BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Implementasi merupakan kelanjutan dari perancangan sistem dan. Langkah-langkah dari proses implementasi adalah urutan dari kegiatan awal sampai kegiatan akhir yang harus dilakukan dalam mewujudkan sistem yang dirancang. Hasil dari tahapan implementasi ini adalah suatu sistem pengolahan data yang sudah berjalan dengan baik.

4.1.1 Perangkat keras yang digunakan

Spesifikasi perangkat keras yang digunakan untuk menguji aplikasi adalah sebagai berikut:

1. *Processor*AMDTurion II *Dual-core Mobile* 500 2,4 GHz
2. RAM sebesar 4GB
3. Kapasitas *free* Hardisk 40 GB
4. Monitor VGA dengan revolusi 1280 x 800
5. Perangkat untuk akses jaringan internet.
6. Mouse, keyboard

4.1.2 Perangkat Lunak yang digunakan

Spesifikasi perangkat lunak (*Software*) yang dipasang pada sistem komputer yang digunakan untuk penerapan aplikasi adalah sebagai berikut :

1. Sistem Operasi Window 7 Home Premium 64 bit
2. *MysQL* untuk *database*
3. WebServer
4. Web browser *Google Chrome*

4.2 Implementasi Basis Data

Pembuatan basis data dilakukan dengan mengunakan aplikasi pemrograman MySQL. Implementasi basis data dalam bahasa SQL adalah sebagai berikut :

1. Pembuatan Database

```
CREATE DATABASE `bbshop` ;
```

2. Tabel *user*

```
CREATE TABLE `user` (
  `id_user` int(11) NOT NULL auto_increment,
  `id_kota` int(11) NOT NULL,
  `nama_user` varchar(50) NOT NULL,
  `alamat_user` text NOT NULL,
  `ttl_user` date NOT NULL,
  `telp_user` varchar(20) NOT NULL,
  `kodepos_user` varchar(6) NOT NULL,
  `email_user` varchar(50) NOT NULL,
  `password_user` varchar(32) NOT NULL,
  `verificationcode_user` varchar(32) NOT NULL,
  `status_user` enum('0','1') NOT NULL,
  `hak_akses` enum('Administrator','User') default 'User',
```

```
  PRIMARY KEY  (`id_user`),

  UNIQUE KEY `email_member` (`email_user`)

  KEY 'id_kota' ('id_kota')

) ALTER TABLE `user`

ADD CONSTRAINT `id_kota_FK` FOREIGN KEY (`id_kota`) REFERENCES
`kota` (`id_kota`);

ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

3. Tabel Barang

```
CREATE TABLE `barang` (

  `id_barang` int(4) unsigned zerofill NOT NULL auto_increment,

  `id_kategori` int(11) NOT NULL,

  `nama_barang` varchar(50) NOT NULL,

  `harga_barang` int(11) NOT NULL,

  `deskripsi_barang` text NOT NULL,

  `diskon_barang` int(11) NOT NULL,

  `rating_barang` float NOT NULL,

  `voterrating_barang` int(11) NOT NULL,

  `viewcounter_barang` int(11) NOT NULL,

  `gambar_barang` varchar(200) NOT NULL,

  PRIMARY KEY  (`id_barang`),

  KEY `id_kategori` (`id_kategori`)

)

ALTER TABLE `barang`

ADD CONSTRAINT `id_kat_FK` FOREIGN KEY (`id_kategori`) REFERENCES
`kategori` (`id_kategori`);

ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

4. Tabel Kategori

```
CREATE TABLE `kategori`

(

  `id_kategori` int(11) NOT NULL auto_increment,

  `kategori` varchar(40) NOT NULL,

  PRIMARY KEY  (`id_kategori`)

)

ENGINE=InnoDB  DEFAULT CHARSET=latin1; AUTO_INCREMENT=17;
```

5. Tabel Barangdetail

```
CREATE TABLE `barangdetail` (

  `id_barangdetail` int(11) NOT NULL auto_increment,

  `id_barang` int(4) unsigned zerofill NOT NULL,

  `id_warna` int(11) default NULL,

  `id_bahan` int(11) NOT NULL,

  `tanggal_barangdetail` datetime NOT NULL,

  `stok_barangdetail` int(11) NOT NULL,

  `berat_barangdetail` float NOT NULL,

  `promo_barangdetail` enum('0','1') NOT NULL,

  PRIMARY KEY  (`id_barangdetail`),

  KEY `id_barang` (`id_barang`),

  KEY `id_warna` (`id_warna`),

  KEY `id_bahan` (`id_bahan`)

 )

 ALTER TABLE `barangdetail`

 ADD CONSTRAINT `id_brg_FK` FOREIGN KEY (`id_barang`) REFERENCES
 `barang` (`id_barang`);

 ADD CONSTRAINT `idwarna_FK` FOREIGN KEY (`id_warna`) REFERENCES
```

```
 `warna` (`id_warna`);

 ADD CONSTRAINT `idbhn_FK` FOREIGN KEY (`id_bahan`) REFERENCES
 `bahan` (`id_bahan`);

 ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

6. Tabel Pembelian

```
CREATE TABLE `pembelian` (

  `id_pembelian` int(11) NOT NULL auto_increment,

  `id_detailpembelian` int(11) default NULL,

  `id_user` int(11) default NULL,

  `id_barangdetail` int(11) NOT NULL,

  `harga_temp` int(11) NOT NULL,

  `stok_temp` int(11) default NULL,

  `berat_temp` float NOT NULL,

  `status_retur` enum('0','1') NOT NULL,

  `session_id` varchar(32) NOT NULL,

   PRIMARY KEY  (`id_pembelian`)

   KEY 'id_detailpembelian' ('id_detailpembelian')

) ALTER TABLE `pembelian`

ADD CONSTRAINT `iddp_FK` FOREIGN KEY (`id_detailpembelian`)
REFERENCES `detailpembelian` (`id_detailpembelian`)

ADD CONSTRAINT `idmember_FK` FOREIGN KEY (`id_user`) REFERENCES
`user` (`id_user`)

ADD CONSTRAINT `idbrgd_retur_FK` FOREIGN KEY (`id_barangdetail`)
REFERENCES `barangdetail` (`id_barangdetail`);

ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

7. Tabel Detailpembelian

```
CREATE TABLE `detailpembelian` (

  `id_detailpembelian` int(11) NOT NULL auto_increment,

  `tanggal_detailpembelian` date NOT NULL,

  `jenis_pembayaran` enum('paypal','rekening') default NULL,

  `status_pengiriman`
enum('dipesan','dikonfirmasi','dibayar','dikirim','diterima')  NOT
NULL,

  `keterangan_detailpembelian` text NOT NULL,

  `nama_pemesan` varchar(50) NOT NULL,

  `alamat_pemesan` text NOT NULL,

  `email_pemesan` varchar(50) NOT NULL,

  `no_telp_pemesan` varchar(20) NOT NULL,

  `kodepos_pemesan` varchar(6) NOT NULL,

  `nama_bank_pemesan` varchar(50) NOT NULL,

  `no_transaksi_pemesan` varchar(20) NOT NULL,

  `no_resi_pemesan` varchar(32) NOT NULL,

  `id_rekening` int(11) NOT NULL,

  `id_ongkir` int(11) NOT NULL,

  `status_pemesanan` enum('ok','cancel') NOT NULL,

  `session_id` varchar(32) NOT NULL,

  PRIMARY KEY  (`id_detailpembelian`)

) ALTER TABLE `detailpembelian`

ADD CONSTRAINT `id_rek _FK` FOREIGN KEY (`id_rekening`) REFERENCES
`rekening` (`id_rekening`)

ADD   CONSTRAINT   `id_ongkir_FK`   FOREIGN   KEY   (`id_ongkir`)
REFERENCES `onkgkir` (`id_ongkir`)

ENGINE=InnoDB  DEFAULT CHARSET=latin1
```

8. Tabel Jenispengiriman

```
CREATE TABLE `jenispengiriman` (

  `id_jenispengiriman` int(11) NOT NULL auto_increment,

  `nama_jenispengiriman` varchar(50) NOT NULL,

  `deskripsi_jenispengiriman` text NOT NULL,

  PRIMARY KEY  (`id_jenispengiriman`)

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

9. Tabel Ongkos Kirim

```
CREATE TABLE `ongkir` (

  `id_ongkir` int(11) NOT NULL auto_increment,

  `id_kota` int(11) NOT NULL,

  `id_jenispengiriman` int(11) NOT NULL,

  `harga_ongkir` int(11) NOT NULL,

  PRIMARY KEY  (`id_ongkir`)

)

ADD CONSTRAINT `id_kota_FK` FOREIGN KEY (`id_kota`) REFERENCES
`kota` (`id_kota`);

ADD CONSTRAINT `idjp_FK` FOREIGN KEY (`id_jenispengiriman`)
REFERENCES `jenispengiriman` (`id_jenispengiriman`);

ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

10. Tabel Provinsi

```
CREATE TABLE `provinsi`

(

  `id_provinsi` int(11) NOT NULL auto_increment,

  `nama_provinsi` varchar(100) default NULL,
```

```
  PRIMARY KEY  (`id_prov`)

)

ENGINE=InnoDB  DEFAULT CHARSET=latin1; AUTO_INCREMENT=34;
```

11. Tabel Kota

```
CREATE TABLE `kota` (

  `id_kota` int(10) NOT NULL auto_increment,

  `id_provinsi` int(10) default NULL,

  `nama_kota` varchar(50) default NULL,

  `kota` varchar(20) default NULL,

  UNIQUE KEY `kota#PX` (`id_kota`),

  KEY `id_provinsi` (`id_provinsi`)

 )

 ADD CONSTRAINT `id_provinsi` FOREIGN KEY (`id_provinsi`)
 REFERENCES `provinsi` (`id_provinsi`);

 ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

12. Tabel Gambar

```
CREATE TABLE `gambar` (

  `id_gambar` int(11) NOT NULL auto_increment,

  `id_barang` int(4) unsigned zerofill NOT NULL,

  `nama_gambar` varchar(200) NOT NULL,

  `profile_gambar` enum('0','1') NOT NULL,

  PRIMARY KEY  (`id_gambar`)

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

13. Tabel Konfirmasi Pembayaran

```
CREATE TABLE `konfirmasipembayaran` (
  `id_konfirmasipembayaran` int(11) NOT NULL auto_increment,
  `id_detailpembelian` int(11) NOT NULL,
  `nama_bank_pengirim` varchar(25) NOT NULL,
  `no_rekening_pengirim` varchar(25) NOT NULL,
  `total_bayar` int(11) NOT NULL,
  `tanggal_transfer` date NOT NULL,
  PRIMARY KEY  (`id_konfirmasipembayaran`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

14. Tabel Kurs

```
CREATE TABLE `kurs` (
  `id_kurs` int(11) NOT NULL auto_increment,
  `kode_kurs` varchar(5) NOT NULL,
  `symbol_kurs` varchar(5) NOT NULL,
  `harga_kurs` int(11) NOT NULL,
  `deskripsi_kurs` text NOT NULL,
  `status_kurs` enum('0','1') NOT NULL,
  PRIMARY KEY  (`id_kurs`),
  UNIQUE KEY `nama_currency` (`kode_kurs`,`symbol_kurs`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

15. Tabel Retur

```
CREATE TABLE `retur`
(
  `id_retur` int(11) NOT NULL auto_increment,
```

```
  `id_user` int(11) NOT NULL,

  `jasa_kirim` varchar(30) NOT NULL,

  `no_kirim` varchar(20) NOT NULL,

  `tgl_retur` date NOT NULL,

  `total_retur` int(11) NOT NULL,

  `status_retur` varchar(10) NOT NULL,

  PRIMARY KEY  (`id_retur`),

  KEY `idmember_retur_FK` (`id_user`)

) ENGINE=InnoDB  DEFAULT CHARSET=latin1; AUTO_INCREMENT=6 ;

ADD CONSTRAINT `idmember_retur_FK` FOREIGN KEY (`id_user`)
REFERENCES `member` (`id_user`);
```

16. Tabel Detail Retur

```
CREATE TABLE `detail_retur`

(

  `id_retur` int(11) NOT NULL,

  `idpembelian ` int(11) NOT NULL,

  `idbarang` int(8) NOT NULL,

  `size` varchar(10) NOT NULL,

  `qty` int(11) NOT NULL,

  `harga` int(11) NOT NULL,

  `diskon` int(11) NOT NULL,

  `komplain` text NOT NULL,

  KEY `idretur_FK` (`id_retur`),

  KEY `idbarang_retur_FK` (`idbarang`)

)

ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `detail_retur`
```

```
ADD CONSTRAINT `idbarang_retur_FK` FOREIGN KEY (`idbarang`)
REFERENCES `barang` (`id_barang`),

ADD CONSTRAINT `idretur_FK` FOREIGN KEY (`id_retur`) REFERENCES
`retur` (`id_retur`);
```

17. Tabel Testimoni Barang

```
CREATE TABLE `testi` (

  `id_testi` int(11) NOT NULL auto_increment,

  `Id_user` int(11) NOT NULL,

  `tgl_testi` DATETIME,

  `testi` text NOT NULL,

  `id_barang` int(4) unsigned zerofill NOT NULL,

  PRIMARY KEY  (`id_testi`)

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

18. Tabel Pengadaan Barang

```
CREATE TABLE `pengadaanbarang` (

  `id_pb` int(4) NOT NULL auto_increment,

  `id_barangdetail` int(4) NOT NULL,

  `jumlah` int(4) NOT NULL,

  `harga` int(11) NOT NULL,

  `tgl_masuk` date NOT NULL,

  `rekanan` varchar(100) NOT NULL,

  `id_user` int(11) NOT NULL,

  PRIMARY KEY  (`id_pb`)) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

19. Tabel Promo

```
CREATE TABLE `promo` (
  `id_barang` int(4) unsigned zerofill NOT NULL,
  `cat_promo` enum('HOT PRODUCT','SALE','NEW PRODUCT') NOT NULL,
  `gambar` varchar(100) NOT NULL,
  `status_promo` tinyint(1) NOT NULL default '0',
  UNIQUE KEY `id_barang` (`id_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

20. Tabel Rekening

```
CREATE TABLE `rekening` (
  `id_rekening` int(11) NOT NULL auto_increment,
  `nama_rekening` varchar(50) NOT NULL,
  `bank_rekening` varchar(50) NOT NULL,
  `cabang_rekening` varchar(100) NOT NULL,
  `no_rekening` varchar(32) NOT NULL,
  `gambar_rekening` varchar(200) NOT NULL,
  PRIMARY KEY  (`id_rekening`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
```

21. Tabel Bahan

```
CREATE TABLE `bahan` (
  `id_bahan` int(11) NOT NULL auto_increment,
  `nama_bahan` varchar(50) NOT NULL,
  `deskripsi_bahan` text NOT NULL,
  PRIMARY KEY  (`id_bahan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

22. Tabel Warna

```
CREATE TABLE `warna` (
  `id_warna` int(11) NOT NULL auto_increment,
  `nama_warna` varchar(50) NOT NULL,
  `format_warna` varchar(10) default NULL,
  PRIMARY KEY  (`id_warna`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

4.3 Implementasi Antarmuka

Tabel 4.1 Implementasi Antarmuka Pengunjungdan *Member*

Menu	Deskripsi	Nama File
Halaman Utama	Menampilkan barang terbaru, special Offer dan Info terbaru perusahaan.	home.php
Registrasi *Member*	Digunakan untuk melakukan pendaftaran *member*baru.	Register.php
Masuk *member*	Digunakan Digunakan sebagai halaman masuk *member*.	masuk.Php
About	Digunakan untuk melihat data profil *blacbox-shop*.	about.php
Pengaturan akun	Digunakan untuk melihat data *member*, dan Edit *member*	edit.account.php
History Pembelian	Digunakan Untuk melihat history pembelian, memilih metode pambayaran, melihat status transaksi, serta history retur	history.php

Menu	Deskripsi	Nama File
Retur barang	Digunaka oleh *member* untuk melakukan kegiatan retur barang yang telah dibeli.	retur.php
Cara Berbelanja	Digunakan untuk melihat cara berbelanja di website *Blacbox-shop*.	cara_belanja. Php
Cara Pembayaran	Digunakan untuk melihat cara pembayaran di *Blacbox-shop.*	cara_bayar.php
ToS (*Term of Service)*	Digunakan untuk melihat syarat dan kondisi yang berlaku dalam proses transaksi di website *Blacbox-shop*.	tos.php
Detail Barang	Digunakan Untuk melihat detail dari barang yang dipilih.	detailbarang.php
Keranjang Belanja / Pemesanan	Digunakan untuk melihat, merubah, membatalkan daftar barang yang telah dipilih.	cart.php
Form Pengiriman	Digunakan untuk mengisi alamat data penerima pengiriman barang yang dibeli.	pengiriman.php
Pembayaran	Digunakan untuk memilih metode pembayaran, menunda dan membatalkan proses pembelian.	bayar.php
Pencarian Barang	Digunakan untuk menampilkan hasil pencarian barang.	search. Php
Keluar	Digunakan *member*untuk keluar dari akunnya.	keluar.php

Tabel 4.2 Implementasi Antarmuka *Admin*

Menu	Deskripsi	Nama File
Masuk *Admin*	Digunakan sebagai halaman masuk *admin*	masuk.php
Halaman Utama	Menampilkan informasi tentang halaman *admin*istrator.	index.php
Data *Admin*	Digunakan jika *admin* inginmerubah data nya.	*admin*.php
Databahan	Digunakan untuk melihat, menambah, merubah dan menghapus data bahan.	bahan.php
Data Kategori	Digunakan untuk melihat, menambah, merubah dan menghapus data kategori.	kategori.php
Data Barang	Digunakan untuk melihat, menambah, merubah dan menghapus data barang.	barang.php
Detail Barang	Digunakan untuk melihat, menambah, merubah dan menghapus data detail dari suatu barang.	detail_barang.php
Data *Member*	Digunakan untuk melihat, menambah, merubah dan menghapus data *member*.	*member*. php
Data Provinsi	Digunakan untuk melihat, menambah, merubah dan menghapus data provinsi	provinsi.php
Data Kota	Digunakan untuk melihat, menambah, merubah dan menghapus data kabupaten dan kota serta ongkos kirimnya.	kota.php
Data Transaksi	Digunakan untuk memproses transaksi darimulai konfirmasi pembayaran, pengiriman sampai selesai.	transaksi.php

Menu	Deskripsi	Nama File
Retur	Digunakan untuk memproses transaksi retur barang.	retur.php
Laporan	Digunakan untuk melihat data laporan harian,bulanan maupun tahunan setra laporan data barang dalam bentuk PDF.	laporan.php
Promo	Digunakan untuk mangatur *banner* di halaman awal situs	promo.php
Buku Tamu	Digunakan Untuk mengelola komentar saran, dan kritik serta feedback baik dari pengunjung maupun *member*.	kontak.php
Keluar	Digunakan *admin*istrator Untuk dapat keluar dari aplikasi	keluar.php

4.4 Pengujian Perangkat Lunak Aplikasi

Dalam pengujian kebenaran program, maka program harus diuji dengan data tes yang telah dipersiapkan sebelumnya, sehingga bila tidak sesuai dengan kebutuhan, kesalahan sintaks, kesalahan logika dan kesalahan pada saat *runtime* dapat diatasi segera.

Pengetesan program dilakukan dengan dua cara, yaitu :

1. Pengetesan setiap modul program secara terpisah

   Pengetesan terhadap setiap modul program dimaksudkan apabila ada kesalahan akan lebih mudah dan cepat dalam memperbaikinya.

2. Pengetesan modul program yang telah terintegrasi

Setelah seluruh modul dites dan tidak terdapat kesalahan, maka selanjutnya dilakukan tes secara keseluruhan menjadi satu unit program. Hal ini dimaksudkan untuk meyakinkan bahwa semua modul dapat terintegrasi tanpa mengalami kesalahan.

4.4.1 Pembentukan Data Awal

Untuk membentuk data awal yang akan dibuat sebagai file dalam komputer perlu dilakukan pengumpulan data yang kemudian akan dimasukkan kedalam sistem komputer. Data awal terdiri dari :

1. Data *Admin*
2. Data Bahan
3. Data Kategori
4. Data Barang
5. Data Provinsi
6. Data kota
7. Data *Member*
8. Data promo

4.4.2 Uji Coba dan Evaluasi Sistem

Pengujian sistem dilakukan setelah implementasi sistem yang bertujuan untuk mengetahui sejauh mana sistem tersebut dapat memenuhi kebutuhan bagi pengguna sistem juga untuk menguji kemampuan sistem dalam mengolah data

yang ada. Pengujian ini diarahkan pada keterkaitan masing-masing komponen dalam sistem.

4.4.3 Pengujian *Alpha*

Pengujian a*lpha* dilakukan dengan menggunakan metode *black box*. Untuk pengujian a*lpha* ini yaitu pada pengujian sebagai pengguna.

4.4.3.1 Skenario Pengujian Alpha

Tabel 4.3 Skenario Aplikasi Pengujian Untuk *Member*

Kelas Uji	Butir Uji	Jenis Pengujian
Registrasi *member*	Menampilkan *form*registrasi *member*	*Black Box*
	Isi *form*registrasi	*Black Box*
	Simpan data *member* baru	*Black Box*
Masuk	Isi data masuk	*Black Box*
	Verifikasi email	*Black Box*
	Verifikasi *password*	*Black Box*
Pengaturan Akun	Menampilkan data *member*	*Black Box*
	Menampilkan *form* ubah akun *member*	*Black Box*
	Verifikasi *password* lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verifikasi konfirmasi *password*	*Black Box*
	Ubah Akun *member*	*Black Box*
	Simpan perubahan data *member*	*Black Box*

Kelas Uji	Butir Uji	Jenis Pengujian
Menu History Pembelian	Menampilkan Daftar history	*Black Box*
	Menampilkan detail Pembelian	*Black Box*
	Manampilkan pilihan pembayaran	*Black Box*
	Memproses pembayaran Paypal, Konfirmasi transfer dan PayPal.	*Black Box*
	Menampilkan Status transaksi dan informasinya.	*Black Box*
Menu Cara Belanja	Menampilkan Informasi Cara berbelanja di Website *Blacbox-shop*.	*Black Box*
Cara Pembayaran	Menampilkan informasi cara pembayaran.	*Black Box*
ToS (*Term of Service)*	Menampilkan syarat dan ketentuan transaksi.	*Black Box*
Menu Barang	Menampikan data Barang	*Black Box*
	Menampilkan barang berdasarkan kategori.	*Black Box*
	Menampilkan barang berdasarkan bahan.	*Black Box*
Menu keranjang belanja	Menampilkan data pemesanan	*Black Box*
	Mengupdate data pemesanan	*Black Box*
	Menghapus data pemesanan	*Black Box*
Form Pengiriman	Menampilkan form pengiriman	*Black Box*
	Menyimpan data pengiriman	*Black Box*
Form Pembayaran	Menampilkan konfirmasi Pemesanan	*Black Box*
	Pilih metode pembayaran	*Black Box*

Kelas Uji	Butir Uji	Jenis Pengujian
	Menunda pembayaran	*Black Box*
	Membatalkan Pesanan	*Black Box*
Pencarian Barang	Menampilkan Hasil pencarian barang	*Black Box*
Keluar	Menghapus *sessionmember*	*Black Box*

Tabel 4.4 Skenario Aplikasi Pengujian Untuk *Admin*

Kelas Uji	Butir Uji	Jenis Pengujian
Masuk	Isi data masuk	*Black Box*
	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Lupa *password*	Menampilkan *form* Lupa *password*	*Black Box*
Data Transaksi	Tampilkan data transaksi	*Black Box*
	Konfirmasi Pembayaran	*Black Box*
	Konfirmasi pengiriman	*Black Box*
Admin	Tampilkan data *admin*	*Black Box*
	Edit data *admin*	*Black Box*
	Tambah data *admin*	*Black Box*
	Hapus data *admin*	*Black Box*
Provinsi	Tampilkan data Provinsi	*Black Box*
	Tambah data Provinsi	*Black Box*
	Ubah data Provinsi	*Black Box*

Kelas Uji	Butir Uji	Jenis Pengujian
	Hapus data Provinsi	*Black Box*
Kota	Tampilkan data Kota	*Black Box*
	Edit data Kota	*Black Box*
	Tambah data Kota	*Black Box*
	Hapus data Kota	*Black Box*
Bahan	Tampilkan data Bahan	*Black Box*
	Ubah data bahan	*Black Box*
	Tambah data bahan	*Black Box*
	Hapus data bahan	*Black Box*
Kategori	Tampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
	Hapus data kategori	*Black Box*
Barang	Tampilkan data barang	*Black Box*
	Tambah data barang	*Black Box*
	Ubah data barang	*Black Box*
	Hapus data barang	*Black Box*
	Tambah Detail Barang	*Black Box*
Detail Barang	Tampilkan detail barang	*Black Box*

Kelas Uji	Butir Uji	Jenis Pengujian
	Tambah detail barang	*Black Box*
	Ubah detail barang	*Black Box*
	Hapus Detail barang	*Black Box*
Member	Tampilan Data *Member*	*Black Box*
	Hapus Data *Member*	*Black Box*
Laporan	Tampilan laporan data barang Harian, Bulanan dan tahunan.	*Black Box*
	Tampilkan laporan data barang	*Black Box*
Keluar	Menghapus *session admin*	*Black Box*

4.4.3.2 Kasus dan Hasil Pengujian

1. Pengujian Daftar *Member*

Tabel 4.5 Pengujian Daftar *member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data pendaftaran, contoh masukan : Nama : Nenden Password : ******** Konfirmasi Password: ******** Alamat Email : ns.jayantina@gmail.com Alamat : Jl.aspal Provinsi : Jawa Barat Kota : Bandung	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftaran.	Form registrasi terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftran.	[√] Diterima [ ] Ditolak

Kodepos : 40134 No Telpon : 085320056131 Masukan Kode : TMLSY			
Kasus dan Hasil Uji (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data kosong	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	[√] Diterima [ ] Ditolak
Nama tidak diisi	Tampilkan pesan kesalahan”Nama lengkap harus diisi”	Dapat menampilkan Pesan kesalahan “Nama lengkap harus diisi”	[√] Diterima [ ] Ditolak
password tidak diisi	Tampilkan pesan kesalahan ”password harus diisi”	Dapat menampilkan pesan kesalahan ”password harus diisi”	[√] Diterima [ ] Ditolak
Konfirmasi *password* tidak diisi	Tampilkan pesan kesalahan ”confirm harus diisi”	Dapat menampilkan pesan kesalahan ”confirm harus diisi”	[√] Diterima [ ] Ditolak
Email tidak diisi/tidak valid	Tampilkan pesan kesalahan “Email harus diisi”	Dapat menampilkan pesan kesalahan “Email harus diisi”	[√] Diterima [ ] Ditolak
Alamat tidak diisi	Tampilkan pesan kesalahan”Alamat harus diisi”	Dapat menampilkan pesan kesalahan ”Alamat harus diisi”	[√] Diterima [ ] Ditolak
Provinsi tidak diisi	Tampilkan pesan kesalahan ”Provinsi harus diisi”	Dapat menampilkan pesan kesalahan ”Provinsi harus diisi”	[√] Diterima [ ] Ditolak
kota tidak diisi	Tampilkan pesan kesalahan ”kota harus diisi”	Dapat menampilkan Pesan kesalahan ”kota harus diisi”	[√] Diterima [ ] Ditolak
Kode pos tidak diisi	Tampilkan pesan kesalahan ” kode pos harus diisi”	Dapat menampilkan pesan kesalahan ”kode pos harus	[√] Diterima [ ] Ditolak

		diisi”	
No Telepon tidak diisi	Tampilkan pesan “data harus diisi”	Dapat menampilkan pesan kesalahan “no telepon haris diisi”	[√] Diterima [ ] Ditolak
Data *captcha* salah	Tampilkan pesan “*captcha* tidak sesuai”	Dapat menampilkan pesan kesalahan “*captcha* tidak sesuai”	[√] Diterima [ ] Ditolak

2. Pengujian Masuk *Member*

Tabel 4.6 Pengujian masuk *member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat *email* : haipos19@gmail.com	Mengisikan data alamat *email* pada *field* alamat *email*.	Dapat mengisi *field* alamat *email* dan dapat mengisi data login dengan benar	[√] Diterima [ ] Ditolak
Memasukan *Password* : ******	Mengisikan data *password* pada *field password*.	Mengisikan data *password* pada *field password*.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data *email* atau *password* yang tidak terdaftar	Tampilkan pesan kesalahan “Maaf akun tidak di temukan”	Muncul pesan “Maaf akun tidak di temukan”	[√] Diterima [ ] Ditolak
mengosongkan data *email* atau *password*	Tampilkan pesan kesalahan “Maaf akun tidak ditemukan”	Dapat menampilkan pesan kesalahan ”Maaf akun tidak ditemukan ”	[√] Diterima [ ] Ditolak
alamat Email tidak diisi	Tampilkan pesan kesalahan “Maaf akun tidak ditemukan”	Tampilkan pesan kesalahan “Maaf akun tidak ditemukan”	[√] Diterima [ ] Ditolak
Password tidak diisi	Tampilkan pesan kesalahan “Maaf akun tidak ditemukan”	Tampilkan pesan kesalahan “Maaf akun tidak ditemukan”	[√] Diterima [ ] Ditolak

3. Pengujian Pengaturan Akun

Tabel 4.7 Pengujian Pengaturan Akun

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
data yang dimasukkan semuanya benar	*Password* baru berhasil disimpan di *database*	Dapat menyimpan data *password* yang baru di *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu field belum terisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* tersebut harus diisi”	[√] Diterima [ ] Ditolak
Data *password* lama salah	muncul pesan "password salah"	Dapat menampilkan pesan kesalahan ”*Password* salah”	[√] Diterima [ ] Ditolak
Password lama tidak diisi	muncul pesan kesalahan "password lama harus diisi”	Dapat menampilkan pesan kesalahan "password lama harus diisi”	[√] Diterima [ ] Ditolak
Confirm *password* tidak diisi	muncul pesan kesalahan "confirm password harus diisi”	Dapat menampilkan pesan kesalahan "confirm password harus diisi”	
Code tidak diisi	Muncul pesan kesalahan”kode spam harus diisi”	Dapat menampilkan pesan kesalahan Kode spam harus diisi	[√] Diterima [ ] Ditolak
Data kosong	Muncul pesan kesalahan field harus diisi	Dapat menampilkan pesan kesalahan field harus diisi	[√] Diterima [ ] Ditolak

4. Pengujian *Menu Keranjang Belanja*

Tabel 4.8Pengujian Menu Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih warna yang dinginkan	Dapat memilih warna	Dapat memilih warna	[√] Diterima [ ] Ditolak
Pilih bahan yang dinginkan	Dapat memilih bahan	Dapat memilih bahan	[√] Diterima [ ] Ditolak
Isi jumlah barang yang akan dibeli	Dapat mengisi jumlah barang yang akan dibeli	Dapat mengisi jumlah barang yang akan dibeli	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Data Masukan	Data Masukan	Data Masukan
Tidak memiilih warna yang dinginkan	Anda belum mengisi warna, bahan, maupun jumlah, silahkan ulangi kembali	Anda belum mengisi warna, bahan, maupun jumlah, silahkan ulangi kembali	[√] Diterima [ ] Ditolak
Tidak mengisi jumlah barang yang akan dibeli	Anda belum mengisi warna, bahan, maupun jumlah, silahkan ulangi kembali	Anda belum mengisi warna, bahan, maupun jumlah, silahkan ulangi kembali	[√] Diterima [ ] Ditolak

5. Pengujian *Form CheckOut*

Tabel 4.9 Pengujian Form *CheckOut*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data pembelian yang ada dalam keranjang belanja	Data pembelian yang disimpan di dalam *databese*	Dapat menyimpan data pembelian yang sisimpan di dalam *database*.	[√] Diterima [ ] Ditolak

6. Pengujian *Form Pengiriman*

Tabel 4.10 Pengujian Form Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih jasa	Menampilkan form	Dapat	[√] Diterima

pengiriman	jasa pengiriman	menampilkan jasa pengiriman	[ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Jenis pengiriman tidak diisi	Menampilkan pesan kesalahan "Anda belum memilih Jenis pengiriman!"	Dapat menampilkan pesan kesalahan "Anda belum memilih Jenis pengiriman!"	[√] Diterima [ ] Ditolak

7. Pengujian *Form Pembayaran*

Tabel 4.11 Pengujian Form Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh No. invoice : 5 Nama Bank : Mandiri Atas Nama : Nenden Jumlah Transfer : 2.000.000	Menampilkan *form* konfirmasi	Dapat memunculkan pesan form konfirmasi.	[√] Diterima [ ] Ditolak
Memasukan data kosong	Tampilkan pesan kesalahan "*field* ini harus diisi"	Dapat menampilkan pesan kesalahan "*field* ini harus diisi"	[√] Diterima [ ] Ditolak

8. Pengujian Masuk *Admin*

Tabel 4.12 Pengujian Masuk *Admin*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat *email* : bbshop_clerk@yahoo.co.id Memasukan *Password* : ******	Mengisikan data alamat *email* pada *field* alamat *email*.	Data *email* dan data *password* benar maka akan masuk ke menu utama	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data *email* atau *password* yang tidak terdaftar	Tampilkan pesan kesalahan "Maaf akun tidak di temukan"	Muncul pesan "Maaf akun tidak di temukan"	[√] Diterima [ ] Ditolak
mengosongkan data *email* atau *password*	Tampilkan pesan kesalahan "Maaf akun tidak ditemukan"	Dapat menampilkan pesan kesalahan "Maaf akun tidak ditemukan "	[√] Diterima [ ] Ditolak
alamat Email tidak diisi	Tampilkan pesan kesalahan "Maaf akun tidak ditemukan"	Tampilkan pesan kesalahan "Maaf akun tidak ditemukan"	[√] Diterima [ ] Ditolak
Password tidak diisi	Tampilkan pesan kesalahan "Maaf akun tidak ditemukan"	Tampilkan pesan kesalahan "Maaf akun tidak ditemukan"	[√] Diterima [ ] Ditolak

9. Pengujian *Pengolahan Data Admin*

Tabel 4.13 Pengujian Pengolahan Data *Admin*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data yang benar	Password baru berhasil disimpan di database	Dapat menyimpan data password yang baru dengan muncul pesan " Data berhasil diubah"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengosongkan salah satu field	Muncul pesan kesalahan field tersebut harus diisi	Dapat menampilkan pesan kesalahan "field ini harus diisi" bila ada field yang belum diisi	[√] Diterima [ ] Ditolak

Memasukan password yang salah	muncul pesan "password salah"	Dapat menampilkan pesan kesalahan ”Password salah”	[√] Diterima [ ] Ditolak
Memasukan data password baru dan konfirmasi password baru yang tidak sama	Muncul pesan "password baru dan konfirmasi password harus sama"	Dapat menampilkan pesan "password baru dan konfirmasi password harus sama"	[√] Diterima [ ] Ditolak
Mengosongkan semua field	Muncul pesan kesalahan field harus diisi	Dapat menampilkan pesan kesalahan field harus diisi	[√] Diterima [ ] Ditolak

10. Pengujian *Pengolahan Data Bahan*

Tabel 4.14 Pengujian pengolahan data bahan

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data bahan Contoh: Nama bahan : Plastik	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian. Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi nama bahan	Tampilkan pesan kesalahan nama bahan harus diisi	Dapat menampilkan pesan kesalahan nama bahan harus diisi	[√] Diterima [ ] Ditolak

11. Pengujian *Pengolahan Data Kategori*

Tabel 4.15 Pengujian Pengolahan data Kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data kategori	Jika semua field terisi atau benar	Jika semua field terisi atau benar	[√] Diterima [ ] Ditolak

Contoh: Nama kategori: *Cube Puzzle*	dalam pengisian, maka akan dapat melanjutkan proses tambah kategori	dalam pengisian, maka akan dapat melanjutkan proses tambah kategori. Data kategori akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	

Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data nama kategori masih kosong	Tampilkan pesan kesalahan “nama kategori harus isi”	Dapat menampilkan pesan kesalahan “nama kategori harus diisi”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data barang kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

12. Pengujian *Pengolahan Data Barang*

Tabel 4.16 Pengujian Pengolahan Data Barang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data barang Contoh: kategori: Cube Puzzle Nama : DIY type-F Harga : 10000 Warna : putih Diskon : deskrispsi Gambar :	Jika semua field terisi semua maka akan muncul pesan “ data berhasil ditambah”	Jika semua field terisi semua maka akan muncul pesan “ data berhasil ditambah”. Data barang akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukkan	Yang diharapkan	Pengamatan	Kesimpulan
Data barang masih kosong	Tampilkan pesan kesalahan “*field*	Dapat menampilkan pesan kesalahan”	[√] Diterima [ ] Ditolak

	tersebut harus diisi"	*field* tersebut harus diisi"	
Nama barang tidak diisi	Muncul pesan kesalahan"nama baramg harus diisi"	Dapat menampilkan pesan kesalahan "nama barang harus diisi"	[√] Diterima [ ] Ditolak
Harga tidak diisi	Muncul pesan kesalahan"harga harus diisi"	Dapat menampilkan pesan kesalahan "harga harus diisi"	[√] Diterima [ ] Ditolak
Warna tidak diisi	Muncul pesan kesalahan"warna harus diisi"	Dapat menampilkan pesan kesalahan "warna harus diisi"	[√] Diterima [ ] Ditolak

13. Pengujian *Pengolahan Data Provinsi*

Tabel 4.17 Pengujian Pengolahan Data Provinsi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data provinsi Contoh: Nama Provinsi : Jawa Barat	Jika semua field terisi maka akan muncul pesan "data berhasil ditambah"	Jika semua field terisi maka akan muncul pesan "data berhasil ditambah".	[√insi] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama provinsi masih kosong	Tampilkan pesan kesalahan "nama provinsi harus diisi"	Dapat menampilkan pesan kesalahan " nama provinsi harus diisi"	[√] Diterima [ ] Ditolak

14. Pengujian *Pengolahan Data Kota*

Tabel 4.18 Penujian Pengolahan Data Kota

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota :	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi	[√] Diterima [ ] Ditolak

Cikarang	pesan kesalahan bahwa field tersebut harus terisi.	atau salah dalam pengisian. Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
nama kota masih kosong	Tampilkan pesan kesalahan nama kota harus disi	Dapat menampilkan pesan kesalahan nama kota harus disi	[√] Diterima [ ] Ditolak

15. Pengujian Pengolahan Ongkos Kirim

Tabel 4.19 Penujian *Pengolahan Ongkos Kirim*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data ongkos kirim Contoh: Provinsi : Jawa Barat Kota : Bandung Jenis pengiriman : Regular service harga: Rp.8000	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah onkos kirim	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah onkos kirim. Data ongkos kirim akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data ongkos kirim masih kosong	Tampilkan pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota : Cikarang	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam	[√] Diterima [ ] Ditolak

	bahwa field tersebut harus terisi.	pengisian. Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
nama kota masih kosong	Tampilkan pesan kesalahan nama kota harus disi	Dapat menampilkan pesan kesalahan nama kota harus disi	[√] Diterima [ ] Ditolak

4.4.3.3 Keimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian dengan kasus *sample* uji yang telah dilakukan *member*ikan kesimpulan bahwa pada proses masih memungkinkan untuk terjadinya kesalahan pada sintaks tetapi secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan.

4.4.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, yaitu instansi yang bersangkutan dan masyarakat calon pengguna aplikasi dengan membuat kuesioner mengenai kepuasan *user* dengan kandungan *point* syarat *user friendly* untuk selanjutnya dibagikan kepada sebagian *user* dengan mengambil sample sebanyak 7 orang dimana 5 untuk web *user* dan 2 untuk web *admin*. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru.

Pengujian beta ini dilakukan dengan cara melakukan pengujian langsung di tempat penelitian. Penelitian ini dilakukan di dua lingkungan :

1. Lingkungan manajemen *Blacbox-shop*

   Penelitian di lingkungan menejemen *Blacbox-shop* dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

2. Lingkungan Masyarakat Pengguna Aplikasi

   Penelitian di lingkungan pada masyarakat umum khususnya yang sudah terbiasa dengan teknologi *internet*, dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat *member*ikan kemudahan masyarakat umum untuk melakukan transaksi pembelian barang secara *online*.

Berdasarkan data hasil kuesioner, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus: $Y = A/B * 100\%$

Keterangan :

A = Banyaknya jawaban responden tiap soal

B = Jumlah responden

Y = Nilai Persentase

4.4.4.1 Kuesioner Pengguna

Kuesioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 5 pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Pertanyaannya adalah sebagai berikut :

1. Kuesioner Pengujian BetaUntuk Pengunjung dan *Member*

PERTANYAAN DAN HASIL PENGUJIAN

Pertanyaan No. 1	" Aplikasi *e-Commerce* yang dibangun ini sudah *user friendly??*"				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	1	4	0
Persentase nilai					
JumlahFrekuensi	0	0	1	4	0
JumlahPersentase:	0 %	0%	20%	80%	0%

Pertanyaan No.2	"Apliaksi *e-Comm*erce yang dibangun mempermudah Anda dalam melakukan transaksi pembelian dibandingkan transaksi secara manual?"

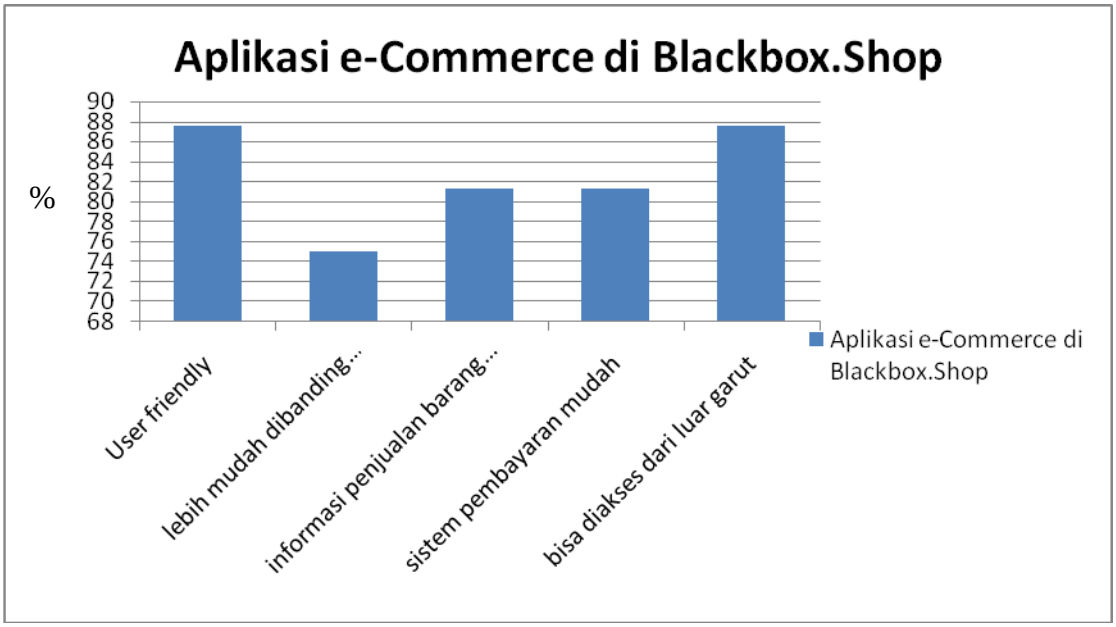
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	1	1	3	0
Persentase nilai					
JumlahFrekuensi	0	1	1	3	0
JumlahPersentase:	0 %	20%	20%	60%	0%

Pertanyaan No.3	"Aplikasi*e-Commerce* yang dibangun dapat membantu Anda untuk mendapatkan informasi terbaru tentang barang - barang *Blacbox-shop*?"				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	2	3	0
Persentase nilai					
JumlahFrekuensi	0	0	2	3	0
JumlahPersentase:	0 %	0%	40%	60%	0%

Pertanyaan No.4	"Sistem pembayaran yang digunakan akan lebih memudahkan anda?"				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	2	3	0
Persentase nilai					

JumlahFrekuensi	0	0	2	3	0
JumlahPersentase:	0 %	0%	40%	60%	0%

Pertanyaan No 5	" Aplikasi *e-Commerce* ini dapat memperluas pangsa pesar, khususnya untuk konsumen diluar Garut?"				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	1	4	0
Persentase nilai					
JumlahFrekuensi	0	0	1	4	0
JumlahPersentase:	0 %	0%	20%	80%	0%



Gambar 4. 1 Diagram kemudahan penggunaan aplikasi menurut pengunjung dan *member*

2. Kuesioner Pengujian Beta Untuk *Admin*

PERTANYAAN DAN HASIL PENGUJIAN

Pertanyaan No. 1	" Aplikasi *e-Commerce* yang dibangun ini *user friendly* untuk *admin*?"				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	1	1	0
Persentase nilai					
JumlahFrekuensi	0	0	1	1	0
JumlahPersentase:	0 %	0%	50%	50%	0%

Pertanyaan No.2	" Aplikasi yang dirancang ini dapat mempermudah dalam proses promosi dan penjualan barang?"				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	0	2	0
Persentase nilai					
JumlahFrekuensi	0	0	0	2	0
JumlahPersentase:	0 %	0%	0%	100%	0%

Pertanyaan No.3	" Aplikasi yang di bangun apakah sudah *member*ikan pengembangan

	terhadap perusahaan?”				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	1	1	0
Persentase nilai					
JumlahFrekuensi	0	0	1	1	0
JumlahPersentase:	0 %	0%	50%	50%	0%

Pertanyaan No 4	“ Aplikasi ini dirancang sudah sesuai kebutuhan?”				
Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	1	1	0
Persentase nilai					
JumlahFrekuensi	0	0	1	1	0
JumlahPersentase:	0 %	0%	50%	50%	0%

PertanyaanNo.5	“Apakah aplikasi ini sudah menjadi sebuah sarana penghubung yang efektif?”

Kategori jawaban	Sangat tidak setuju	Tidak setuju	Biasa saja	Setuju	Sangat setuju
Frekuensi jawaban	0	0	1	1	0
Persentase nilai					
JumlahFrekuensi	0	0	1	1	0
JumlahPersentase:	0 %	0%	50%	50%	0%

4.4.4.2 Kesimpulan Pengujian Beta

Dari pengujian beta yang telah dilakukan, yaitu dengan pengujian perhitungan pilihan kategori jawaban dari kuesioner yang telah dibagikan di lapangan didapat kesimpulan bahwa aplikasi yang dirancang dinilai cukup dapat membantu dalam pengolahan data, sesuai kebutuhan, menjadi suatu sarana yang bermanfaat, keefisienanwaktu didapat, info yang disampaikan cepat, mudah dan akurat.