BAB 4

IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian tarhadap sistem yang baru. Tahap ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman yang akan digunakan. Setelah implementasi maka dilakukan pengujian sistem yang baru dimana akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk pengembangan sistem selanjutnya.

4.1 Implemantasi Sistem

Tahap implementasi merupakan tahap penciptaan perangkat lunak, tahap kelanjutan dari kegiatan perancangan sistem. Tahap ini merupakan tahap dimana sistem siap untuk dioperasikan, yang terdiri dari penjelesan mengenai lingkungan implementasi, dan implementasi program.

4.1.1 Lingkungan Sistem

Untuk mendukung aplikasi yang akan diterapkan pada lingkungan implementasi, maka dalam hal ini menggunakan perangkat keras dan perangkat lunak yang menunjang dalam pembangunan aplikasi penjualan online pada Toko Sawargi.

4.1.1.1 Perangkat Keras Yang Dibutuhkan

Spesifikasi perangkat keras yang dibutuhkan dalam pembangunan perangkat keras adalah tercantum pada tabel 4.1 sebagai berikut :

Table 4-1 Perangkat Keras Yang Dibutuhkan

Perangkat Keras	Spesifikasi (Min)
Processor	Kecepatan 1GHz
RAM	128Mb
Harddisk	20Gb
Monitor	Resolution 1024 x 768
VGA	32Mb
Lan Card	10/100 Mbps
Koneksi Internet	64 Kbps

4.1.1.2 Perangkat Lunak Yang Dibutuhkan

Spesifikasi perangkat lunak yang dibutuhkan dalam membangun aplikasi penjualan online pada Toko Sawargi adalah sebagai berikut :

Table 4-2 Perangkat Lunak Yang Dibutuhkan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP, Windows 7
2	Bahasa Pemograman	PHP
3	Web Server	XAMPP
4	Database Server	MySQL
5	Web Browser	Mozilla firefox 3.0
6	Code Editor	Adobe Dreamweaver CS3
7	Image Editor	Adobe Photoshop CS
8	DFD Modeler	Visio 2007

4.2 Implementasi Database

Pembuatan *database* dilakukan dengan menggunakan aplikasi *DBMS MySQL*. Implementasi *database* dalam bahasa *SQL* adalah sebagai berikut :

Table admin	Table detail_pesanan
CREATE TABLE IF NOT EXISTS `sawargi`.`admin` (□`username` VARCHAR(50) NOT NULL ,□ `password` VARCHAR(50) NOT NULL ,□ `nama_lengkap` VARCHAR(50) NOT NULL ,□ `email` VARCHAR(50) NOT NULL,□ `no_telp` VARCHAR(20) NOT NULL,□ PRIMARY KEY	CREATE TABLE IF NOT EXISTS `sawargi`.`detail_pesanan` (□ `id_detail_pesanan` INT(5) NOT NULL AUTO_INCREMENT ,□ `id_pesanan` INT(7) NOT NULL ,□ `qty` INT(11) NOT NULL ,□ `jumlah` INT(5) NOT NULL ,□ `harga` DOUBLE NOT NULL ,□

(`username`))□ENGINE = InnoDB□DEFAULT CHARACTER SET = latin1;	`total` DOUBLE NOT NULL ,□ `diskon` DOUBLE NOT NULL DEFAULT '0' ,□`id_member` INT(5) NOT NULL ,□`berat_total` DECIMAL(5,2) NOT NULL ,□`id_produk` INT(5) NOT NULL ,□ PRIMARY KEY (`id_detail_pesanan`) ,□ INDEX `id_pesanan` (`id_pesanan` ASC) ,□ INDEX `id_member` (`id_member` ASC) ,□ INDEX `fk_detail_pesanan_produk1` (`id_produk` ASC) ,□ CONSTRAINT `fk_detail_pesanan_produk1`□ FOREIGN KEY (`id_produk`)□ REFERENCES `sawargi`.`produk` (`id_produk`)□ ON DELETE NO ACTION□ ON UPDATE NO ACTION)□ENGINE = InnoDB□DEFAULT CHARACTER SET = latin1;
Table harga_kirim	**Table jasa**
CREATE TABLE IF NOT EXISTS `sawargi`.`harga_kirim` (`id_harga` INT(10) NOT NULL AUTO_INCREMENT , `harga` INT(10) NOT NULL , `id_paket` INT(5) NOT NULL , `id_kota` INT(5) NOT NULL , PRIMARY KEY (`id_harga`) , INDEX `fk_harga_kirim_paket1` (`id_paket` ASC) , INDEX `fk_harga_kirim_kota1` (`id_kota` ASC) , CONSTRAINT `fk_harga_kirim_paket1` FOREIGN KEY (`id_paket`) REFERENCES `sawargi`.`paket` (`id_paket`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `fk_harga_kirim_kota1` FOREIGN KEY (`id_kota`) REFERENCES `sawargi`.`kota` (`id_kota`) ON DELETE NO ACTION	CREATE TABLE IF NOT EXISTS `sawargi`.`jasa` (`id_jasa` INT(2) NOT NULL AUTO_INCREMENT , `nama_jasa` VARCHAR(50) NOT NULL , PRIMARY KEY (`id_jasa`)) ENGINE = InnoDB AUTO_INCREMENT = 3 DEFAULT CHARACTER SET = latin1;

```
   ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 54□DEFAULT
CHARACTER SET = latin1;
```

Table katagori

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`kategori` (
  `id_kategori` INT(5) NOT
NULL ,
`nama_kategori` VARCHAR(100)
NOT NULL ,
`kategori_seo` VARCHAR(100)
NOT NULL ,
  PRIMARY KEY
(`id_kategori`) )
ENGINE = InnoDB

DEFAULT CHARACTER SET =
latin1;
```

Table keranjang

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`keranjang` (
`id_keranjang` INT(5) NOT
NULL AUTO_INCREMENT ,
`qty` DECIMAL(10,0) NOT NULL
,
`berat_total` DECIMAL(10,0)
NOT NULL ,
  `jumlah` DOUBLE NOT NULL ,
  `diskon` DOUBLE NOT NULL
DEFAULT '0' ,
  `total` DOUBLE NOT NULL ,
`id_member` INT(5) NOT NULL ,
`id_produk` INT(5) NOT NULL ,
  PRIMARY KEY
(`id_keranjang`) ,
INDEX `fk_keranjang_member1`
(`id_member` ASC) ,
INDEX `fk_keranjang_produk1`
(`id_produk` ASC) ,
CONSTRAINT
`fk_keranjang_member1`
 FOREIGN KEY (`id_member` )
REFERENCES `sawargi`.`member`
(`id_member` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT
`fk_keranjang_produk1`
  FOREIGN KEY (`id_produk` )
  REFERENCES
`sawargi`.`produk`
(`id_produk` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 13

DEFAULT CHARACTER SET =
latin1;
```

Table komentar

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`komentar` (
  `id_komen` INT(5) NOT NULL
AUTO_INCREMENT ,
  `tanggal` DATE NOT NULL ,
```

Table kota

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`kota` (
`id_kota` INT(5) NOT NULL
AUTO_INCREMENT ,
```

```
  `isi` TEXT NOT NULL ,
  `status_komentar`
ENUM('Diterima','Ditolak')
NOT NULL ,
  `jam` TIME NOT NULL ,
  `id_produk` INT(5) NOT
NULL ,
  `id_member` INT(5) NOT
NULL ,
  PRIMARY KEY (`id_komen`) ,
INDEX `fk_komentar_produk1`
(`id_produk` ASC) ,
INDEX `fk_komentar_member1`
(`id_member` ASC) ,
CONSTRAINT
`fk_komentar_produk1`
FOREIGN KEY (`id_produk` )
 REFERENCES
`sawargi`.`produk`
(`id_produk` )
   ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT
`fk_komentar_member1`
 FOREIGN KEY (`id_member` )
  REFERENCES
`sawargi`.`member`
(`id_member` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

DEFAULT CHARACTER SET =
latin1;
```

```
  `nama_kota` VARCHAR(20) NOT
NULL ,
 `id_provinsi` INT(5) NOT
NULL ,
PRIMARY KEY (`id_kota`) ,
INDEX `fk_kota_provinsi1`
(`id_provinsi` ASC) ,
CONSTRAINT
`fk_kota_provinsi1`
FOREIGN KEY (`id_provinsi` )
REFERENCES
`sawargi`.`provinsi`
(`id_provinsi` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 305

DEFAULT   CHARACTER   SET   =
latin1;
```

Table member

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`member` (
  `id_member` INT(5) NOT
NULL AUTO_INCREMENT ,
  `email` VARCHAR(50) NOT
NULL ,
  `password` VARCHAR(50) NOT
NULL ,
  `nama_member` VARCHAR(30)
NOT NULL ,
  `alamat` VARCHAR(100) NOT
NULL ,
  `kota` VARCHAR(10) NOT
NULL ,
 `kodepos` INT(6) NOT NULL ,
  `aktif` ENUM('Y','T') NOT
```

Table paket

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`paket` (
 `id_paket` INT(5) NOT NULL
AUTO_INCREMENT ,
  `nama_paket` VARCHAR(25)
NOT NULL ,
  `id_jasa` INT(2) NOT NULL ,
PRIMARY KEY (`id_paket`) ,
INDEX `fk_paket_jasa1`
(`id_jasa` ASC) ,
  CONSTRAINT `fk_paket_jasa1`
    FOREIGN KEY (`id_jasa` )
    REFERENCES
`sawargi`.`jasa` (`id_jasa` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
```

```
NULL ,
  `no_telp` VARCHAR(20) NOT
NULL ,
  `id_provinsi` INT(5) NOT
NULL COMMENT '
      ' ,
 `id_kota` INT(5) NOT NULL ,
 PRIMARY KEY (`id_member`) ,
INDEX `id_provinsi`
(`id_provinsi` ASC) ,
 INDEX `fk_member_kota1`
(`id_kota` ASC) ,
CONSTRAINT `fk_member_kota1`
    FOREIGN KEY (`id_kota` )
 REFERENCES `sawargi`.`kota`
(`id_kota` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB
AUTO_INCREMENT = 2

DEFAULT CHARACTER SET =
latin1;
```

```
ENGINE = InnoDB
AUTO_INCREMENT = 4

DEFAULT   CHARACTER   SET   =
latin1;
```

Table pembayaran

```
  `id_pembayaran` INT(5) NOT
NULL AUTO_INCREMENT ,
  `tanggal` DATE NOT NULL ,
  `metode` VARCHAR(20) NOT
NULL ,
  `jumlah_pembayaran` DOUBLE
NOT NULL ,
  `bank` VARCHAR(20) NOT
NULL ,
  `catatan` TEXT NOT NULL ,
  `id_pesanan` INT(7) NOT
NULL ,
  `id_member` INT(5) NOT
NULL ,
PRIMARY KEY
(`id_pembayaran`) ,
 INDEX
`fk_pembayaran_pesanan1`
(`id_pesanan` ASC) ,
  INDEX
`fk_pembayaran_member1`
(`id_member` ASC) ,
CONSTRAINT
`fk_pembayaran_pesanan1`
FOREIGN KEY (`id_pesanan` )
REFERENCES
`sawargi`.`pesanan`
```

Table pengiriman

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`pengiriman` (
`id_pengiriman` INT(5) NOT
NULL ,
`nama_penerima` VARCHAR(30)
NOT NULL ,
 `alamat_penerima` TEXT NOT
NULL ,
 `no_resi` INT(5) NOT NULL ,
  `id_pesanan` INT(7) NOT
NULL ,
  `id_kota` INT(5) NOT NULL ,
 `id_paket` INT(5) NOT NULL ,
 PRIMARY KEY
(`id_pengiriman`) ,
  INDEX
`fk_pengiriman_pesanan1`
(`id_pesanan` ASC) ,
  INDEX `fk_pengiriman_kota1`
(`id_kota` ASC) ,
  INDEX
`fk_pengiriman_paket1`
(`id_paket` ASC) ,
 CONSTRAINT
`fk_pengiriman_pesanan1`
 FOREIGN KEY (`id_pesanan` )
 REFERENCES
```

```
(`id_pesanan` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
 CONSTRAINT
`fk_pembayaran_member1`
 FOREIGN KEY (`id_member` )
REFERENCES
`sawargi`.`member`
(`id_member` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

DEFAULT CHARACTER SET =
latin1;
```

```
`sawargi`.`pesanan`
(`id_pesanan` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
 CONSTRAINT
`fk_pengiriman_kota1`
  FOREIGN KEY (`id_kota` )
  REFERENCES `sawargi`.`kota`
(`id_kota` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
 CONSTRAINT
`fk_pengiriman_paket1`
  FOREIGN KEY (`id_paket` )
 REFERENCES `sawargi`.`paket`
(`id_paket` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT   CHARACTER   SET   =
latin1;
```

Table pesanan

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`pesanan` (
  `id_pesanan` INT(7) NOT
NULL ,
  `tanggal_pesanan` DATE NOT
NULL ,
  `jam_pesanan` TIME NOT
NULL ,
  `total_berat`
DECIMAL(10,0) NOT NULL ,
  `jml_total_ongkir` DOUBLE
NOT NULL ,
  `total_bayar` DOUBLE NOT
NULL ,
`status_pesanan`
ENUM('Dipesan','Dibayar','Di
kirim','Diterima') NOT NULL
,
 `status_lunas`
ENUM('Ya','Tidak') NOT NULL
,
  `id_member` INT(5) NOT
NULL ,
  `id_produk` INT(5) NOT
NULL ,
`id_jasa` INT(2) NOT NULL ,
  PRIMARY KEY (`id_pesanan`,
```

Table produk

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`produk` (
`id_produk` INT(5) NOT NULL ,
  `nama_produk` VARCHAR(50)
NOT NULL ,
  `status` VARCHAR(20) NOT
NULL ,
  `produk_seo` VARCHAR(100)
NOT NULL ,
  `deskripsi` TEXT NOT NULL ,
  `stok` INT(5) NOT NULL ,
  `berat` DECIMAL(5,2) NOT
NULL ,
  `gambar` VARCHAR(200) NOT
NULL ,
  `dibeli` INT(5) NOT NULL
DEFAULT '1' ,
  `dilihat` INT(5) NOT NULL ,
  `harga` DOUBLE NOT NULL ,
  `diskon` DOUBLE NOT NULL
DEFAULT '0' ,
  `tgl_masuk` DATE NOT NULL ,
  `total_rating` INT(5) NOT
NULL ,
  `id_kategori` INT(5) NOT
NULL ,
  PRIMARY KEY (`id_produk`) ,
```

```
`id_member`, `id_produk`) ,
  INDEX `fk_pesanan_member1`
(`id_member` ASC) ,
  INDEX `fk_pesanan_produk1`
(`id_produk` ASC) ,
  INDEX `fk_pesanan_jasa1`
(`id_jasa` ASC) ,
 CONSTRAINT `pesanan_ibfk_1`
 FOREIGN KEY (`id_pesanan` )
 REFERENCES
`sawargi`.`detail_pesanan`
(`id_pesanan` ),
 CONSTRAINT
`fk_pesanan_member1`
 FOREIGN KEY (`id_member` )
 REFERENCES
`sawargi`.`member`
(`id_member` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
  CONSTRAINT
`fk_pesanan_produk1`
 FOREIGN KEY (`id_produk` )
 REFERENCES
`sawargi`.`produk`
(`id_produk` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT
`fk_pesanan_jasa1`
    FOREIGN KEY (`id_jasa` )
    REFERENCES
`sawargi`.`jasa` (`id_jasa`
)
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET =
latin1;
```

```
  INDEX `fk_produk_kategori1`
(`id_kategori` ASC) ,
CONSTRAINT
`fk_produk_kategori1`
 FOREIGN KEY (`id_kategori` )
REFERENCES
`sawargi`.`kategori`
(`id_kategori` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

ENGINE = InnoDB

DEFAULT CHARACTER SET =
latin1;
```

Table provinsi

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`provinsi` (
  `id_provinsi` INT(5) NOT
NULL AUTO_INCREMENT ,
  `nama_provinsi`
VARCHAR(20) NOT NULL ,
  PRIMARY KEY
(`id_provinsi`) )
ENGINE = InnoDB
```

Table retur

```
CREATE  TABLE IF NOT EXISTS
`sawargi`.`retur` (
  `id_retur` INT(10) NOT NULL
AUTO_INCREMENT ,
  `keluhan` VARCHAR(300) NOT
NULL ,
  `id_detail_pesanan` INT(5)
NOT NULL ,
  `jumlah_retur` INT(11) NOT
```

```
AUTO_INCREMENT = 4

DEFAULT CHARACTER SET =
latin1;
```

```
NULL ,
`id_member` INT(5) NOT NULL ,
  `gambar_retur` VARCHAR(200)
NOT NULL COMMENT '    ' ,
  `status_retur`
ENUM('Dikirim', 'Diterima',
'Ditolak', 'Diretur') NOT
NULL ,
  `tgl_retur` DATE NOT NULL ,
`id_member` INT(5) NOT NULL ,
  PRIMARY KEY (`id_retur`) ,
 INDEX
`fk_retur_detail_pesanan1`
(`id_detail_pesanan` ASC) ,
  INDEX `fk_retur_member1`
(`id_member` ASC) ,
  CONSTRAINT
`fk_retur_detail_pesanan1`
FOREIGN KEY
(`id_detail_pesanan` )
 REFERENCES
`sawargi`.`detail_pesanan`
(`id_detail_pesanan` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT `fk_retur_member1`
FOREIGN KEY (`id_member` )
 REFERENCES
`sawargi`.`member`
(`id_member` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)
ENGINE = InnoDB

DEFAULT CHARACTER SET =
latin1;
```

Table rating

```
CREATE  TABLE IF NOT EXISTS
`mydb`.`rating` (
  `id_rating` INT NOT NULL ,
  `rating` INT(5) NOT NULL ,
  `id_produk` INT(5) NOT
NULL ,
  `id_member` INT(5) NOT
NULL ,
PRIMARY KEY (`id_rating`) ,
 INDEX `fk_rating_produk`
(`id_produk` ASC) ,
INDEX `fk_rating_member1`
(`id_member` ASC) ,
CONSTRAINT
```

```
`fk_rating_produk`
 FOREIGN KEY (`id_produk` )
 REFERENCES
`sawargi`.`produk`
(`id_produk` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION,
CONSTRAINT
`fk_rating_member1`
 FOREIGN KEY (`id_member` )
 REFERENCES
`sawargi`.`member`
(`id_member` )
    ON DELETE NO ACTION
    ON UPDATE NO ACTION)

ENGINE = InnoDB;
```

4.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengunjung dan pelanggan, admin. Adapun untuk tampilan implementasi pembangunan aplikasi penjualan online di Toko Sawargi akan dilampirkan pada halaman lampiran.

Table 4-3 Implentasi Antar Muka Pengunjung

Menu	Deskprisi	Nama File
Daftar	Untuk mendaftarkan diri sebagai pelanggan	registrasi.php
Beranda	Untuk menampilkan halaman utama beranda	index.php
Pencarian barang	Untuk melihat data barang yang tersedia yang sesuai yang akan dilihat	produk.php
Cara pembelian	Untuk melihat cara membeli produk	carabeli.html
Tentang kami	Untuk melihat tentang kami	Tentangkami.html

Table 4-4 Implementasi Antarmuka Member

Menu	Deskprisi	Nama File
Login	Untuk menangani proses *login* pelanggan	proses.php
Logout		index.php
Beranda	Untuk menampilkan halaman utama beranda	index.php
Lihat profile	Untuk melihat data profile pelanggan	profil.php
Ubah profile	Untuk mengubah data profile pelanggan	profile.php
Ubah password	Untuk mengubah password pelanggan	profile.php
Daftar belanja	Untuk mengolah data barang yang akan dipesan seperti hapus barang pesanan.	keranjang.php
Konfirmasi retur	Untuk melakukan konfirmasi retur jika telah melakukan belanja dan pembayaran.	retur.php
history	Untuk melihat daftar data barang yang telah dipesan.	pesanan.php
Komentar	Untuk mengolah data komentar seperti tambah komentar	komentar.php
Kategori	Untuk melihat data barang sesuai dengan kategori barang.	kategori.php

Table 4-5 Implementasi antarmuka admin

Menu	Deskprisi	Nama File
Halaman utama	Untuk menampilkan halaman utama *website* administrator	index.php
Transaksi	Untuk mengelola pesanan pelanggan, dimana didalamnya terdapat pesanan baru, pesanan dibayar, pesanan lunas, pesanan	pesanan.php, retur.php

	dikirim, pesanan diterima dan pesanan batal/kadaluarsa, mengelola pereturan yang dilakukan pelanggan	

4.4 Pengujian Perangkat Lunak

Dalam pengujian perangkat lunak ada elemen kritis dari jaminan kualitas perangkat lunak yang mempresentasikan kajian pokok dari spesifikasi, desain dan pengkodean. Pengujian yang digunakan untuk menguji sistem ini menggunakan metode pengujian *black-box*, dimana pengujian ini berfokus pada persyaratan fungsional perangkat lunak.

4.4.1 Pengujian Alpha

Pengujian alpha dilakukan dengan menggunakan metode *black-box* dan dibagi menjadi 2 bagian pengujian yaitu, pengujian aplikasi pelanggan, pengujian aplikasi admin.

4.4.2 Rencana Pengujian

Rencana pengujian akan dilakukan dengan menguji sistem secara alpha dan beta. Pengujian perangkat lunak aplikasi penjualan online pada setiap masing-masing pengguna dapat dilihat pada tabel berikut ini :

Table 4-6 Rencana pengujian aplikasi pelanggan

Item Uji	Detail Pengujian	Jenis Pengujian
Pendaftaran	Isi *form* pendaftaran	*Black-box*
Login	Isi *form login*	*Black-box*
Reset Password	Isi *form* alamat *email*	*Black-box*

Hubungi Kami	Isi *form* hubungi kami	*Black-box*
Data member	Ubah *form* data pelanggan	*Black-box*
Keranjang Belanja	Ubah data pembelian	*Black-box*
Alamat Pengiriman	Isi *form* alamat pengiriman	*Black-box*
Konfirmasi Pembayaran	Isi *form* pembayaran	*Black-box*

Table 4-7 Rencana pengujian aplikasi admin

Item Uji	Detail Pengujian	Jenis Pengujian
Login	Isi *form login*	*Black-box*
Produk	Tambah produk Ubah produk Hapus produk	*Black-box*
Kategori	Tambah kategori Ubah kategori Hapus kategori	*Black-box*
Jasa	Tambah jasa Ubah jasa Hapus jasa	*Black-box*
Paket	Tambah paket Hapus paket	*Black-box*
Harga Kirim	Tambah harga kirim Ubah harga kirim	*Black-box*
Provinsi	Tambah provinsi Ubah provinsi Hapus provinsi	*Black-box*
Kota	Tambah kota Ubah kota Hapus kota	*Black-box*
Data Pelanggan	Ubah status pelanggan	*Black-box*

4.4.3 Kasus dan Hasil Pengujian

4.4.3.1 Pengujian Pendaftaran Pelanggan

Tabel berikut akan menjelaskan mengenai pengujian pendaftaran pelanggan dengan kasus & hasil uji data benar dan data salah.

Table 4-8 Pengujian pendaftaran pelanggan

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data pendaftaran, contoh : Nama lengkap : edgar Alamat : jln gagak Provinsi : jawa barat Kota : Bandung Kode pos : 40123 No Telepon : 0852348789 Email : edgarprima@gmail.com Password : 1234567 Ulangi password : 1234567	Jika data yang dimasukkan telah lengkap dan tepat, sistem akan memproses data tersebut	Semua field terisi atau benar dalam pengisian, maka akan tampil pesan pendaftaran berhasil silahkan cek email untuk aktifasi akun	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Email tidak diisi	Dapat menampilkan pesan kesalahan "*kolom ini harus*	Menampilkan pesan kesalahan "*kolom ini harus diisi*"	[√] Diterima [ ] Ditolak

	diisi”.		
Email yang diisi tidak valid	Dapat menampilkan pesan kesalahan “*Email yang Anda masukkan tidak valid*”	Menampilkan pesan kesalahan “*Email yang Anda masukkan tidak valid*”	[√] Diterima [ ] Ditolak
Email yang diisi telah digunakan	Dapat menampilkan pesan kesalahan “*Email sudah dipakai*”	Menampilkan pesan kesalahan “*Email sudah dipakai*”	[√] Diterima [ ] Ditolak
Sandi tidak diisi	Dapat menampilkan pesan kesalahan “*kolom ini harus diisi*”	Menampilkan pesan kesalahan “*kolom ini harus diisi*”	[√] Diterima [ ] Ditolak
Nama Lengkap tidak diisi	Dapat menampilkan pesan kesalahan “*kolom ini harus diisi*”	Menampilkan pesan kesalahan “*kolom ini harus diisi*”	[√] Diterima [ ] Ditolak
Alamat tidak diisi	Dapat menampilkan pesan kesalahan “*kolom ini harus diisi*”	Menampilkan pesan kesalahan ” *kolom ini harus diisi*”	[√] Diterima [ ] Ditolak
Provinsi belum dipilih	Dapat menampilkan pesan kesalahan	Menampilkan pesan kesalahan “*kolom ini harus*	[√] Diterima [ ] Ditolak

	"*kolom ini harus diisi*"	*diisi*"	
Kota belum dipilih	Dapat menampilkan pesan kesalahan "*kolom ini harus diisi*"	Menampilkan pesan kesalahan "*kolom ini harus diisi*"	[√] Diterima [ ] Ditolak
Kode Pos belum diisi	Dapat menampilkan pesan kesalahan "*kolom ini harus diisi*"	Menampilkan pesan kesalahan "*kolom ini harus diisi*"	[√] Diterima [ ] Ditolak

4.4.3.2 Pengujian *Login* Pelanggan

Tabel berikut akan menjelaskan mengenai pengujian *login* pelanggan dengan kasus & hasil uji data benar dan data salah.

Table 4-9 Pengujian login pelanggan

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data *login*, contoh : *Email* : edgarprima@gmail.com Sandi : 1234567	Jika data *login* valid, maka pelanggan akan masuk ke dalam sistem	Semua *field* terisi/benar dalam pengisian, maka pelanggan akan masuk ke dalam sistem	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			

Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengosongkan semua *field* dalam *form login*	Dapat menampilkan pesan kesalahan "*email dan password salah, silahkan login kembali*"	Menampilkan pesan kesalahan " *email dan password salah, silahkan login kembali*"	[√] Diterima [ ] Ditolak
Mengisikan alamat *email* yang tidak terdaftar	Dapat menampilkan pesan kesalahan bahwa "*email dan password salah, silahkan login kembali"*	Menampilkan pesan kesalahan "*email dan password salah, silahkan login kembali*".	[√] Diterima [ ] Ditolak
Mengisikan alamat *email* yang tidak valid	Dapat menampilkan pesan kesalahan bahwa "*email dan password salah, silahkan login kembali"*	Menampilkan pesan kesalahan "*email dan password salah, silahkan login kembali"*	[√] Diterima [ ] Ditolak
Sandi yang diisi tidak valid	Dapat menampilkan pesan kesalahan "*email dan password salah, silahkan login kembali"*	Menampilkan pesan kesalahan "*email dan password salah, silahkan login kembali"*	[√] Diterima [ ] Ditolak

4.4.3.3 Pengujian Reset Password

Tabel berikut akan menjelaskan mengenai pengujian Reset Password pelanggan dengan kasus & hasil uji data benar dan data salah.

Table 4-10 Pengujian reset password

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data lupa sandi, contoh : *Email* : edgarprima@gmail.com	Jika data pelanggan ditemukan, maka sistem akan memberikan link untuk reset password	Data pelanggan ditemukan, sistem memberikan link untuk reset password	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Email belum diisi	Dapat menampilkan pesan kesalahan "*email masih kosong*"	Menampilkan pesan kesalahan " *email masih kosong*"	[√] Diterima [ ] Ditolak
Email yang diisi belum terdaftar	Dapat menampilkan pesan "*Email yang Anda masukkan tidak terdaftar di sistem kami*"	Menampilkan pesan kesalahan "*Email yang Anda masukkan tidak terdaftar di sistem kami*"	[√] Diterima [ ] Ditolak

Email yang diisi tidak valid	Dapat menampilkan pesan kesalahan "*Email yang Anda masukkan tidak valid*"	Menampilkan pesan kesalahan "*Email yang Anda masukkan tidak valid*"	[√] Diterima [ ] Ditolak

4.4.3.4 Pengujian Data Pelanggan

Tabel berikut akan menjelaskan mengenai pengujian data pelanggan dengan kasus & hasil uji data benar dan data salah.

Table 4-11 Pengujian data pelanggan

asus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengubah data pelanggan, contoh : nama lengkap : edgar prima alamat : jln. Gagak provinsi : jawa barat kota : bandung kode pos : 40123 no telepon : 085234567896 email : edgarprima@gmail.com password : 1234567 ulangi password : 1234567	Jika data lengkap dan tepat, sistem akan menyimpan data pelanggan ke dalam *database*	Data lengkap dan tepat, sistem menyimpan data pelanggan ke dalam *database*	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Lengkap tidak diisi	Dapat menampilkan pesan kesalahan "*kolom ini masih kosong*"	Menampilkan pesan kesalahan "*kolom ini masih kosong*"	[√] Diterima [ ] Ditolak
Alamat tidak diisi	Dapat menampilkan pesan kesalahan "*kolom ini masih kosong*"	Menampilkan pesan kesalahan "*kolom ini masih kosong*"	[√] Diterima [ ] Ditolak
Provinsi belum dipilih	Dapat menampilkan pesan kesalahan "*kolom ini masih kosong*"	Menampilkan pesan kesalahan "*kolom ini masih kosong*"	[√] Diterima [ ] Ditolak
Kota belum dipilih	Dapat menampilkan pesan kesalahan "*kolom ini masih kosong*"	Menampilkan pesan kesalahan "*kolom ini masih kosong*"	[√] Diterima [ ] Ditolak
Kode Pos belum diisi	Dapat menampilkan pesan kesalahan "*kolom ini masih kosong*"	Menampilkan pesan kesalahan "*kolom ini masih kosong*"	[√] Diterima [ ] Ditolak

4.4.3.5 Pengujian pesan produk

Tabel berikut akan menjelaskan mengenai pesan produk dengan kasus & hasil uji data benar dan data salah.

Table 4-12 pengujian pesan produk

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Menambah jumlah beli contoh : Jumlah : 5	Jika data sesuai, sistem akan menyimpan data jumlah beli	Data sesuai, sistem menyimpan data jumlah beli	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Jumlah yang hendak dibeli diisi dengan nilai 0	Dapat menampilkan pesan kesalahan ”*Jumlah pembelian tidak boleh 0*”	Menampilkan pesan kesalahan “*Jumlah pembelian tidak boleh 0*”	[√] Diterima [ ] Ditolak
Jumlah yang hendak dibeli diisi dengan alfabet	Tidak dapat mengisi *field* jumlah dengan alfabet	Tidak dapat mengisi *field* jumlah dengan alfabet	[√] Diterima [ ] Ditolak

4.4.3.6 Pengujian Alamat Pengiriman

Tabel berikut akan menjelaskan mengenai pengujian alamat pengiriman dengan kasus & hasil uji data benar dan data salah.

Table 4-13 Pengujian alamat pengiriman

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data nama penarima : edgar Alamat penerima : jln gagak Provinsi : jawa barat Kota : bandung Jasa pengiriman : JNE Jenis paket : yes	Jika data yang diisikan lengkap, maka sistem akan memproses data tersebut dan menyimpannya	Data yang diisikan lengkap, sistem memproses data tersebut dan menyimpannya	√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama penerima belum diisi	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan "*kolom wajib diisi*"	[√] Diterima [ ] Ditolak
Alamat belum diisi	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan "*kolom wajib diisi*"	[√] Diterima [ ] Ditolak

Kode Pos belum diisi	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan "*A kolom wajib diisi*"	[√] Diterima [ ] Ditolak
Provinsi belum dipilih	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan "*kolom wajib diisi"*	[√] Diterima [ ] Ditolak
Kota belum dipilih	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan "*kolom wajib diisi*"	[√] Diterima [ ] Ditolak
Jasa Kirim belum dipilih	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan "*kolom wajib diisi*"	[√] Diterima [ ] Ditolak
Paket belum dipilih	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan "*kolom wajib diisi*"	[√] Diterima [ ] Ditolak

4.4.3.7 Pengujian Konfirmasi Pembayaran

Tabel berikut akan menjelaskan mengenai pengujian konfirmasi pembayaran dengan kasus & hasil uji data benar dan data salah.

Table 4-14 Pengujian konfirmasi pembayaran

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

Mengisi data konfirmasi pembayaran, contoh : Tanggal bayar : 01 Juni 2012 Metode pembayaran : transfer tunai Bank yang digunakan : BNI Jumlah pembayaran : Rp. 30.000	Jika data yang diisikan lengkap, maka sistem akan memproses data tersebut dan menyimpannya	Data yang diisikan lengkap, sistem memproses data tersebut dan menyimpannya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Bank Pengirim belum dipilih	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan " *kolom wajib diisi*"	[√] Diterima [ ] Ditolak
Tanggal pembayaran masih kosong	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan " *kolom wajib diisi*"	[√] Diterima [ ] Ditolak
Metode pembayaran masih kosong	Dapat menampilkan pesan kesalahan "*kolom wajib*	Menampilkan pesan kesalahan " *kolom wajib*	[√] Diterima [ ] Ditolak

	diisi"	*diisi*"	
Jumlah yang ditransfer belum diisi	Dapat menampilkan pesan kesalahan "*kolom wajib diisi*"	Menampilkan pesan kesalahan " *kolom wajib diisi*"	[√] Diterima [ ] Ditolak

4.4.3.8 Pengujian Login Admin

Tabel berikut akan menjelaskan mengenai pengujian login admin dengan kasus & hasil uji data benar dan data salah.

Table 4-15 pengujian login admin

asus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data *login*, contoh : *Username* : personalia Sandi : 123456	Jika data *login* valid, maka pegawai akan masuk ke dalam sistem administrator	Data *login* valid, pegawai masuk ke dalam sistem administrator	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Username belum diisi	Dapat menampilkan pesan kesalahan "*login gagal silahkan cek*	Menampilkan pesan kesalahan "*login gagal silahkan cek kembali*	[√] Diterima [ ] Ditolak

	kembali username dan password"	*username dan password*",	
Sandi belum diisi	Dapat menampilkan pesan kesalahan "*login gagal silahkan cek kembali username dan password*"	Menampilkan pesan kesalahan "*login gagal silahkan cek kembali username dan password*"	[√] Diterima [ ] Ditolak
Username yang diisi tidak terdaftar	Dapat menampilkan pesan kesalahan "*login gagal silahkan cek kembali username dan password*"	Menampilkan pesan kesalahan "*login gagal silahkan cek kembali username dan password*"	[√] Diterima [ ] Ditolak
Sandi yang diisi tidak valid	Dapat menampilkan pesan kesalahan " *login gagal silahkan cek kembali username dan password*"	Menampilkan pesan kesalahan " *login gagal silahkan cek kembali username dan password*"	[√] Diterima [ ] Ditolak

4.4.3.9 Pengujian Data Pelanggan

Tabel berikut akan menjelaskan mengenai pengujian data pelanggan dengan kasus & hasil uji data benar.

Table 4-16 Pengujian data pelanggan

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah status blokir	Jika status blokir T, maka sistem akan merubah status blokir menjadi Y, begitu juga sebaliknya	Mengubah status blokir pelanggan dari T menjadi Y, dan sebaliknya	[√] Diterima [ ] Ditolak

4.4.3.10 Pengujian Produk

4.4.3.10.1 Pengujian Tambah Produk

Tabel berikut akan menjelaskan mengenai pengujian tambah produk dengan kasus & hasil uji data benar dan data salah.

Table 4-17 Pengujian tambah produk

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data tambah produk, contoh : Kategori : kacang Nama produk : kacang atom Stok : 200 Berat : 300 gr Harga : 13000	Jika data yang dimasukkan telah lengkap dan tepat, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan lengkap dan tepat, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak

Diskon : 20 Gambar : kacang.jpg Deskripsi : enak			
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
kategori belum diisi	Dapat menampilkan pesan kesalahan "*kolom ini wajib diisi*"	Menampilkan pesan kesalahan "*kolom ini wajib diis0069*"	[√] Diterima [ ] Ditolak
Kategori Produk belum dipilih	Dapat menampilkan pesan kesalahan "*kolom ini wajib diisi*"	Menampilkan pesan kesalahan "*kolom ini wajib diisi*"	[√] Diterima [ ] Ditolak
Nama Produk belum diisi	Dapat menampilkan pesan kesalahan "*kolom ini wajib diisi*"	Menampilkan pesan kesalahan "*kolom ini wajib diisi*"	[√] Diterima [ ] Ditolak
Stok belum diisi	Dapat menampilkan pesan kesalahan "*kolom ini wajib diisi*"	Menampilkan pesan kesalahan "*kolom ini wajib diisi*"	[√] Diterima [ ] Ditolak
Bahan belum diisi	Dapat menampilkan pesan kesalahan "*Anda belum mengisikan bahan*"	Menampilkan pesan kesalahan "*Anda belum mengisikan bahan*"	[√] Diterima [ ] Ditolak
Berat belum diisi	Dapat menampilkan pesan kesalahan	Menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

	"*kolom ini wajib diisi*"	"*kolom ini wajib diisi*"	

4.4.3.10.2 Pengujian Ubah Produk

Tabel berikut akan menjelaskan mengenai pengujian ubah produk dengan kasus & hasil uji data benar.

Table 4-18 Pengujian ubah produk

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah deskripsi	Jika data yang dimasukkan tidak kosong, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan tidak kosong, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak

4.4.3.11 Pengujian Kategori

4.4.3.11.1 Pengujian Tambah Kategori

Tabel berikut akan menjelaskan mengenai pengujian tambah kategori dengan kasus & hasil uji data benar dan data salah.

Table 4-19 Pengujian tambah kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data tambah	Jika data yang	Data yang	[√] Diterima

kategori, contoh : Nama Kategori : Miniatur	dimasukkan telah lengkap dan tepat, sistem akan memproses data tersebut untuk disimpan	dimasukkan lengkap dan tepat, sistem memproses data tersebut untuk disimpan	[ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Kategori belum diisi	Dapat menampilkan pesan kesalahan "*kolom harus diisi*"	Menampilkan pesan kesalahan "*kolom harus diisi*"	[√] Diterima [ ] Ditolak

4.4.3.11.2 Pengujian Ubah Kategori

Tabel berikut akan menjelaskan mengenai pengujian ubah kategori dengan kasus & hasil uji data benar.

Table 4-20 Pengujian ubah kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengubah kategori	Jika selesai mengubah maka kategori berhasil di rubah	Mengubah mengubah kategori	[√] Diterima [ ] Ditolak

4.4.3.12 Pengujian Jasa Kirim

4.4.3.12.1 Pengujian Tambah Jasa Kirim

Tabel berikut akan menjelaskan mengenai pengujian tambah jasa kirim dengan kasus & hasil uji data benar dan data salah.

Table 4-21 Pengujian tambah jasa kirim

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data tambah jasa kirim, contoh : Nama Jasa Kirim : JNE	Jika data yang dimasukkan telah lengkap dan tepat, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan lengkap dan tepat, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Jasa Kirim belum diisi	Dapat menampilkan pesan kesalahan "*Anda belum mengisikan nama jasa kirim*"	Menampilkan pesan kesalahan "*Anda belum mengisikan nama jasa kirim*"	[√] Diterima [ ] Ditolak
Nama Jasa Kirim yang diisi sudah ada	Dapat menampilkan pesan kesalahan "*Nama jasa kirim sudah ada*"	Menampilkan pesan kesalahan "*Nama jasa kirim sudah ada*"	[√] Diterima [ ] Ditolak

4.4.3.12.2 Pengujian Ubah Jasa Kirim

Tabel berikut akan menjelaskan mengenai pengujian ubah jasa kirim dengan kasus & hasil uji data benar.

Table 4-22 Pengujian ubah jasa kirim

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah nama jasa kirim	Jika data yang dimasukkan tidak kosong, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan tidak kosong, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak

4.4.3.13 Pengujian Paket

4.4.3.13.1 Pengujian Tambah Paket

Tabel berikut akan menjelaskan mengenai pengujian tambah paket dengan kasus & hasil uji data benar dan data salah.

Table 4-23 Pengujian tambah paket

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data tambah paket, contoh :	Jika data yang dimasukkan telah	Data yang dimasukkan	[√] Diterima

Jasa Kirim : JNE Jenis Paket : Reg	lengkap dan tepat, sistem akan memproses data tersebut untuk disimpan	lengkap dan tepat, sistem memproses data tersebut untuk disimpan	[ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Jasa Kirim belum dipilih	Dapat menampilkan pesan kesalahan "*Anda belum memilih jasa kirim*"	Menampilkan pesan kesalahan "*Anda belum memilih jasa kirim*"	[√] Diterima [ ] Ditolak
Jenis Paket belum diisi	Dapat menampilkan pesan kesalahan "*Anda belum mengisikan jenis paket*"	Menampilkan pesan kesalahan "*Anda belum mengisikan jenis paket*"	[√] Diterima [ ] Ditolak

4.4.3.13.2 Pengujian Ubah Paket

Tabel berikut akan menjelaskan mengenai pengujian ubah paket dengan kasus & hasil uji data benar.

Table 4-24 Pengujian ubah paket

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan

Ubah jasa kirim	Jika data yang dimasukkan tidak kosong, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan tidak kosong, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak
Ubah jenis paket	Jika data yang dimasukkan tidak kosong, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan tidak kosong, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak

4.4.3.14 Pengujian Provinsi

4.4.3.14.1 Pengujian Tambah Provinsi

Tabel berikut akan menjelaskan mengenai pengujian tambah provinsi dengan kasus & hasil uji data benar dan data salah.

Table 4-25 Pengujian tambah provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data tambah provinsi, contoh : Provinsi : Jawa Barat	Jika data yang dimasukkan telah lengkap dan tepat, sistem akan memproses data tersebut untuk	Data yang dimasukkan lengkap dan tepat, sistem memproses data tersebut untuk	[√] Diterima [ ] Ditolak

	disimpan	disimpan	
Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Provinsi tidak diisi	Dapat menampilkan pesan kesalahan "*Anda belum mengisikan provinsi*"	Menampilkan pesan kesalahan "*Anda belum mengisikan provinsi*"	[√] Diterima [ ] Ditolak

4.4.3.14.2 Pengujian Ubah Provinsi

Tabel berikut akan menjelaskan mengenai pengujian ubah provinsi dengan kasus & hasil uji data benar.

Table 4-26 Pengujian ubah provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah provinsi	Jika data yang dimasukkan tidak kosong, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan tidak kosong, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak

4.4.3.15 Pengujian Kota

4.4.3.15.1 Pengujian Tambah Kota

Tabel berikut akan menjelaskan mengenai pengujian tambah kota dengan kasus & hasil uji data benar dan data salah.

Table 4-27 Pengujian tambah kota

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data tambah provinsi, contoh : Provinsi : Jawa Barat Kota : Kota Bandung	Jika data yang dimasukkan telah lengkap dan tepat, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan lengkap dan tepat, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji Salah (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Provinsi belum dipilih	Dapat menampilkan pesan kesalahan "*Anda belum memilih provinsi*"	Menampilkan pesan kesalahan "*Anda belum memilih provinsi*"	[√] Diterima [ ] Ditolak
Kota belum diisi	Dapat menampilkan pesan kesalahan "*Anda belum mengisikan kabupaten kota*"	Menampilkan pesan kesalahan "*Anda belum mengisikan kabupaten kota*"	[√] Diterima [ ] Ditolak

4.4.3.15.2 Pengujian Ubah Kota

Tabel berikut akan menjelaskan mengenai pengujian ubah kota dengan kasus & hasil uji data benar.

Table 4-28 Pengujian ubah kabupaten kota

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Ubah kota	Jika data yang dimasukkan tidak kosong, sistem akan memproses data tersebut untuk disimpan	Data yang dimasukkan tidak kosong, sistem memproses data tersebut untuk disimpan	[√] Diterima [ ] Ditolak

4.4.4 Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian yang telah dilakukan terhadap aplikasi *e-commerce*, dapat disimpulkan bahwa setiap proses yang ada pada aplikasi ini secara fungsional dapat digunakan dan menghasilkan *output* yang diharapkan.

4.5 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif, dimana pengujian dilakukan secara langsung yaitu terjun ke toko yang bersangkutan dengan membagikan kuesioner yang telah dibuat guna memperoleh data tentang kepuasan pengguna terhadap penelitian penerapan sistem yang baru, dengan kandungan poin syarat *user friendly*. Adapun metode penelitian yang digunakan yaitu metode kuantitatif berdasarkan data sample dari pengguna.

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

1. Linkungan penjual

   Penelitian dilakukan dilingkungan penjual agar diketahui sejauh mana sistem yang telah dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan diawal.

2. Lingkungan masyarakat umum

   Penelitian dilakukan dilingkungan masyarakat umum agar diketahui sejauh mana sistem yang dibangun dapat memberikan kemudahan bagi masyarakat umum dalam melakukan proses pembelian barang secara *online*.

4.5.1 Kuesioner Pengguna

Kuesioner disebarkan menggunakan teknik sampling yaitu, *Simple Random Sampling* yang disebarkan kepada 30 orang pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuesioner terdiri dari 5 pernyataan (contoh kuesioner dapat dilihat dibagian lampiran) dengan menggunakan skala 1 sampai 5 (skala likert), dengan ketentuan skala untuk tiap peryataan berikut :

4.5.1.1 Kuesioner Pengujian Beta Untuk Pelanggan

Untuk mencari nilai persentase dari masing-masing jawaban kuesioner digunakan rumus skala likert.

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuesioner ini diujikan kepada 30 orang.

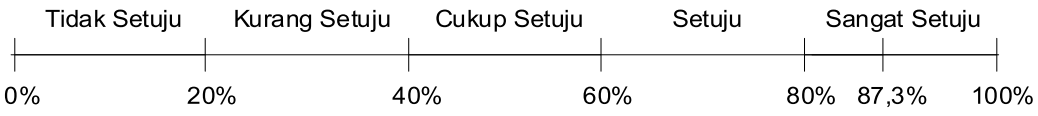
1. Tampilan dari aplikasi ini menarik dan mudah dipahami oleh Saya

Table 4-29 Hasil pengujian kuesioner pelanggan soal no 1

Pertanyaan	Keterangan	N	R	N x R
1	Sangat Setuju	5	14	70
	Setuju	4	13	52
	Cukup Setuju	3	3	9
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**131**

$$Y = \frac{131}{30 \times 5} \times 100\% = 87{,}3\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap tampilan dari aplikasi ini menarik dan mudah dipahami adalah 87,3%, atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



2. Saya merasa mudah dalam menggunakan aplikasi ini

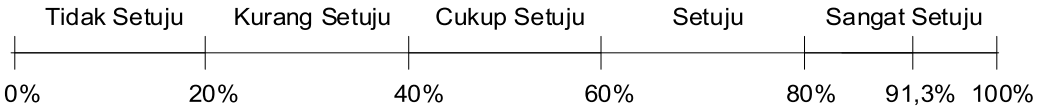
Table 4-30 Hasil pengujian kuesioner pelanggan soal no 2

Pertanyaan	Keterangan	N	R	N x R

2	Sangat Setuju	5	18	90
	Setuju	4	11	44
	Cukup Setuju	3	1	3
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**137**

$$Y = \frac{137}{30 \times 5} \times 100\% = 91{,}3\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi ini mudah dalam pengoperasiannya adalah 91,3%, atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



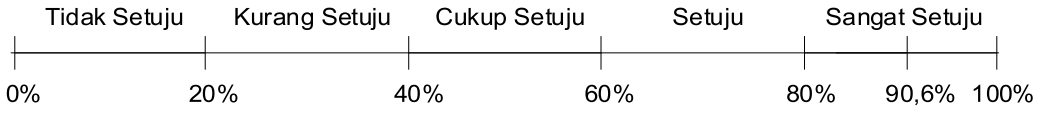
3. Saya merasa senang setiap kali berkunjung menggunakan aplikasi ini

Table 4-31 Hasil pengujian kuesioner pelanggan soal no 3

Pertanyaan	Keterangan	N	R	N x R
3	Sangat Setuju	5	16	80
	Setuju	4	14	56
	Cukup Setuju	3	0	0
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**136**

$$Y = \frac{136}{30 \times 5} \times 100\% = 90{,}6\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap pengguna merasa senang setiap kali berkunjung menggunakan aplikasi ini adalah 90,6%, atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



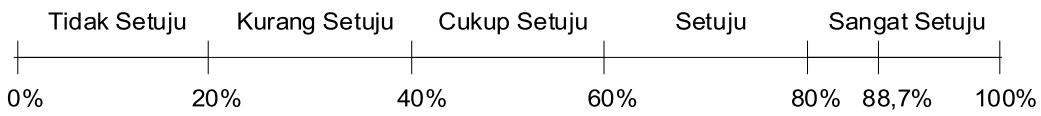
4. Aplikasi ini memberikan informasi mengenai stok, keterangan, deskripsi barang dan informasi lain yang Saya butuhkan

Table 4-32 Hasil pengujian kuesioner pelanggan soal no 4

Pertanyaan	Keterangan	N	R	N x R
4	Sangat Setuju	5	15	75
	Setuju	4	13	52
	Cukup Setuju	3	2	6
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**133**

$$Y = \frac{133}{30 \times 5} \times 100\% = 88{,}7\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi ini memberikan informasi mengenai stok, keterangan, deskripsi barang dan informasi lain yang pengguna butuhkan adalah 88,7%, atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



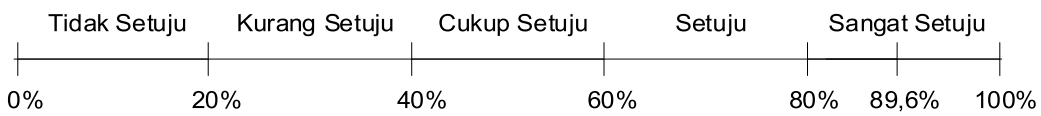
5. Aplikasi ini memudahkan Saya dalam melakukan transaksi pembelian barang yang Saya inginkan tanpa harus datang ke toko

Table 4-33 Hasil pengujian kuesioner pelanggan soal no 5

Pertanyaan	Keterangan	N	R	N x R
5	Sangat Setuju	5	15	75
	Setuju	4	14	56
	Cukup Setuju	3	1	3
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**30**	**134**

$$Y = \frac{134}{30 \times 5} \times 100\% = 89{,}3\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap aplikasi ini memudahkan pengguna dalam melakukan pembelian barang yang pengguna inginkan tanpa harus datang ke toko adalah 89,3%, atau bisa dikategorikan sebagai sangat setuju. Berikut kriteria interpretasi skor :



4.5.2 Wawancara Pegawai

Wawancara dilakukan dengan menggunakan metode wawancara mendalam, yang nantinya akan diajukan pertanyaan-pertanyaan secara bebas dan leluasa mengenai penilaian penerapan sistem yang baru, yang tidak membuat kejenuhan di kedua belah pihak, sehingga diperoleh informasi yang lebih kaya. Dari hasil jawaban wawancara tersebut akan ditarik sebuah kesimpulan terhadap penilaian penerapan sistem yang baru. Berikut ini adalah hasil wawancara yang dilakukan kepada pegawai toko :

Bagian Kasir :

Penampilan website ini terbilang cukup menarik. Hal ini membuat saya tertarik dan bersemangat untuk menggunakannya. Tampilan yang sederhana dan efisien membuat website ini menarik untuk dipelajari. Pengoperasiannya juga cukup sederhana. Program ini secara otomatis memberikan dan mengelola data-data yang saya perlukan saja, yaitu data pegawai dan pelanggan website. Jadi, saya bisa mengetahui siapa saja pelanggan yang berbelanja di website ini. Walau begitu, aksesnya yang lambat membuat saya harus bersabar menunggu *loading*-nya. Semuanya pasti ada kelebihan dan kekurangannya, jadi tidak masalah buat saya.

Bagian Pemilik Toko :

Program pengadaan website ini bermanfaat untuk saya dalam melakukan pengecekan laporan keuangan toko online, kapanpun saya mau, tanpa perlu merepotkan pihak lain untuk membuatkan laporan tersebut. Tampilannya sederhana dan bagus, mempermudah saya dan pegawai untuk mengelolanya.

4.5.3 Kesimpulan Hasil Pengujian Beta

Berdasarkan hasil persentase yang diperoleh dari pengujian beta pengguna dan wawancara pegawai, sebagian besar mengatakan sangat setuju terhadap penerapan sistem yang baru telah memiliki kandungan poin syarat *user friendly* dan sesuai dengan tujuan yang ingin dicapai, yaitu mempermudah proses pengelolaan dalam bertransaksi di toko, memudahkan toko dalam berbagi informasi kepada pelanggan mengenai berbagai macam makanan di toko sawargi, Untuk memperluas jangkauan promosi dan pemasaran dengan media internet, Agar lebih mudah dalam mendapatkan informasi mengenai jumlah persediaan barang atau stok barang dan memudahkan pihak toko dalam membuat laporan penjualan.