BAB V

IMPLEMENTASI DAN PENGUJIAN PROGRAM

Setelah dilakukan perancangan sistem, tahap selanjutnya adalah 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.

5.1.1. Batasan Implementasi (*Optional*)

Dalam mengimplementasikan perangkat lunak *Website E-Commerce* pada Riin and Shop Collection 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* (1.6.8), sebagai *web server*.
    b. *PHP* (5.2.6), sebagai bahasa pemograman.
    c. *MySQL*(5.0.67), untuk pengembangan dalam pembuatan *database*.
    d. *Adobe Dreamweaver Cs,* sebagai editor dalam pengembangan *design 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 3.5.1
-- http://www.phpmyadmin.net
--
-- Host: localhost
-- Generation Time: Jan 20, 2014 at 04:40 AM
-- Server version: 5.5.24-log
-- PHP Version: 5.4.3
```

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

--
-- Database: `db_sidik`
--

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

--
-- Table structure for table `ci_sessions`
--

CREATE TABLE IF NOT EXISTS `ci_sessions` (
  `session_id` varchar(40) NOT NULL DEFAULT '0',
  `ip_address` varchar(16) NOT NULL DEFAULT '0',
  `user_agent` varchar(50) NOT NULL,
  `last_activity` int(10) unsigned NOT NULL DEFAULT '0',
  `user_data` text NOT NULL,
  PRIMARY KEY (`session_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

--
-- Table structure for table `informasi`
--

CREATE TABLE IF NOT EXISTS `informasi` (
  `id_informasi` int(11) NOT NULL AUTO_INCREMENT,
  `nama_informasi` varchar(50) DEFAULT NULL,
  `isi_informasi` text,
  `tanggal` datetime DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id_informasi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

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

--
-- Table structure for table `katagori`
--

CREATE TABLE IF NOT EXISTS `katagori` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `kode_katagori` varchar(10) NOT NULL,
  `nama_katagori` varchar(60) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=20 ;

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

--
-- Table structure for table `konfirmasi_pembayaran`
```

```
--

CREATE TABLE IF NOT EXISTS `konfirmasi_pembayaran` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `tgl_transfer` date NOT NULL,
  `nama_bank` varchar(30) NOT NULL,
  `nama_nasabah` varchar(50) NOT NULL,
  `nominal` int(11) DEFAULT NULL,
  `tujuan_transfer` varchar(30) NOT NULL,
  `id_pemesanan` varchar(30) NOT NULL,
  `id_member` int(11) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_pemesanan` (`id_pemesanan`),
  KEY `id_member` (`id_member`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

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

--
-- Table structure for table `kontak`
--

CREATE TABLE IF NOT EXISTS `kontak` (
  `id_kontak` int(11) NOT NULL AUTO_INCREMENT,
  `nama_pengirim` varchar(50) DEFAULT NULL,
  `email` varchar(30) DEFAULT NULL,
  `telp` varchar(20) DEFAULT NULL,
  `pesan` text,
  `tanggal` datetime DEFAULT NULL,
  PRIMARY KEY (`id_kontak`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;

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

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

CREATE TABLE IF NOT EXISTS `kota` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama_kota` varchar(50) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  `id_provinsi` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_provinsi` (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;

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

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

```
CREATE TABLE IF NOT EXISTS `member` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `email` varchar(50) NOT NULL,
  `password` varchar(150) NOT NULL,
  `nama_depan` varchar(100) NOT NULL,
  `nama_belakang` varchar(100) DEFAULT NULL,
  `id_provinsi` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `alamat` varchar(150) NOT NULL,
  `kode_pos` varchar(7) NOT NULL,
  `telp` varchar(15) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `email` (`email`),
  KEY `id_provinsi` (`id_provinsi`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;

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

--
-- Table structure for table `mitra_pengiriman`
--

CREATE TABLE IF NOT EXISTS `mitra_pengiriman` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `kode_mitra` varchar(10) NOT NULL,
  `nama_mitra` varchar(50) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

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

--
-- Table structure for table `pemesanan`
--

CREATE TABLE IF NOT EXISTS `pemesanan` (
  `id` varchar(30) NOT NULL,
  `id_member` int(11) NOT NULL,
  `nama_depan` varchar(50) DEFAULT NULL,
  `nama_belakang` varchar(50) DEFAULT NULL,
  `id_provinsi` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `alamat` varchar(150) NOT NULL,
  `kode_pos` varchar(5) NOT NULL,
  `telp` varchar(15) NOT NULL,
  `nama_depan2` varchar(50) DEFAULT NULL,
  `nama_belakang2` varchar(50) DEFAULT NULL,
  `id_provinsi2` int(11) NOT NULL,
  `id_kota2` int(11) NOT NULL,
  `alamat2` varchar(150) NOT NULL,
  `kode_pos2` varchar(5) NOT NULL,
  `telp2` varchar(15) NOT NULL,
```

```
  `id_mitra_kirim` int(11) NOT NULL,
  `sub_total` int(11) DEFAULT NULL,
  `biaya_kirim` int(11) DEFAULT NULL,
  `grand_total` int(11) DEFAULT NULL,
  `jenis_pembayaran` varchar(30) NOT NULL,
  `komentar` varchar(200) DEFAULT NULL,
  `date` datetime DEFAULT NULL,
  `nomer_pengiriman` varchar(30) DEFAULT NULL,
  `status_pengiriman` varchar(3) DEFAULT NULL,
  `tgl_pengiriman` date DEFAULT NULL,
  `status` varchar(3) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_mitra_kirim` (`id_mitra_kirim`),
  KEY `id_provinsi` (`id_provinsi`),
  KEY `id_provinsi2` (`id_provinsi2`),
  KEY `id_kota` (`id_kota`),
  KEY `id_kota2` (`id_kota2`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

--
-- Table structure for table `pemesanan_det`
--

CREATE TABLE IF NOT EXISTS `pemesanan_det` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_produk_det` int(11) NOT NULL,
  `harga` int(11) DEFAULT NULL,
  `bnyk` int(11) NOT NULL,
  `total` int(11) DEFAULT NULL,
  `disk` int(11) DEFAULT NULL,
  `sub_total` int(11) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  `id_pemesanan` varchar(30) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_pemesanan` (`id_pemesanan`),
  KEY `id_produk_det` (`id_produk_det`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=17 ;

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

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

CREATE TABLE IF NOT EXISTS `produk` (
  `id` varchar(10) NOT NULL,
  `nama_produk` varchar(50) NOT NULL,
  `warna` varchar(30) NOT NULL,
  `harga` int(11) DEFAULT NULL,
  `disk` int(11) DEFAULT NULL,
  `stok` int(11) DEFAULT NULL,
  `id_katagori` int(11) DEFAULT NULL,
```

```
  `keterangan` varchar(150) DEFAULT NULL,
  `tanggal` datetime DEFAULT NULL,
  `image1` varchar(50) DEFAULT NULL,
  `image2` varchar(50) DEFAULT NULL,
  `image3` varchar(50) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_katagori` (`id_katagori`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

--
-- Table structure for table `produk_det`
--

CREATE TABLE IF NOT EXISTS `produk_det` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `size_produk` varchar(20) NOT NULL,
  `stok` int(11) DEFAULT NULL,
  `berat` double(11,1) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  `id_produk` varchar(10) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_produk` (`id_produk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=32 ;

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

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

CREATE TABLE IF NOT EXISTS `provinsi` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama_provinsi` varchar(50) NOT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;

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

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

CREATE TABLE IF NOT EXISTS `retur` (
  `id` varchar(30) NOT NULL,
  `tgl_retur` datetime NOT NULL,
  `id_transaksi` varchar(30) NOT NULL,
  `status` varchar(3) NOT NULL,
  `status_pengiriman` varchar(3) DEFAULT NULL,
  `id_mitra_kirim` int(11) DEFAULT NULL,
  `tgl_pengiriman_retur` date DEFAULT NULL,
```

```
  `nomer_pengiriman` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`id`),
  KEY `id_transaksi` (`id_transaksi`),
  KEY `id_mitra_kirim` (`id_mitra_kirim`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

--
-- Table structure for table `retur_det`
--

CREATE TABLE IF NOT EXISTS `retur_det` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `id_pemesanan_det` int(11) NOT NULL,
  `bny` int(11) DEFAULT NULL,
  `alasan` varchar(250) DEFAULT NULL,
  `id_retur` varchar(30) NOT NULL,
  `status` varchar(3) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_pemesanan_det` (`id_pemesanan_det`),
  KEY `id_retur` (`id_retur`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;

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

--
-- Table structure for table `setting`
--

CREATE TABLE IF NOT EXISTS `setting` (
  `id` int(11) NOT NULL DEFAULT '0',
  `value` varchar(200) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

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

--
-- Table structure for table `tarif_kirim`
--

CREATE TABLE IF NOT EXISTS `tarif_kirim` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `harga_kirim` int(11) NOT NULL,
  `id_mitra` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  KEY `id_mitra` (`id_mitra`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=21 ;

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

```
--
-- Table structure for table `user`
--

CREATE TABLE IF NOT EXISTS `user` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama_user` varchar(50) NOT NULL,
  `password` varchar(150) NOT NULL,
  `akses` tinyint(1) DEFAULT NULL,
  `status` tinyint(1) DEFAULT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

5.1.5. Implementasi Antar Muka dan Penggunaan Program

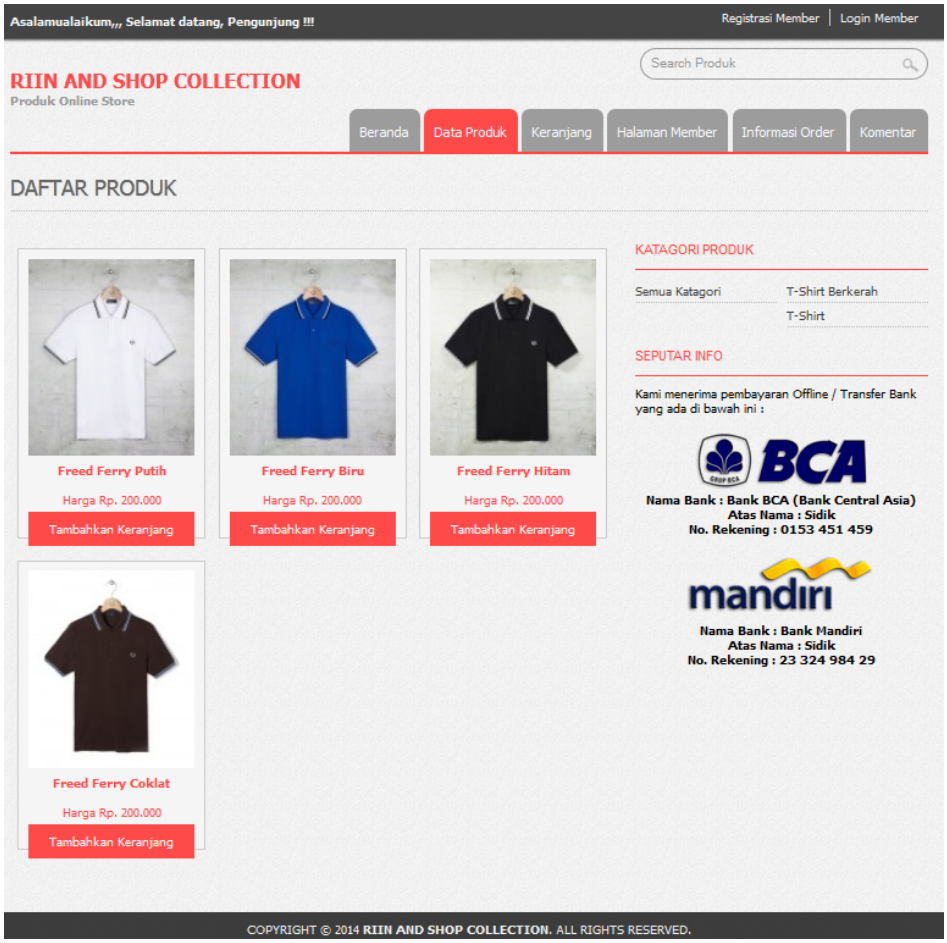
Dalam pembuatan aplikasi apapun, implementasi antar muka juga mempunyai peranan yang penting. Implementasi antar muka dapat memudahkan *user* dalam mengoperasikan aplikasi yang telah dibuat. Oleh karena itu, antar muka pada aplikasi *Website E-Commerce* pada Riin and Shop Collection dibuat menarik dan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

1. ***User Client***

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

1.1 Halaman Beranda

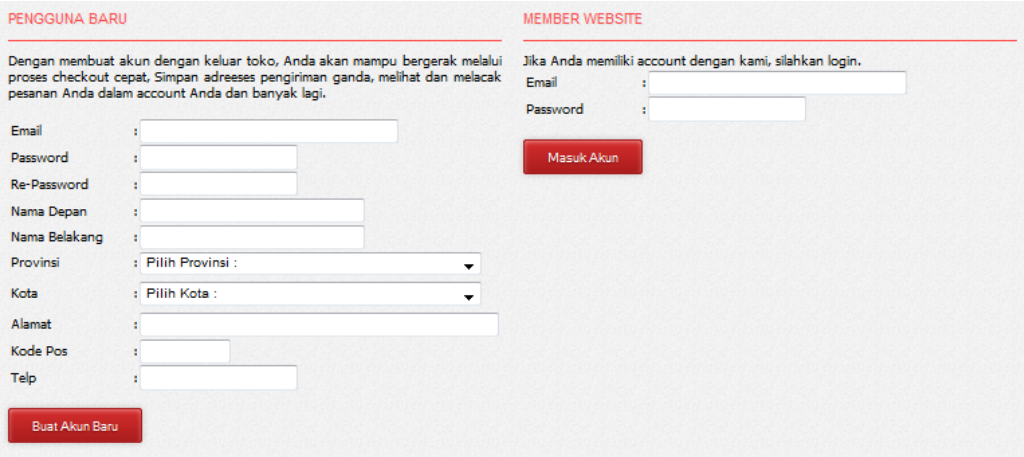
Pada halaman beranda terdapat beberapa *menu* yang dapat membantu para pengguna untuk menggunakan *website* ini, diantaranya pada *header* terdapat *menu* (Beranda, Keranjang Belanja, Member, Informasi dan Kontak) dan di content terdapat daftar produk dan kategori produk



Gambar 5.1 Halaman Beranda *Website* Pada Riin and Shop Collection

1.2 Menu *Login* Member dan Pendaftaran Member Baru

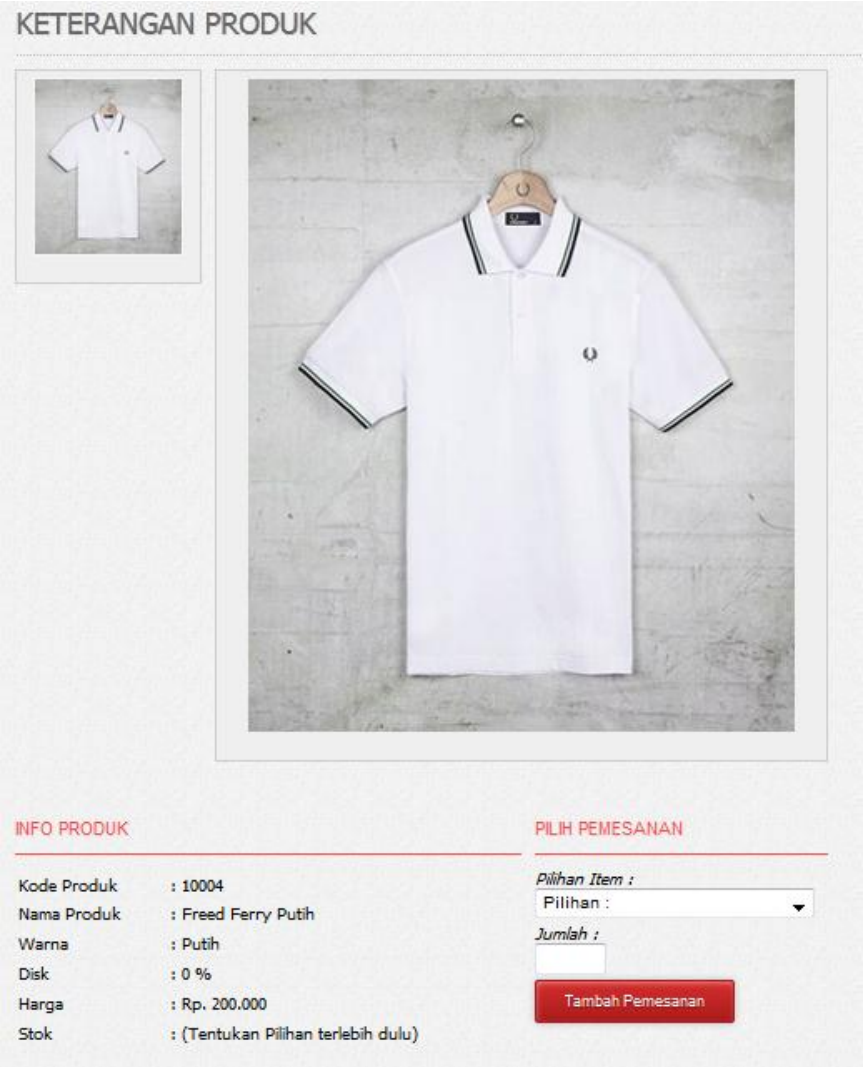
Jika seorang pengguna ingin melakukan transaksi, maka diwajibkan *login* terlebih dahulu dan jika belum daftar maka di wajibkan mengisi form pendaftaran member untuk menjadi member baru.



Gambar 5.2 *Form Login* Member dan Pendaftaran Member Baru Riin and Shop Collection

1.3 Menu Detail Produk

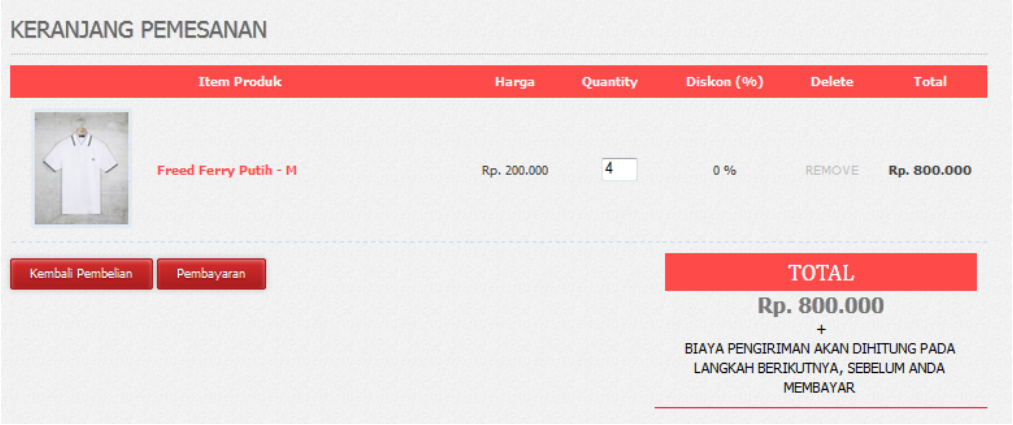
Pada menu ini ketika si pengguna ingin melakukan pemesanan maka si pengguna harus memilih dulu pilih ukuran produk dan banyak lalu klik *button* tambah keranjang belanja agar masuk ke keranjang belanja.



Gambar 5.3 Menu Detail Produk Pada Riin and Shop Collections

1.4 Menu Keranjang belanja

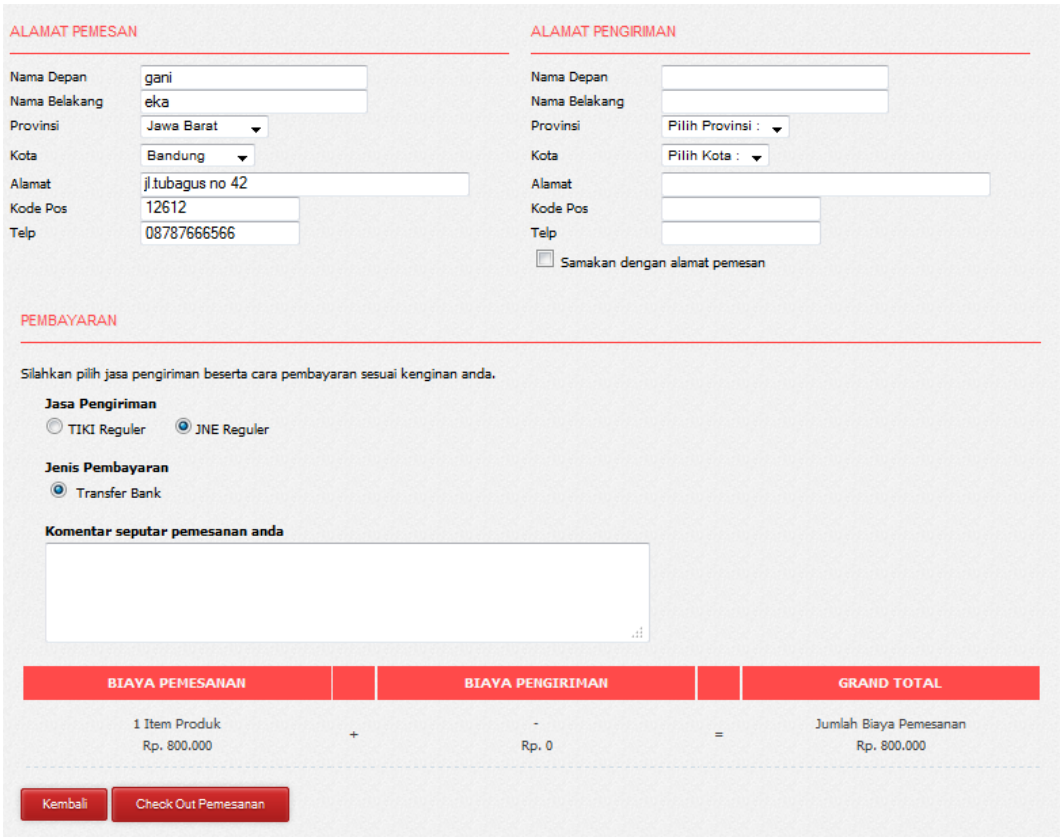
Pada menu ini si pengguna dapat melihat daftar produk yang di pesan dan apabila terjadi kesalahan maka si pengguna dapat melakukan *remove* di keranjang belanja, jika akan melakukan pemesanan lagi *klik button* kembali ke pembelian dan apabila akan meneruskan pemesanan *klik button* pembayaran.



Gambar 5.4 Menu Keranjang Belanja Pada Riin and Shop Collection

1.5 Menu *CheckOut* Pemesanan

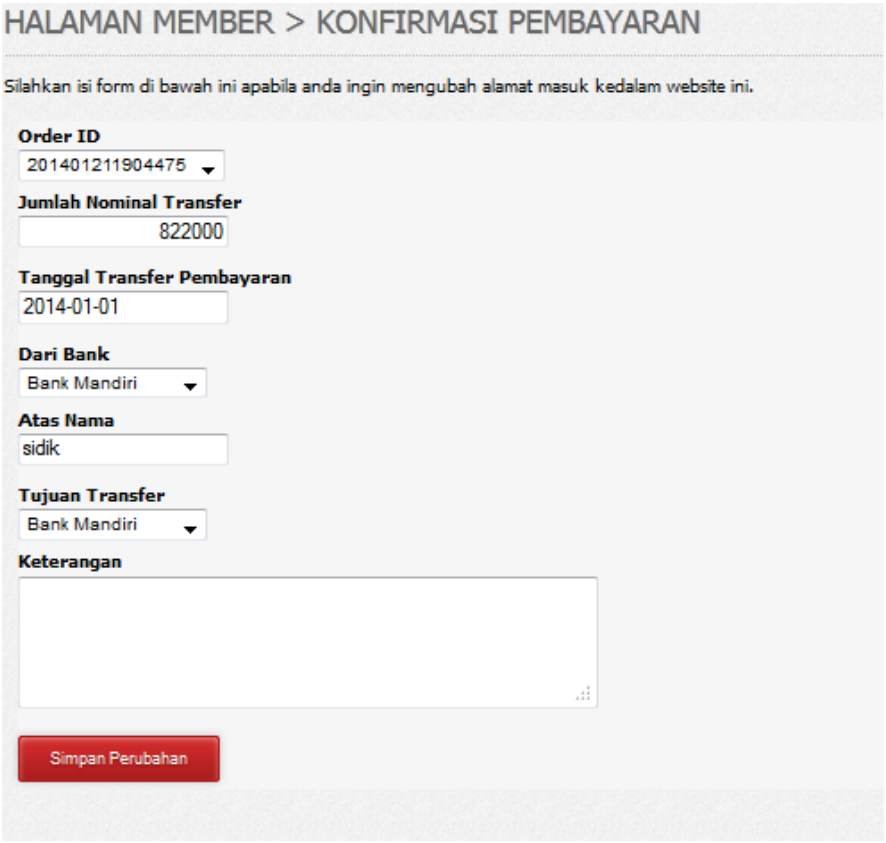
Pada menu ini si pengguna mengisi *form* data pengiriman, jika alamat pemesan sama dengan alamat tujuan pengiriman makan bisa di cek samakan dengan alamat pemesan secara otomatis akan sama dengan data alamat pemesan kemudian memilih jasa pengiriman dan memilih jenis pembayaran yang akan di pakai si pengguna lalu klik *button CheckOut* Pemesanan.



Gamabar 5.5 Menu *CheckOut* Pemesanan Pada Riin and Shop Collection

1.6 Menu Konfirmasi Pembayaran

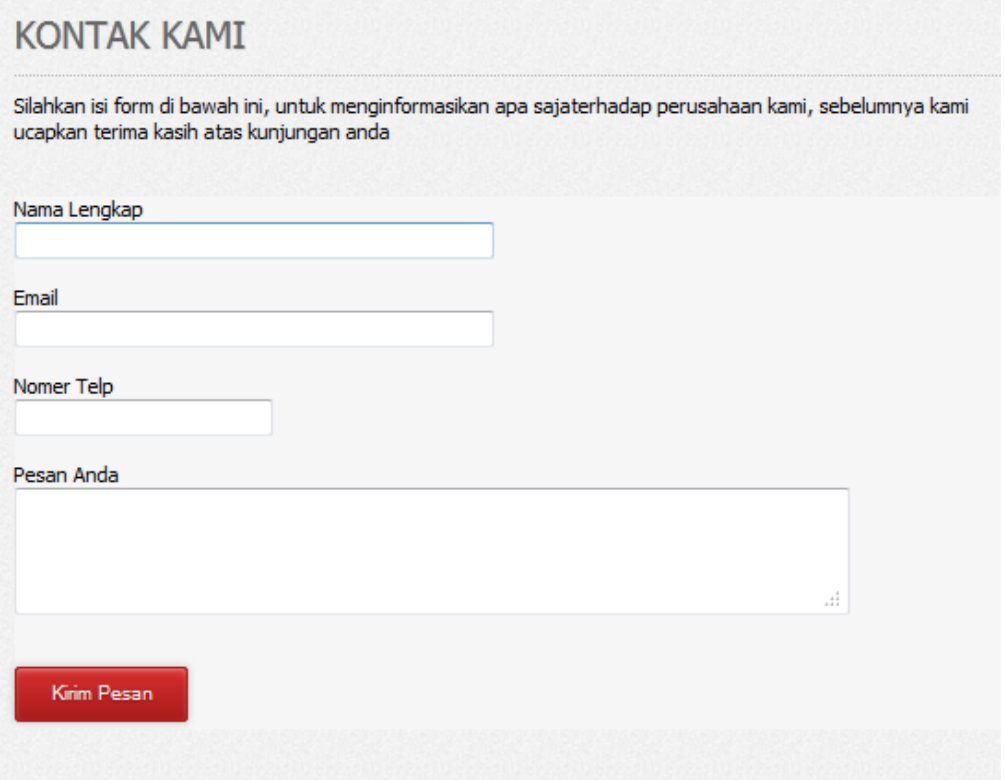
Pada menu ini sesudah si pengguna *checkout* pemesanan dan si pemesan sudah mentransfer jumlah dari total, maka di haruskan melakukan konfirmasi dengan memilih order id, jumlah nominal uang yang di transfer, tanggal, dari bank, atas nama, tujuan transfer apabila ada keterangan silahkan si pennguna menulis dan klik *button* simpan.



Gambar 5.6 Menu Konfirmasi Pembayaran Pada Riin and Shop Collection

1.7 Menu Kontak

Menu ini berfungsi untuk berkomunikasi dengan *Admin* apabila si pengguna ingin menanyakan sesuatu, si pengguna bisa menulis pesan di menu kontak ini.



Gambar 5.7 Menu Kontak Pada Riin and Shop Collection

1.8 Menu Cara Belanja

Pada Menu ini menjelaskan mengenai tata cara bagaiman cara melakukan transaksi / pemesanan produk pada *website* Riin and Shop Collection.



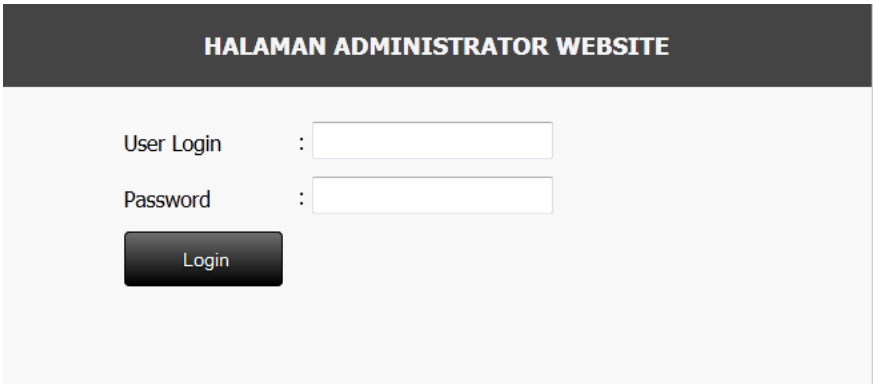
Gambar 5.8 Menu Cara Belanja Riin and shop Collection

2. Administrator

Tampilan administrator merupakan tampilan yang digunakan oleh *admin* untuk melakukan kontrol terhadap data-data yang akan ditampilkan pada *website* Riin and Shop Collection. 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 administrator* dengan memasukkan *username admin* dan *password admin*.

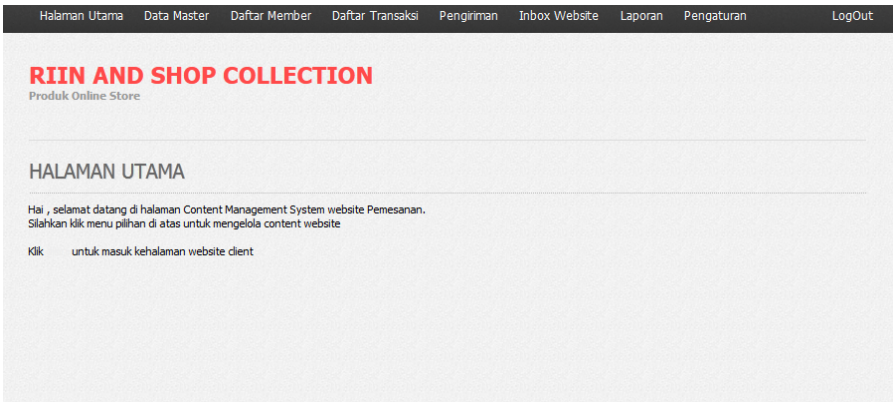


Gambar 5.9 Menu *Login Administrator* Pada Riin and Shop Collection

2.2. Halaman Utama *Administrator*

Pada halaman utama ini terdapat menu-menu yang dapat membantu *admin* untuk menjalankan program ini, diantaranya pada *header* terdapat *link* (Beranda, Data Master dengan sub menu Kategori, Produk, Biaya Pengiriman, Jasa Pengiriman, Kota, Provinsi, Data Member, Status dengan sub menu Transaksi

Pemesanan, Pengiriman Pemesanan, Konfirmasi pembayaran, Laporan dengan sub menu Laporan Pemesanan, Laporan Stok Barang, Pengaturan dan *Logout*)



Gambar 5.10 Menu Halaman Utama *Administrator* Pada Riin and Shop Collection

2.3 Menu Data Master

Di dalam menu ini terdapat 6 sub menu-menu yaitu :

1. Sub Menu Kategori

   a. Data Kategori

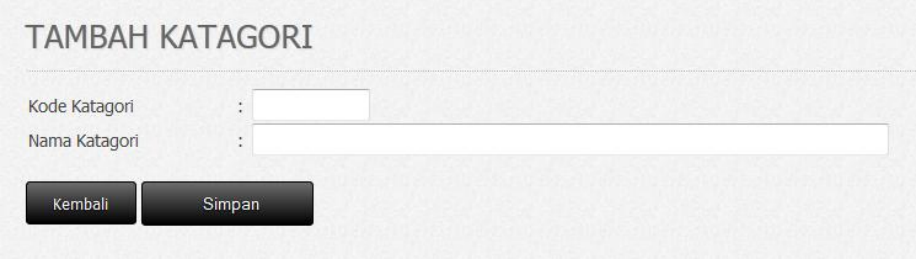
      Di menu ini admin dapat melakukan tambah, edit dan hapus data kategori.



Gambar 5.11 Data Kategori Pada Riin and Shop Collection

b. Tambah Data Kategori

Di menu ini admin dapat menambah data kategori dengan mengisi form kode kategori dan nama kategori kemudian simpan.

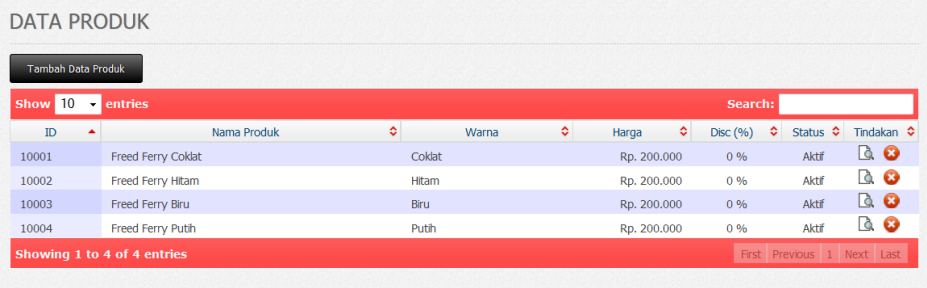


Gambar 5.12 Tambah Data Kategori Pada Riin and Shop Collection

2. Sub Menu Produk

a. Data Produk

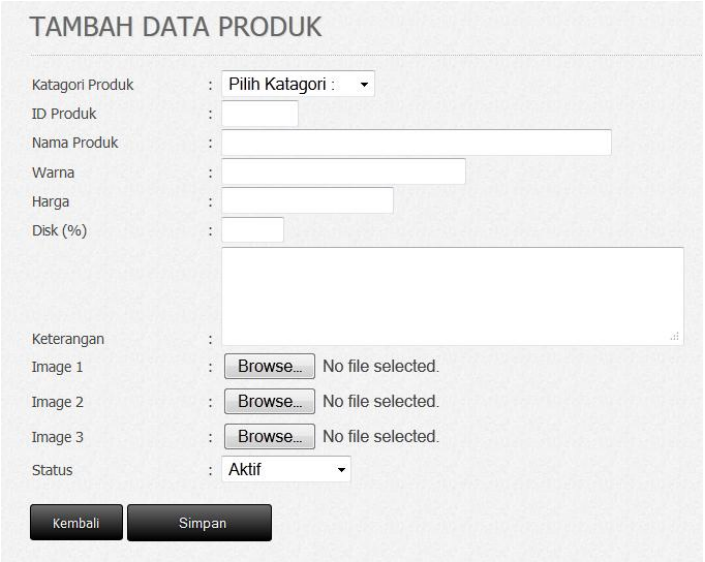
Di menu ini admin dapat melakukan tambah, edit dan hapus data produk



Gambar 5.13 Data Produk Pada Riin and Shop Collection

b. Tambah Data Produk

Di menu ini admin dapat menambah data produk dengan mengisi form di bawah ini kemudian simpan.



Gambar 5.14 Tambah Data Produk Pada Riin and Shop Collection

c. Tambah Detail Data Produk

Di menu ini admin dapat menambah detail data produk dengan mengisi form di bawah ini kemudian simpan.

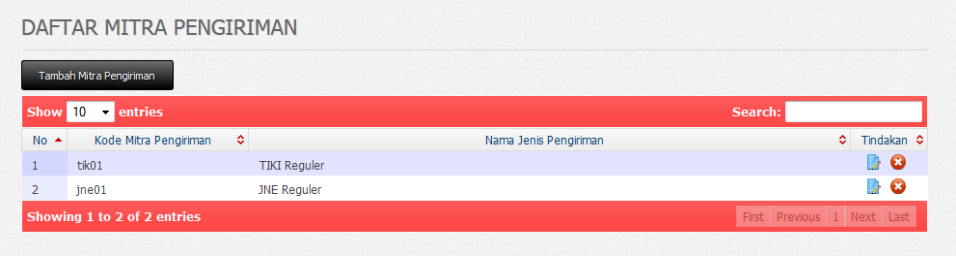


Gambar 5.15 Data Detail Produk Pada Riin and Shop Collection

3. Sub Menu Jasa Pengiriman

a. Daftar Jasa Pengiriman

Di menu ini admin dapat melakukan tambah, edit dan hapus data produk.



Gambar 5.16 Data Jasa Pengiriman Pada Riin and Shop Collection

4. Sub Menu Kota

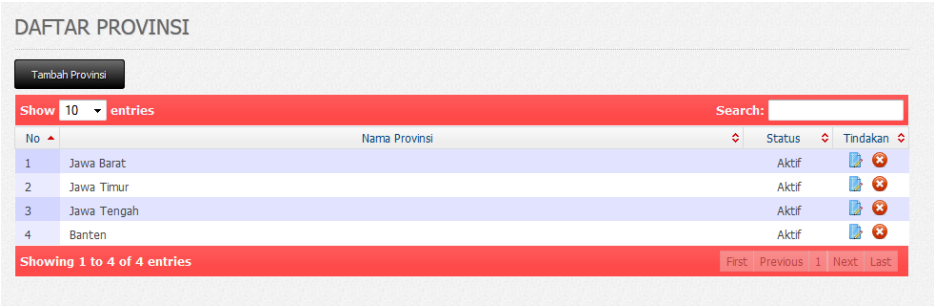
   a. Daftar Kota



Gambar 5.17 Data Kota

5. Sub Menu Provinsi

   a. Daftar Provinsi

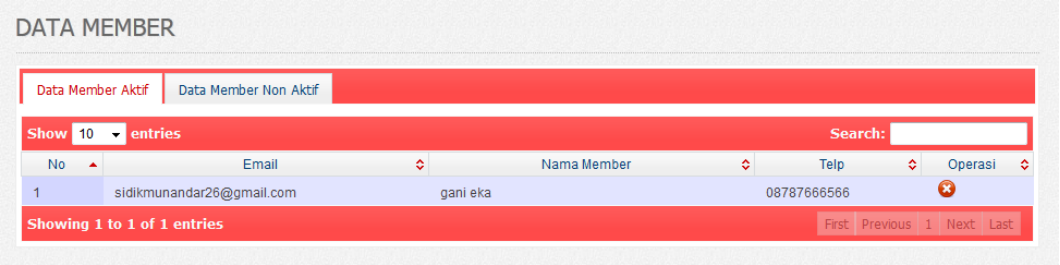


Gambar 5.18 Data Provinsi Pada Riin and Shop Collection

2.4 Menu Data Member

a. Data Member Aktif

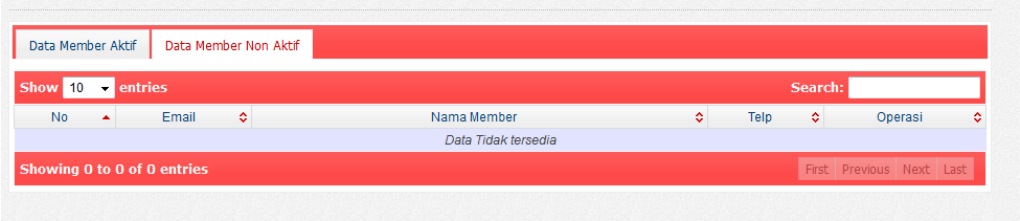
Pada menu ini berfungsi untuk menunjukan data member yang aktif.



Gambar 5.19 Data Member Aktif Pada Riin and Shop Collection

b. Data Member Non Aktif

Pada menu ini berfungsi untuk menunjukan data member yang non aktif.



Gambar 5.20 Data Member Non Aktif Pada Riin and Shop Collection

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. Instalasi *software WampServer* untuk servernya.
2. *Setting* software *WampServer* (php setting dan php extensions)
3. *Setting Apache* modules dan *MySQL*.
4. *Setting database* yang ada pada *PhpMyAdmin*, dengan menuju *link*, *http://localhost/phpmyadmin/*.
5. Setting penyimpanan file data *website* Riin and Shop Collection secara *offline*, dengan menyimpan file *website* Riin and Shop Collection di folder *wamp/www/*
6. *Setting* koneksi *database*, pastikan file *database* yang ada pada folder *website* Riin and Shop Collection sama dengan nama data koneksinya adalah aksesdb.php, dan harus memiliki sintak *database* yang sama.
7. Mengakses menu utama *website* Riin and Shop Collection menggunakan alamat *http://localhost/penjualan_parfum/*
8. Mengakses menu admin *website* Riin and Shop Collection menggunakan alamat *http://localhost/penjualan_parfum/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 client* dan *user admin* pada program aplikasi yang telah dibuat.

Tabel di bawah ini akan menjelaskan mengenai rencana pengujian *Website E-Commerce* Riin and Shop Collection.

Tabel 5.1 Rencana Pengujian Untuk Tampilan *User*

Kasus Uji	Butir Uji	Jenis Pengujian
Pendaftaran Member	Melakukan Pengisian data pendaftaran member.	*Blackbox*
Login Pengguna	Melakukan proses *login* member.	*Blackbox*
Ganti *password*	Melakukan ganti *password* member.	*Blackbox*
Transaksi Pesanan	Memasukkan pesanan data produk ke keranjang belanja.	*Blackbox*
	Melakukan *update* pesanan di keranjang belanja.	*Blackbox*
Input data pengirman	Melakukan penginputan data pengirim.	*Blackbox*

Tabel 5.2 Rencana Pengujian Untuk Tampilan *Administrator*

Kasus Uji	Butir Uji	Jenis Pengujian
Login user admin	Melakukan proses *login* sebagai *admin*	*Blackbox*
Kategori	Melakukan *input*, tambah, *edit*, dan hapus data kategori	*Blackbox*
Produk	Melakukan *input*, tambah, *edit*, dan hapus data produk.	*Blackbox*
Status Transaksi	Melakukan proses perubahan status transaksi pesanan proses, gagal,dan sukses	*Blackbox*
Status Pengiriman	Melakukaan proses perubahan status pengiriman proses dan sukses	*Blackbox*
Data Member	Melakukan proses perubahan status member aktif dan non aktif	

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* Riin and Shop Collection sesuai dengan rencana pengujian yang telah dibuat, antara lain sebagai berikut :

1. Proses Pendaftaran *Member*

Berikut ini adalah hasil pengujian pendaftaran member

Tabel 5.3 Pengujian Pendaftaran *Member*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi lengkap data pendaftaran member	Berhasil mendaftar dan akan muncul alert notifikasi sukses	Berhasil mendaftar dan muncul alert notifikasi sukses	[ ] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi lengkap *form* pendaftaran	Pendaftaran tidak berhasil dan muncul alert pesan untuk mengisi lengkap *form* pendaftaran	Muncul alert pesan "isi terlebih dahulu"	[ ] diterima [ ] ditolak
Mengisi *Form* dengan format data yang tidak *valid*	Muncul alert pesan data berdasarkan *form* yang di isi	Muncul alert pesan validasi berdasarkan *form* yang salah di isi	[ ] diterima [ ] ditolak

2. Proses *Login User Client*

Berikut ini adalah hasil pengujian *login user*.

Tabel 5.4 Pengujian *Login User Client*

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi *Username* dan *Password Valid*	Berhasil *login* dan masuk ke menu utama	*Login* berhasil dan masuk ke halaman menu utama	[ ] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi Kombinasi *Password* dan *Username* dengan data tidak *valid*	Gagal *login* dan muncul pesan "email dan password tidak sesuai silahkan login lagi,"	*Login* gagal dan muncul pesan "email dan password tidak sesuai"	[ ] 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 ulangi password	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 terlebih dahulu"	Gagal mengganti *password* dan muncul pesan "isi terlebih dahulu"	[ ] diterima [ ] ditolak
Mengisi *password* lama dengan data tidak *valid*	Gagal mengganti *password* dan muncul pesan "error, 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 "fields do not match"	Gagal mengganti *password* dan muncul pesan "fields do not match"	[ ] diterima [ ] ditolak

4. Transaksi pesanan

Berikut ini adalah hasil pengujian transaksi pesanan

Tabel 5.6 Pengujian Pemesanan Produk

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukkan data pemesanan ke keranjang belanja dengan menekan button "Tambah keranjang	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

Belanja”			
Mengupdate jumlah pesanan pada field jumlah di keranjang belanja	Jumlah data dikeranjang belanja berhasil di *update*.	Berhasil mengupdate jumlah pesanan produk	[ ] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Belum *login* dan menekan button “Pembayaran”	Pemesanan dilakukan dan Muncul halaman login belum bisa melakukan pembayaran	Muncul halaman login belum bisa melakukan pembayaran	[ ] diterima [ ] ditolak
Memasukkan *field* banyak bukan angka	Muncul alert pesan “not a valid integer”	Muncul alert pesan “not a valid integer”	[ ] diterima [ ] ditolak

5. Input Data Pengiriman

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

Tabel 5.7 Pengujian *Input* Data Pengiriman

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi lengkap data pemesan dengan benar	Muncul alert pesan “Terima kasih,,, Pemesanan anda telah kami simpan, silahkan melak'ukan pembayaran secepat'ya. !!!”	Muncul alert pesan “Terima kasih,,, Pemesanan anda telah kami simpan, silahkan melakukan pembayaran secepat'ya. !!!”	[ ] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Mengosongkan salah satu *field* pemesan	Muncul alert pesan untuk “isi terlebih dahulu”	Muncul alert pesan untuk “isi terlebih dahulu”	[ ] 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* Riin and Shop Collection bebas dari kesalahan *syntax* dan secara fungsional telah mengeluarkan hasil yang *valid* karena proses validasi data dapat berfungsi dengan baik.