BAB V

IMPLEMENTASI DAN PENGUJIAN SISTEM

5.1. Implementasi

Implementasi sistem merupakan kelanjutan dari kegiatan rancangan program dan dapat dipandang sebagai usaha untuk mewujudkan sistem yang telah dirancang serta merupakan tahap meletakkan sistem supatya siap untuk diimplementasikan pada perusahaan.

5.1.1. Batasan Implementasi

Dalam mengimplementasikan perangkat lunak Sistem Informasi Penjualan Pakaian Jadi pada Distro Bloodyfresh Hardwear Bandung, terdapat beberapa hal yang menjadi batasan dalam implementasi sistem, yaitu :

1. Basis data yang digunakan dalam implementasi sistem adalah MySQL
2. Sistem informasi penjualan barang ini diimplementasikan khusus untuk studi kasus pada Distro Bloodyfresh Hardwear Bandung.

5.1.2. Implementasi Perangkat Lunak

Untuk mendukung kelancaran sistem informasi pendaftaran pelatihan, diperlukan perangkat lunak untuk menjalankan program yang telah dirancang. Dimana perangkat lunak berperan sebagai pendukung dari sistem. Perangkat lunak pendukung yang digunakan adalah PHP, XAMPP, dan Web Browser. PHP digunakan sebagai perangkat lunak pengembang karena menyediakan fasilitas yang cukup memadai dalam hal perangkat

lunak yang dapat diakses oleh web browser. XAMPP digunakan sebagai perangkat lunak pendukung dalam pembuatan serta pengembangan dari basis data.

5.1.3. Implementasi Perangkat Keras

Perangkat keras dibutuhkan dalam menjalankan sistem informasi penjualan barang berdasarkan kebutuhan minimal sistem yang harus dipenuhi, terdiri dari :

1. Processor Intel Pentium IV atau setara
2. RAM 512 MB
3. VGA Card Onboard 64 MB
4. Harddisk 40 GB

5.1.4. Implementasi Basis Data

Untuk pembuatan serta pengembangan dari basias data pada sistem informasi pendaftaran pelatihan, perangkat lunak yang digunakan adalah XAMPP server. Berikut adalah implementasi basis data menggunakan XAMPP :

1. Tabel admin

```
CREATE TABLE IF NOT EXISTS `admin` (
`id_admin` int(11) NOT NULL,
`nama_lengkap` varchar(35) COLLATE latin1_general_ci NOT NULL,
`username` varchar(15) COLLATE latin1_general_ci NOT NULL,
`password` varchar(50) COLLATE latin1_general_ci NOT NULL,
PRIMARY KEY (`id_admin`)
```

```
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

2. Tabel category

```
CREATE TABLE IF NOT EXISTS `category` (
 `kode_kategori` varchar(3) COLLATE latin1_general_ci NOTNULL,
 `nama_kategori` varchar(35) COLLATE latin1_general_ci NOT
NULL,
 `foto` varchar(500) COLLATE latin1_general_ci NOT NULL,
 PRIMARY KEY (`kode_kategori`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

3. Tabel confirm

```
CREATE TABLE IF NOT EXISTS `confirm` (
 `no_pesanan` varchar(6) COLLATE latin1_general_ci NOT NULL,
 `no_transfer` varchar(15) COLLATE latin1_general_ci NOT NULL,
 `nama_lengkap` varchar(35) COLLATE latin1_general_ci NOT
NULL,
 `nama_pengirim` varchar(35) COLLATE latin1_general_ci NOT
NULL,
 `status` int(11) NOT NULL,
 PRIMARY KEY (`no_pesanan`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

4. Tabel member

```
CREATE TABLE IF NOT EXISTS `member` (
 `nama_lengkap` varchar(35) COLLATE latin1_general_ci NOT NULL,
 `jenis_kelamin` varchar(6) COLLATE latin1_general_ci NOT NULL,
 `alamat` varchar(100) COLLATE latin1_general_ci NOT NULL,
 `telepon` varchar(12) COLLATE latin1_general_ci NOT NULL,
 `email` varchar(50) COLLATE latin1_general_ci NOT NULL,
 `username` varchar(10) COLLATE latin1_general_ci NOT NULL,
 `password` varchar(50) COLLATE latin1_general_ci NOT NULL,
 PRIMARY KEY (`username`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
```

5. Tabel news

```
CREATE TABLE IF NOT EXISTS `news` (
 `id` int(11) NOT NULL,
 `judul` varchar(50) COLLATE latin1_general_ci NOT NULL,
 `tanggal` varchar(30) COLLATE latin1_general_ci NOT NULL,
 `sumber` varchar(100) COLLATE latin1_general_ci NOT NULL,
 PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
```

6. Tabel ongkos

```
CREATE TABLE IF NOT EXISTS `ongkos` (
  `id` int(11) NOT NULL,
  `kota` varchar(25) COLLATE latin1_general_ci NOT NULL,
  `harga` double NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

7. Tabel pesanan

```
CREATE TABLE IF NOT EXISTS `pesanan` (
  `id` int(11) NOT NULL,
  `no_pesanan` varchar(6) COLLATE latin1_general_ci NOT NULL,
  `total` double NOT NULL,
  `status` int(11) NOT NULL,
  PRIMARY KEY (`no_pesanan`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

8. Tabel pic_news

```
CREATE TABLE IF NOT EXISTS `pic_news` (
  `id` int(11) NOT NULL,
  `foto` varchar(500) COLLATE latin1_general_ci NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

9. Tabel products

```
CREATE TABLE IF NOT EXISTS `products` (
 `kode_kategori` varchar(3) COLLATE latin1_general_ci NOT NULL,
 `kode_produk` varchar(6) COLLATE latin1_general_ci NOT NULL,
 `nama_produk` varchar(35) COLLATE latin1_general_ci NOT NULL,
 `harga` double NOT NULL,
 `foto` varchar(500) COLLATE latin1_general_ci NOT NULL,
 `id` int(11) NOT NULL,
 PRIMARY KEY (`id`)
)ENGINE=MyISAM DEFAULT CHARSET=latin1 COLLATE=latin1_general_ci;
```

10. Tabel transaksi

```
CREATE TABLE IF NOT EXISTS `transaksi` (
 `id` int(11) NOT NULL,
 `tanggal` varchar(30) COLLATE latin1_general_ci NOT NULL,
 `no_pesanan` varchar(5) COLLATE latin1_general_ci NOT NULL,
 `nama_pemesan` varchar(35) COLLATE latin1_general_ci NOT NULL,
 `alamat` varchar(100) COLLATE latin1_general_ci NOT NULL,
 `alamat_tujuan` varchar(50) COLLATE latin1_general_ci NOT NULL,
 `kode_pos` int(11) NOT NULL,
```

```
  `telepon` varchar(12) COLLATE latin1_general_ci NOT NULL,
  `email` varchar(50) COLLATE latin1_general_ci NOT NULL,
  `kode_produk` varchar(6) COLLATE latin1_general_ci NOT NULL,
  `jumlah` int(11) NOT NULL,
  `total` double NOT NULL,
  `status` int(11) NOT NULL,
  `username` varchar(10) COLLATE latin1_general_ci NOT NULL,
  `ukuran` varchar(2) COLLATE latin1_general_ci NOT NULL,
  `kota` varchar(25) COLLATE latin1_general_ci NOT NULL,
  PRIMARY KEY (`id`)
)ENGINE=MyISAMDEFAULTCHARSET=latin1
COLLATE=latin1_general_ci;
```

11. Tabel ukuran

```
CREATE TABLE IF NOT EXISTS `ukuran` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `kode_produk` varchar(6) COLLATE latin1_general_ci NOT NULL,
  `ukuran` varchar(2) COLLATE latin1_general_ci NOT NULL,
  `stock` int(11) NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci AUTO_INCREMENT=190 ;
```

5.1.5. Implementasi Antar Muka

Berikut adalah implementasi antar muka yang telah dirancang dan dibangun dengan menggunakan bahasa pemrograman PHP.

1. Halaman home

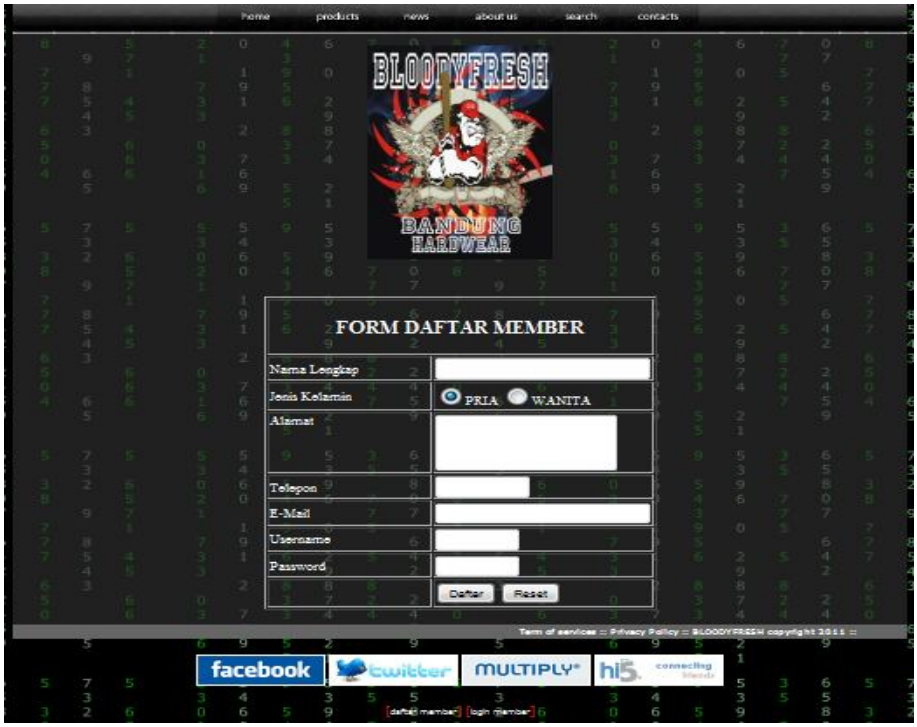
   Tampilan menu utama merupakan tampilan utama dari sistem informasi penjualan barang pada Distro Bloodyfresh Hardwear Bandung.



Gambar 5.1 Halaman home

2. Halaman daftar member

   Halaman untuk pendaftaran member baru yang akan memesan produk, di halaman ini member baru dapat mengisi data diri lengkap, seperti : nama lengkap, jenis kelamin, alamat, telepon, *e-mail*, *username* dan *password*. Setelah data diri lengkap calon member dapat mengklik tombol Daftar untuk menyimpan dan tombol reset untuk mengkoreksi data diri



Gambar 5.2 Halaman Daftar member

3. Halaman Login Member

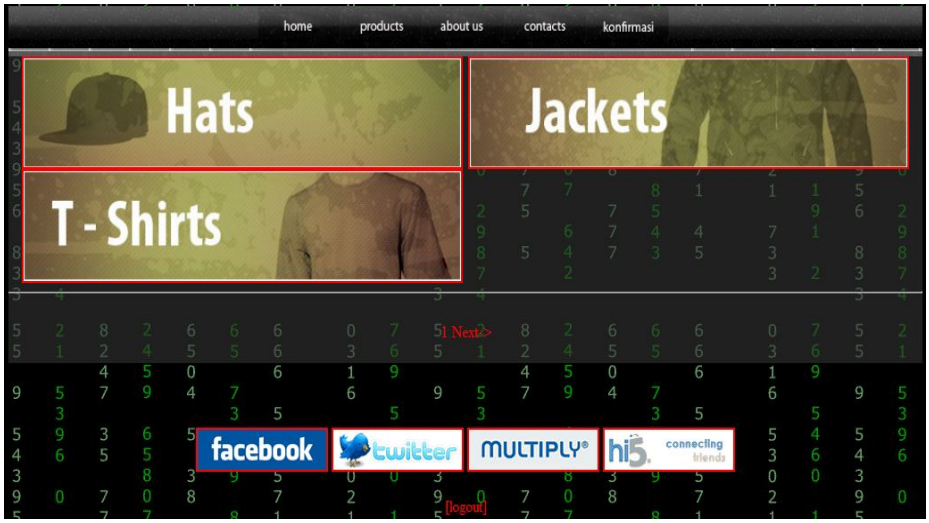
Halaman ini merupakan halaman untuk member yang akan memesan produk, dengan mengisikan username dan password member yang benar sesuai data yang telah disimpan. Kemudian klik tombol Login untuk dapat memesan produk.



Gambar 5.3 Halaman Login member

4. Halaman Products

Halaman ini merupakan halaman yang menampilkan produk-produk yang tersedia yang berdasarkan tiga kategori produk, yaitu : Hats, T-Shirt dan Jacket, member cukup mengklik gambar kategori untuk melakukan pemesanan.



Gambar 5.4 Halaman Produk

5. Halaman Pemesanan Produk

Halaman ini merupakan halaman untuk memesan produk yang telah dipilih, yang menampilkan nomor peasanan, informasi produk, dan data member, lalu member mengisi alamat tujuan, kode pos, ukuran produk, kota tujuan dan jumlah produk yang akan dipesan sebagai daftar pesanan, lalu klik tombol simpan.



Gambar 5.5 Halaman Pemesanan Produk

6. Tampilan daftar pesanan

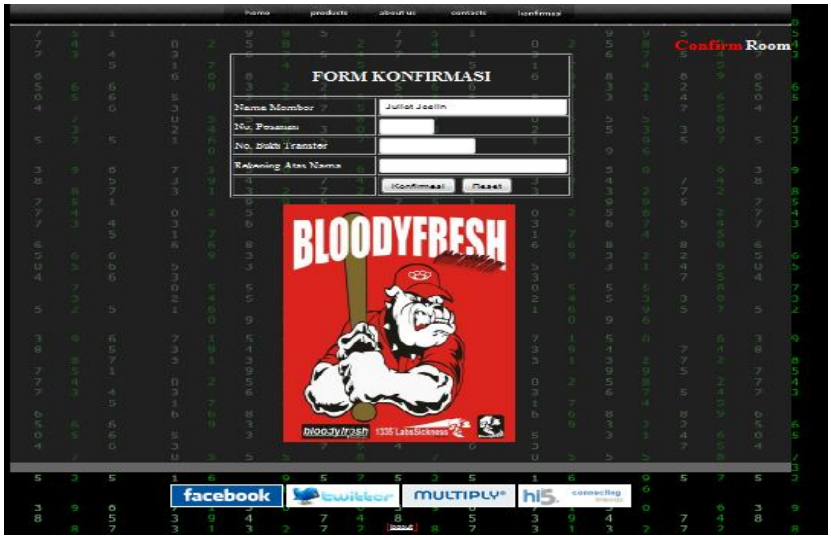
Pada tampilan ini menampilkan peringatan mengenai cara pembayaran melalui rekening antar bank dan juga menampilkan daftar pesanan, yang meliputi : tanggal pesanan, nomor pesanan, data member dan total harga yang harus dibayar. Klik DEAL bila akan memesan dan batl untuk membatalkan.



Gambar 5.6 Tampilan Daftar pesanan

7. Tampilan Konfirmasi pembayaran

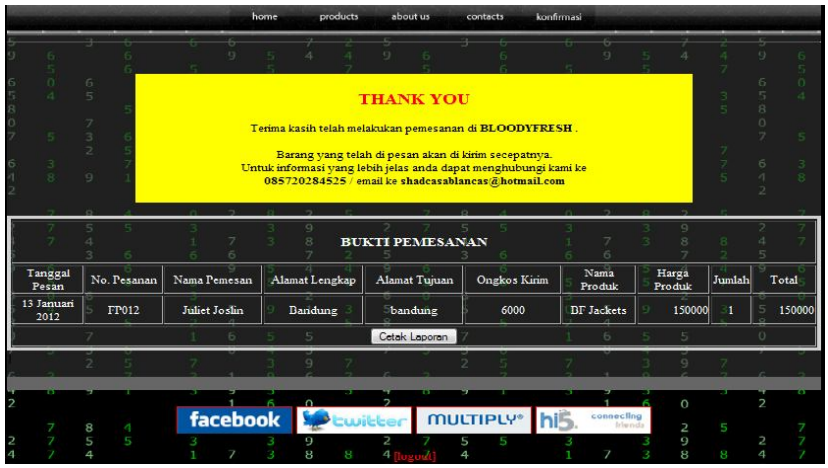
Merupakan tampilan untuk mengkonfirmasi bahwa member telah melakukan pembayaran, dengan mengisikan nomor pesanan sebelumnya, nomor bukti transfer, dan rekening atas nama pengirim. Kilik konfirmasi jika telah melakukan pembayaran.



Gambar 5.7 Tampilan konfirmasi pembayaran

8. Bukti pemesanan

Merupakan bukti bahwa member telah melakukan pembayaran.



Gambar 5.8 Bukti pemesanan

9. List Pesanan

Merupakan halaman pada admin yang menampilkan daftar pesanan dari member, dengan status “konfirmasi” jika telah melakukan pembayaran dan “belum konfirmasi” jika belum melakukan pembayaran.



Gambar 5.9 List Pesanan

10. List Penjualan

Merupakan tampilan laporan penjualan dan pendapatan.



Gambar 5.10 List Penjualan

5.1.6. Implementasi Instalasi Program

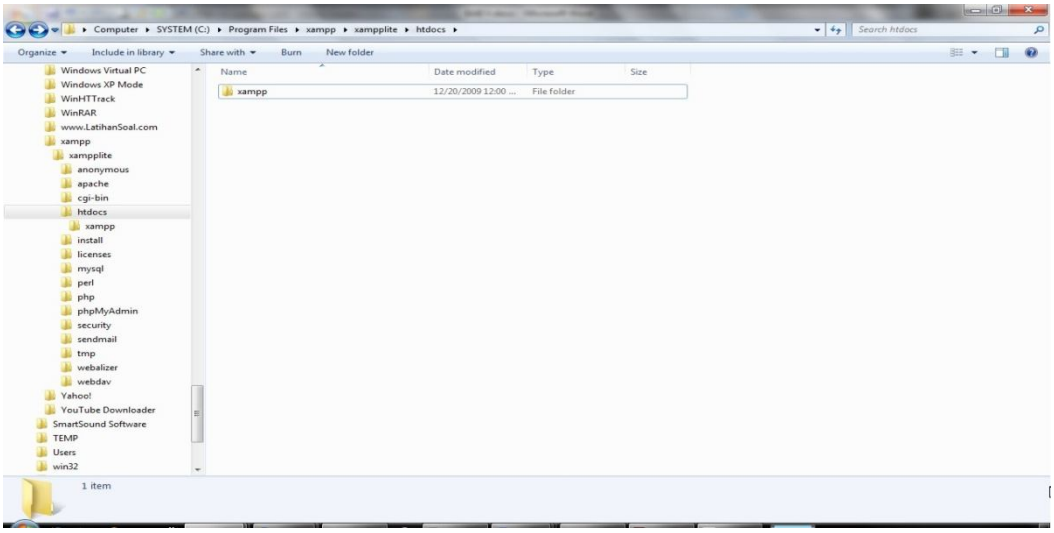
Berikut adalah tahapan-tahapan dalam instalasi program Sistem Informasi Penjualan Barang pada Distro Bloodyfresh Hardwear Bandung :

1. Terlebih dahulu install XAMPP sebagai perangkat lunak pendukung. Cara penginstalannya tidak jauh berbeda dengan cara penginstalan perangkat lunak pada umumnya.



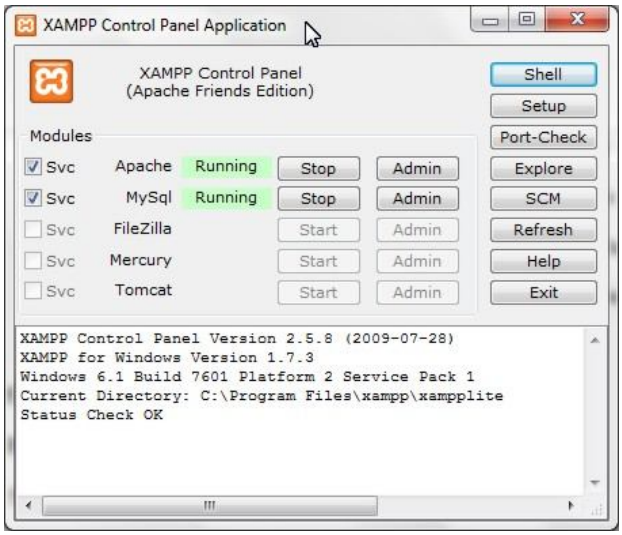
Gambar 5.11 Tampilan *Installer* XAMPP

2. Kemudian buka subfolder htdocs yang berada pada folder xampp, dan copy folder yang berisi data Sistem Informasi Penjualan Barang yang berekstensi .php kedalamnya.



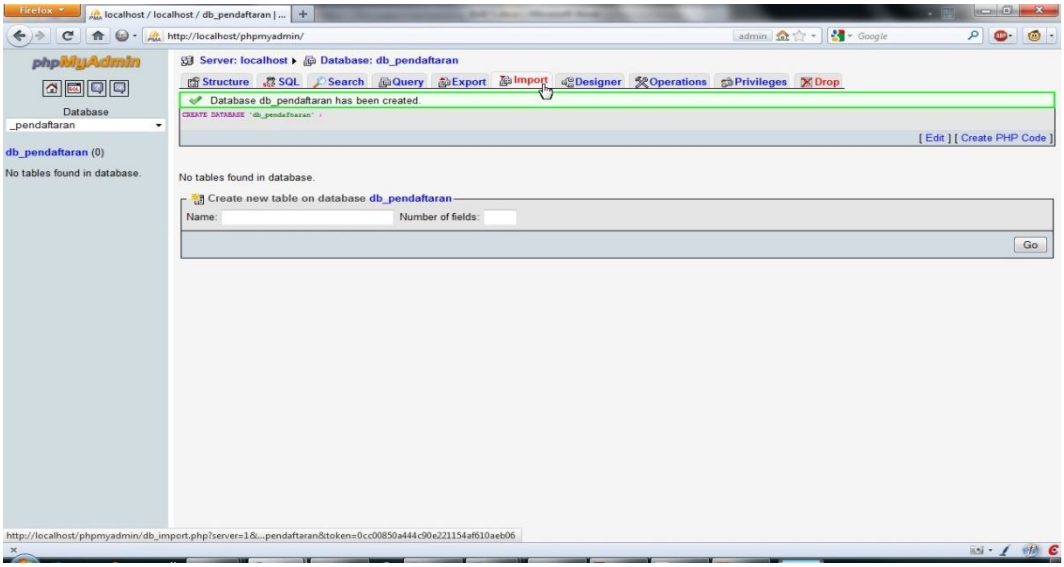
Gambar 5.12 Tampilan subfolder htdocs pada folder xampp

3. Buka xampp control panel dan pastikan Apache dan MySql dalam kondisi *running*.



Gambar 5.13 Tampilan XAMPP *Control Panel*

4. Setelah itu buka web browser, dan ketik "http://localhost/phpmyadmin/", dan buat database bloodyfresh. Kemudian import tabel-tabel yang sebelumnya telah dibuat. Ke dalam database blooodyfresh.



Gambar 5.14 Tampilan halaman phpMyAdmin

5.1.7. Penggunaan Program

Program ini digunakan dengan tujuan untuk membantu proses penjualan, input dan mengelola data barang, dan laporan penjualan. Program ini hanya bisa diakses oleh dua pengguna yaitu admin dan member.

5.2. Pengujian

Pengujian program adalah pengujian dimana pengguna program memasukkan data ke dalam sistem agar dapat diproses sistem tersebut, dan untuk mempermudah user. Pengujian dilakukan untuk menjamin kualitas serta untuk mengetahui kekurangan dari program. Metode pengujian program yang digunakan adalah dengan menggunakan metode *black box*. Metode ini hanya bekerja pada fungsional program, tidak pada alur internal program.

5.2.1. Rencana Pengujian

Berikut adalah rencana pengujian yang akan dilakukan pada Sistem Informasi Penjualan Barana pad Distro Bloodyfresh Hardwear Bandung dengan menggunakan metode *black box* yang terdapat pada Tabel 5.1 :

Tabel 5.1 Rencana Pengujian

Kelas	Butir Uji	Jenis Pengujian
Login User	Member	BlackBox
	Admin	BlackBox
Input / Ubah	Input data member	BlackBox
	Input pesanan	BlackBox
	Input konfirmasi pembayaran	BlacBox
	Input data barang	BlackBox
	Ubah data barang	BlackBox
Pencetakan	Pencetakan bukti pemesanan	BlackBox
	Pencetakan laporan penjualan	BlackBox

5.2.2. Kasus dan Hasil Pengujian

Berikut adalah hasil pengujian yang dilakukan sesuai dengan rencana pengujian diantaranya :

1. Pengujian Login

   Berikut adalah pengujian pada proses login yang terdapat pada Tabel 5.2 :

Tabel 5.2 Pengujian login

Kasus dan hasil pengujian				
Form di uji	**Data masukan**	**Yang di harapkan**	**Hasil pengamatan**	**Kesimpulan**
Login	Masukkan username dan password yang benar	Masuk ke halaman utama	Berhasil login	[X] diterima [ ] ditolak
	Masukkan username dan password yang salah	Masuk ke halaman utama	Akan muncul pemberitahuan "username dan password salah"	[ ] diterima [X] ditolak

2. Pengujian input data member

Tabel 5.3 Pengujian Input Data Member

Kasus dan hasil pengujian				
Form di uji	**Data masukan**	**Yang di harapkan**	**Hasil pengamatan**	**Kesimpulan**
Input Data	Masukkan	Data akan	Muncul	

Member	data yang lengkap pada form yang tersedia	masuk ke dalam database	pemberitahuan “data berhasil disimpan”	[X] diterima [ ] ditolak
	Masukkan data yang tidak lengkap pada form yang tersedia	Data tidak akan masuk ke dalam database ,kembali ke form daftar member	Akan muncul pemberitahuan "data tidak lengkap, cek kembali"	[ ] diterima [X] ditolak

3. Pengujian Input pesanan

Tabel 5.4 Pengujian Input Pesanan

Kasus dan hasil pengujian				
Form di uji	**Data Masukan**	**Yang di harapkan**	**Hasil pengamtan**	**Kesimpulan**
Input Pesanan	Masukkan data yang lengkap pada form yang tersedia	Data akan masuk ke dalam database	Muncul pemberitahuan “data berhasil disimpan”	[X] diterima [ ] ditolak
	Masukkan data yang tidak	Data tidak akan masuk	Akan muncul pemberitahuan	[ ] diterima

	lengkap pada form yang tersedia	ke dalam databse kembali ke form pesanan	"data tidak lengkap"	[X] ditolak

4. Pengujian Input konfirmasi pembayaran

Tabel 5.5 Pengujian Input Konfirmasi Pembayaran

Kasus dan hasil pengujian				
Form di uji	**Data Masukan**	**Yang di harapkan**	**Hasil pengamatan**	**Kesimpulan**
Input konfirmasi pembayaran	Masukkan data yang lengkap pada form yang tersedia	Data akan masuk ke dalam database	Muncul pemberitahuan “data berhasil disimpan”	[X] diterima [ ] ditolak
	Masukkan data yang tidak lengkap pada form yang tersedia	Data tidak akan masuk ke dalam databse kembali ke form konfirmasi	Akan muncul pemberitahuan "data tidak lengkap"	[ ] diterima [X] ditolak

5. Pengujian Input Data Barang

Tabel 5.6 Pengujian Input data Barang

Kasus dan hasil pengujian				
Form di uji	**Data Masukan**	**Yang di harapkan**	**Hasil pengamtan**	**Kesimpulan**
Input data barang	Masukkan data yang lengkap pada form yang tersedia	Data akan masuk ke dalam database	Muncul pemberitahuan “data berhasil disimpan”	[X] diterima [ ] ditolak
	Masukkan data yang tidak lengkap pada form yang tersedia	Data tidak akan masuk ke dalam databse kembali ke form data barang	Akan muncul pemberitahuan "data tidak lengkap"	[ ] diterima [X] ditolak

6. Pengujian Ubah Data Barang

Tabel 5.6 Pengujian Ubah Data Barang

Kasus dan hasil pengujian				
Form di uji	**Data Masukan**	**Yang di harapkan**	**Hasil Pengamtan**	**Kesimpulan**
Ubah Data Barang	Ubah data barang sesuai dengan data yang akan diubah	Data pada database akan diperbaharui	Muncul pemberitahuan “data berhasil diubah”	[X] diterima [ ] ditolak

5.2.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan kasus uji sample di atas, maka dapat ditarik kesimpulan bahwa pada proses pengisian data di setiap form dapat berjalan dengan serta menghasilkan keluaran yang diharapkan.