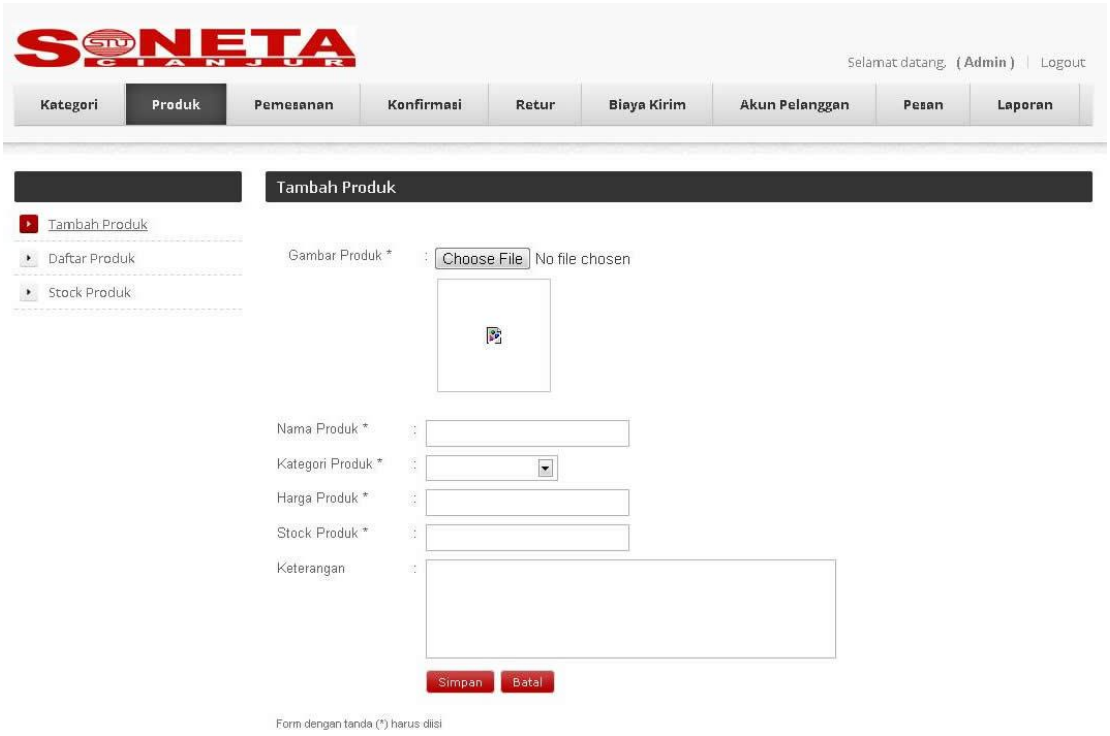
   Menu ini berfungsi untuk menampilkan kategori produk. Di dalam menu ini *admin* melakukan tambah, ubah, dan hapus kategori.



Gambar 5.11 Tampilan Menu Admin Kategori

2. Menu Produk

   Menu ini berfungsi untuk menampilkan data produk. Di dalam menu ini *admin* melakukan tambah, ubah, dan hapus produk.



Gambar 5.12 Tampilan Menu Produk

3. Menu Pemesanan

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



Gambar 5.13 Tampilan Menu Pemesanan

4. Menu Konfirmasi

   Menu ini berfungsi untuk menampilkan data yang sudah dan belum di konfirmasi. Di dalam menu ini *admin* dapat melihat status konfirmasi dan dapat melakukan perubahan terhadap status konfirmasi.



Gambar 5.14 Tampilan Menu Konfirmasi

5. Menu Retur

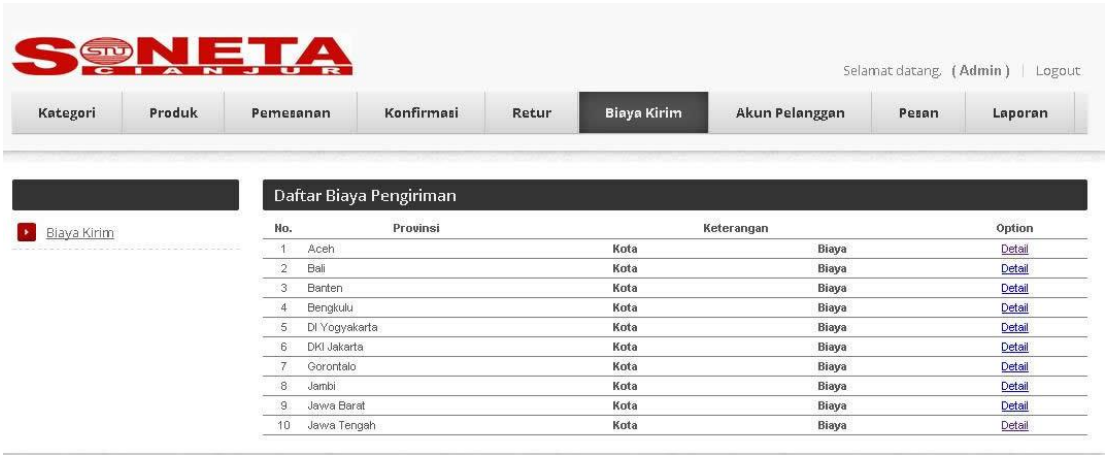
   Menu ini berfungsi untuk melihat laporan retur barang yang dilakukan oleh *user*.



Gambar 5.15 Tampilan Menu Retur

6. Menu Biaya Kirim

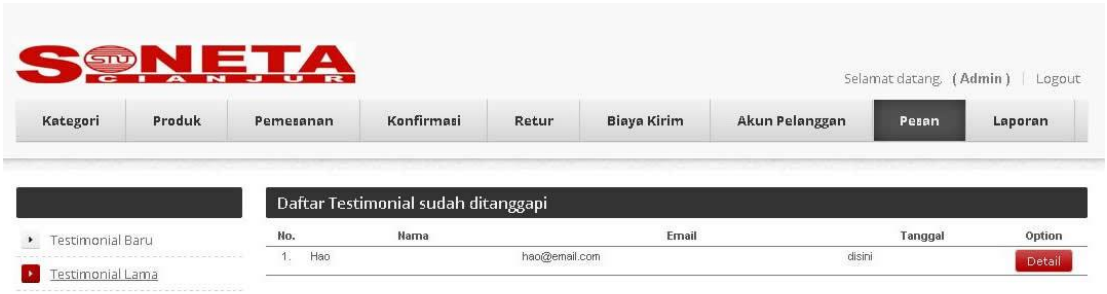
   Menu ini berfungsi menampilkan biaya kirim ke daerah-daerah tujuan. Pada menu ini, Admin hanya dapat mengubah jumlah biaya kirim. Bila ingin menambahkan/mengubah nama daerah, dapat dilakukan pada *database.*



Gambar 5.16 Tampilan Menu Biaya Kirim

7. Menu Pesan

   Menu ini berfungsi untuk menampilkan pesan/testimoni dari *user*.



Gambar 5.17 Tampilan Menu Pesan

8. Menu Laporan

   Menu ini berfungsi untuk menampilkan jumlah transaksi penjualan yang terjadi pada Toko Soneta. Tidak semua transaksi ditampilkan pada laporan ini. Hanya transaksi yang telah berstatus ‘terkirim’ yang akan ditampilkan. Karena, status ‘terkirim’ menunjukan bahwa pesanan tersebut telah dibayar dan dikirim.



Gambar 5.18 Tampilan Menu Laporan

5.1.6. Implementasi Instalasi Program

Implementasi instalasi program merupakan aspek awal untuk menjelaskan bagaimana perancangan aplikasi *Website E-Commerce* 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*, dengan menekan *Start WampServer* untuk mengaktifkan Apache dan MySQL.
3. Konfigurasi *database* yang ada pada PhpMyAdmin, dengan menuju *link*, http://localhost/phpmyadmin/.
4. Konfigurasi penyimpanan file data *web* Toko Soneta secara *offline*, dengan menyimpan file *website* Toko Soneta pada folder *www*.
5. Konfigurasi Koneksi Database, pastikan file *database* yang ada pada folder *Website* Toko Soneta 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 Soneta menggunakan alamat : http://localhost/project_web dan untuk *admin* menggunakan alamat http://localhost/project_web/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 E-Commerce* Toko Soneta.

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 Sandi	Melakukan ganti Sandi 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 Soneta 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 "Harap isi semua *form* pendaftaran dengan lengkap"	[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 Pengguna dan Sandi *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 Pengguna dan Sandi dengan data yang tidak *valid*	Gagal *login* dan muncul pesan "Login Gagal"	*Login* gagal dan muncul pesan "Login Gagal"	[X] diterima [ ] ditolak

3. Proses Ganti *Password*

Berikut ini adalah hasil pengujian ganti *password.*

Tabel 5.5 Pengujian Lupa Kata Sandi

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi *field email* pada saat registrasi	Berhasil merubah Sandi lama *user* menjadi Sandi yang baru	Perubahan Sandi *user* berhasil.	[X] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi lengkap *form* ganti Kata Sandi	Gagal mengganti Kata Sandi dan muncul pesan "isi data yang lengkap jika ingin merubah password Anda"	Ganti Kata Sandi *user* gagal dan muncul pesan"isi data yang lengkap jika ingin merubah password anda"	[X] diterima [ ] ditolak
Mengisi Kata Sandi lama dengan data tidak *valid*	Gagal mengganti Kata Sandi dan muncul pesan "Kata Sandi lama anda salah"	Kata Sandi gagal diganti dan muncul pesan "Kata Sandi lama anda salah"	[X] diterima [ ] ditolak
Mengisi *field* Kata Sandi baru berbeda dengan *field* konfirmasi Kata Sandi	Gagal mengganti Kata Sandi dan muncul pesan "konfirmasi Kata Sandi baru salah"	Kata Sandi gagal diganti dan muncul pesan "konfirmasi Kata Sandi 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 "Beli Sekarang"	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 "Beli Sekarang"	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 "Harap isi semua form pemesan dengan lengkap"	[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 E-Commerce* Toko Soneta bebas dari kesalahan *syntax* dan secara fungsional telah mengeluarkan hasil yang *valid* karena proses validasi data dapat berfungsi dengan baik.