BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi

Tahap implementasi merupakan tahap penciptaan perangkat lunak, tahap lanjutan dari perancangan sistem. Sistem akan diuji untuk kesiapan dalam pengoperasiannya, yang terdiri dari penjelasan mengenai lingkungan implementasi, dan implementasi program.

4.1.1 Perangkat Keras

Spesifikasi perangkat keras yang digunakan dalam pembangunan *website E-commerce* pada Toko dua saudara adalah tercantum pada tabel 4.1 sebagai berikut.

Tabel 4.1 Perangkat keras yang digunakan

No	Perangkat	Spesifikasi
1.	*Processor*	Intel Core i3-2330m
2.	RAM	2 GB
3.	Harddisk	500 GB
4.	Monitor	Resolusi 1366 x 768
5.	VGA	1 GB

4.1.2 Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan dalam pembangunan *website e-commerce* pada Toko dua saudara adalah sebagai berikut.

Tabel 4.2 Perangkat lunak yang digunakan

No.	Perangkat Lunak	Keterangan
1.	Sistem Operasi	Windows xp sp3
2.	Bahasa Pemrograman	PHP
3.	Web Server	XAMPP 1.7.3
4.	Database Server	MySQL
5.	Web Browser	Mozilla firefox dan Google Chrome

6.	Code editor	Adobe Dreamweaver 8
7.	DFD modeler	Microsoft Visio 2007
8.	Image Editor	Adobe Photoshop CS3

4.2 Implementasi Basis Data

Pada tahap awal pembangunan sebuah aplikasi, dibutuhkan media penyimpanan data (*database*), implementasi database untuk sistem penerapan *E-commerce* untuk sistem penjualan adalah sebagai berikut :

4.1 Tabel Admin

```
CREATE TABLE IF NOT EXISTS `admin` (
  `id_admin` int(11) NOT NULL auto_increment,
  `nama_admin` varchar(50) NOT NULL,
  `alamat_admin` text NOT NULL,
  `email_admin` varchar(50) NOT NULL,
  `telp_admin` varchar(20) NOT NULL,
  `password_admin` varchar(32) NOT NULL,
  `verification_admin` varchar(32) NOT NULL,
  `status_login` enum('0','1') NOT NULL,
  `waktu_login` int(11) NOT NULL,
  PRIMARY KEY  (`id_admin`),
  UNIQUE KEY `email_admin` (`email_admin`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Administrator' AUTO_INCREMENT=11
```

4.2 Tabel barang

```
CREATE TABLE IF NOT EXISTS `barang` (
  `id_barang` int(4) unsigned zerofill NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `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` float 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`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Barang' AUTO_INCREMENT=15 ;
ALTER TABLE `barang`
  ADD CONSTRAINT `barang_ibfk_1` FOREIGN KEY (`id_kategori`)
REFERENCES `kategori` (`id_kategori`) ON DELETE CASCADE ON
UPDATE CASCADE,
  ADD CONSTRAINT `barang_ibfk_2` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.3 Tabel barangdetail

```
CREATE TABLE IF NOT EXISTS `barangdetail` (
  `id_barangdetail` int(11) NOT NULL auto_increment,
  `id_barang` int(4) unsigned zerofill NOT NULL,
  `id_warna` int(11) default NULL,
  `id_ukuran` int(11) NOT NULL,
  `tanggal_barangdetail` datetime NOT NULL,
  `stok_barangdetail` int(11) NOT NULL,
  `barang_terjual` int(20) NOT NULL,
  `jumlah_retur` 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_ukuran` (`id_ukuran`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Detail Barang' AUTO_INCREMENT=8 ;
ALTER TABLE `barangdetail`
  ADD CONSTRAINT `barangdetail_ibfk_4` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`) ON DELETE
CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `barangdetail_ibfk_5` FOREIGN KEY
(`id_warna`) REFERENCES `warna` (`id_warna`) ON DELETE
CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `barangdetail_ibfk_6` FOREIGN KEY
(`id_ukuran`) REFERENCES `ukuran` (`id_ukuran`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

4.4 Berita

```
CREATE TABLE IF NOT EXISTS `berita` (
  `id_berita` int(11) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `judul_berita` varchar(100) NOT NULL,
  `isi_berita` text NOT NULL,
  `publish` int(11) NOT NULL,
  PRIMARY KEY  (`id_berita`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=5 ;
ALTER TABLE `berita`
  ADD CONSTRAINT `berita_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.5 Config

```
CREATE TABLE IF NOT EXISTS `config` (
  `id_config` int(11) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `nama_config` varchar(100) NOT NULL,
  `deskripsi_config` text NOT NULL,
  PRIMARY KEY  (`id_config`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel Config' AUTO_INCREMENT=2 ;
ALTER TABLE `config`
  ADD CONSTRAINT `config_ibfk_1` FOREIGN KEY (`id_admin`) REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.6 Detailpembelian

```
CREATE TABLE IF NOT EXISTS `detailpembelian` (
  `id_detailpembelian` int(11) NOT NULL auto_increment,
  `tanggal_detailpembelian` datetime NOT NULL,
  `tgl_diterima` datetime NOT NULL,
  `jenis_pembayaran` enum('paypal','transfer bank') default NULL,
  `status_pengiriman`
enum('dipesan','dikonfirmasi','dibayar','dikirim','diterima','dibatalkan') NOT NULL,
  `id_forwarder` int(20) 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(20) NOT NULL,
  `id_ongkir` int(20) NOT NULL,
  `id_jenispengiriman` int(20) default NULL,
  `status_pemesanan` enum('ok','cancel') NOT NULL,
  `session_id` varchar(32) NOT NULL,
  PRIMARY KEY  (`id_detailpembelian`),
  KEY `id_jenispengiriman` (`id_jenispengiriman`),
  KEY `id_rekening` (`id_rekening`),
  KEY `id_forwarder` (`id_forwarder`),
  KEY `id_ongkir` (`id_ongkir`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
detailpembelian' AUTO_INCREMENT=37 ;
```

4.7 Detail_retur

```
CREATE TABLE IF NOT EXISTS `detail_retur` (
  `id_retur` int(11) default NULL,
  `id_detailpembelian` int(11) NOT NULL,
  `id_barangdetail` int(11) NOT NULL,
  `qty_retur` int(11) NOT NULL,
  `harga` int(11) NOT NULL,
  `komplain` text NOT NULL,
   `deskripsi` text NOT NULL,
  `session_id` varchar(32) NOT NULL,
  KEY `id_detailpembelian` (`id_detailpembelian`),
  KEY `id_barangdetail` (`id_barangdetail`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `detail_retur`
  ADD CONSTRAINT `detail_retur_ibfk_1` FOREIGN KEY
(`id_detailpembelian`) REFERENCES `detailpembelian`
```

```
(`id_detailpembelian`) ON DELETE CASCADE ON UPDATE CASCADE,
 ADD CONSTRAINT `detail_retur_ibfk_2` FOREIGN KEY (`id_barangdetail`) REFERENCES `barangdetail` (`id_barangdetail`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.8 Forwarder

```
CREATE TABLE IF NOT EXISTS `forwarder` (
 `id_forwarder` int(11) NOT NULL auto_increment,
 `id_admin` int(11) NOT NULL,
 `nama_forwarder` varchar(30) NOT NULL,
 PRIMARY KEY  (`id_forwarder`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
ALTER TABLE `forwarder`
 ADD CONSTRAINT `forwarder_ibfk_1` FOREIGN KEY (`id_admin`) REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.9 Jenispengiriman

```
CREATE TABLE IF NOT EXISTS `jenispengiriman` (
 `id_jenispengiriman` int(11) NOT NULL auto_increment,
 `id_forwarder` int(11) NOT NULL,
 `id_admin` int(11) NOT NULL,
 `nama_jenispengiriman` varchar(50) NOT NULL,
 `deskripsi_jenispengiriman` text NOT NULL,
 PRIMARY KEY  (`id_jenispengiriman`),
 KEY `id_forwarder` (`id_forwarder`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel Jenis Pengiriman' AUTO_INCREMENT=7 ;
ALTER TABLE `jenispengiriman`
```

```
 ADD CONSTRAINT `jenispengiriman_ibfk_1` FOREIGN KEY
(`id_forwarder`) REFERENCES `forwarder` (`id_forwarder`) ON
DELETE CASCADE ON UPDATE CASCADE,
 ADD CONSTRAINT `jenispengiriman_ibfk_2` FOREIGN KEY
(`id_admin`) REFERENCES `admin` (`id_admin`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

4.10 Kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `kode_kategori` varchar(3) NOT NULL,
  `nama_kategori` varchar(50) NOT NULL,
  `deskripsi_kategori` text NOT NULL,
  PRIMARY KEY  (`id_kategori`),
  UNIQUE KEY `nama_kategori` (`nama_kategori`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Kategori' AUTO_INCREMENT=4 ;
ALTER TABLE `kategori`
 ADD CONSTRAINT `kategori_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.11 Konfirmasipembayaran

```
CREATE TABLE IF NOT EXISTS `konfirmasipembayaran` (
  `id_konfirmasipembayaran` int(11) NOT NULL auto_increment,
  `id_pembelian` 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` datetime NOT NULL,
  PRIMARY KEY  (`id_konfirmasipembayaran`),
  KEY `id_pembelian` (`id_pembelian`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Konfirmasi' AUTO_INCREMENT=9 ;
ALTER TABLE `konfirmasipembayaran`
  ADD CONSTRAINT `konfirmasipembayaran_ibfk_1` FOREIGN KEY
(`id_pembelian`) REFERENCES `pembelian` (`id_detailpembelian`) ON
DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `konfirmasipembayaran`
  ADD CONSTRAINT `konfirmasipembayaran_ibfk_1` FOREIGN KEY
(`id_pembelian`) REFERENCES `pembelian` (`id_detailpembelian`) ON
DELETE CASCADE ON UPDATE CASCADE;
```

4.12 Konfterima

```
CREATE TABLE IF NOT EXISTS `konfterima` (
  `id_konfterima` int(11) NOT NULL auto_increment,
  `id_pembelian` int(11) NOT NULL,
  `nama_penerima` varchar(30) NOT NULL,
  `no_kirim` varchar(20) NOT NULL,
  `total_bayar` int(11) NOT NULL,
  `tanggal_penerimaan` datetime NOT NULL,
  PRIMARY KEY  (`id_konfterima`),
  KEY `id_pembelian` (`id_pembelian`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=4 ;
ALTER TABLE `konfterima`
  ADD CONSTRAINT `konfterima_ibfk_1` FOREIGN KEY
(`id_pembelian`) REFERENCES `pembelian` (`id_detailpembelian`) ON
DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `konfterima`
```

```
 ADD CONSTRAINT `konfterima_ibfk_1` FOREIGN KEY (`id_pembelian`) REFERENCES `pembelian` (`id_detailpembelian`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.13 Kota

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(10) NOT NULL auto_increment,
  `id_provinsi` int(10) default NULL,
  `id_admin` int(11) NOT NULL,
  `nama_kota` varchar(50) default NULL,
  `kabkota` varchar(20) default NULL,
  PRIMARY KEY  (`id_kota`),
  UNIQUE KEY `kota#PX` (`id_kota`),
  KEY `id_provinsi` (`id_provinsi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1001 ;
ALTER TABLE `kota`
 ADD CONSTRAINT `kota_ibfk_1` FOREIGN KEY (`id_provinsi`) REFERENCES `provinsi` (`id_provinsi`) ON DELETE CASCADE ON UPDATE CASCADE;
ALTER TABLE `kota`
 ADD CONSTRAINT `kota_ibfk_1` FOREIGN KEY (`id_provinsi`) REFERENCES `provinsi` (`id_provinsi`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.14 Kurs

```
CREATE TABLE IF NOT EXISTS `kurs` (
  `id_kurs` int(11) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `rp_kurs` varchar(10) NOT NULL,
```

```
 `flag_kurs` timestamp NOT NULL default CURRENT_TIMESTAMP
on update CURRENT_TIMESTAMP,
 PRIMARY KEY  (`id_kurs`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
AUTO_INCREMENT=2 ;
ALTER TABLE `kurs`
 ADD CONSTRAINT `kurs_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.15 Member

```
CREATE TABLE IF NOT EXISTS `member` (
 `id_member` int(11) NOT NULL auto_increment,
 `id_kota` int(11) NOT NULL,
 `nama_member` varchar(50) NOT NULL,
 `alamat_member` text NOT NULL,
 `ttl_member` date NOT NULL,
 `telp_member` varchar(20) NOT NULL,
 `kodepos_member` varchar(6) NOT NULL,
 `email_member` varchar(50) NOT NULL,
 `password_member` varchar(32) NOT NULL,
 `verificationcode_member` varchar(32) NOT NULL,
 `status_member` enum('0','1') NOT NULL,
 PRIMARY KEY  (`id_member`),
 UNIQUE KEY `email_member` (`email_member`),
 KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Member' AUTO_INCREMENT=121 ;
ALTER TABLE `member`
```

```
 ADD CONSTRAINT `member_ibfk_1` FOREIGN KEY (`id_kota`)
REFERENCES `kota` (`id_kota`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

4.16 Member_login

```
CREATE TABLE IF NOT EXISTS `member_login` (
 `id_member` int(11) NOT NULL,
 `waktu` datetime NOT NULL,
 KEY `id_member` (`id_member`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `member_login`
 ADD CONSTRAINT `member_login_ibfk_1` FOREIGN KEY
(`id_member`) REFERENCES `member` (`id_member`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

4.17 Ongkir

```
CREATE TABLE IF NOT EXISTS `ongkir` (
 `id_ongkir` int(11) NOT NULL auto_increment,
 `id_kota` int(11) NOT NULL,
 `id_jenispengiriman` int(11) NOT NULL,
 `id_admin` int(11) NOT NULL,
 `harga_ongkir` int(11) NOT NULL,
 PRIMARY KEY (`id_ongkir`),
 KEY `id_jenispengiriman` (`id_jenispengiriman`),
 KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Tabel
Ongkos Kirim' AUTO_INCREMENT=26 ;
ALTER TABLE `ongkir`
 ADD CONSTRAINT `ongkir_ibfk_1` FOREIGN KEY
(`id_jenispengiriman`) REFERENCES `jenispengiriman`
(`id_jenispengiriman`) ON DELETE CASCADE ON UPDATE
CASCADE,
```

```
 ADD CONSTRAINT `ongkir_ibfk_2` FOREIGN KEY (`id_kota`) REFERENCES `kota` (`id_kota`) ON DELETE CASCADE ON UPDATE CASCADE,
 ADD CONSTRAINT `ongkir_ibfk_3` FOREIGN KEY (`id_admin`) REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.18 Pembelian

```
CREATE TABLE IF NOT EXISTS `pembelian` (
 `id_pembelian` int(11) NOT NULL auto_increment,
 `id_detailpembelian` int(11) default NULL,
 `id_member` int(11) default NULL,
 `id_barangdetail` int(11) NOT NULL,
 `harga_temp` int(11) NOT NULL,
 `stok_temp` int(11) NOT NULL,
 `qty_retur` int(11) NOT NULL,
 `berat_temp` float NOT NULL,
 `status_retur` enum('0','1','2') NOT NULL,
 `session_id` varchar(32) NOT NULL,
 PRIMARY KEY  (`id_pembelian`),
 KEY `id_detailpembelian` (`id_detailpembelian`),
 KEY `id_member` (`id_member`),
 KEY `id_barangdetail` (`id_barangdetail`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel Pembelian' AUTO_INCREMENT=41 ;
ALTER TABLE `pembelian`
 ADD CONSTRAINT `pembelian_ibfk_3` FOREIGN KEY (`id_detailpembelian`) REFERENCES `detailpembelian` (`id_detailpembelian`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.19 Provinsi

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(10) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `nama_provinsi` varchar(30) default NULL,
  PRIMARY KEY  (`id_provinsi`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=34 ;
ALTER TABLE `provinsi`
  ADD CONSTRAINT `provinsi_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.20 Rating

```
CREATE TABLE IF NOT EXISTS `rating` (
  `id_rating` int(11) NOT NULL auto_increment,
  `id_member` int(11) NOT NULL,
  `id_barang` int(11) NOT NULL,
  `nilai_rating` int(20) default NULL,
  PRIMARY KEY  (`id_rating`),
  KEY `id_member` (`id_member`),
  KEY `id_barang` (`id_barang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=1 ;
ALTER TABLE `rating`
  ADD CONSTRAINT `rating_ibfk_2` FOREIGN KEY (`id_barang`)
REFERENCES `barang` (`id_barang`) ON DELETE CASCADE ON
UPDATE CASCADE,
  ADD CONSTRAINT `rating_ibfk_1` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.21 Received

```
CREATE TABLE IF NOT EXISTS `received` (
  `id_received` int(11) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `nama_toko` varchar(50) NOT NULL,
  `pemilik` varchar(50) NOT NULL,
  `no_telp` varchar(100) NOT NULL,
  `no_telp_toko` varchar(32) NOT NULL,
  `logo_toko` varchar(200) NOT NULL,
  PRIMARY KEY  (`id_received`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
received' AUTO_INCREMENT=3 ;
ALTER TABLE `received`
  ADD CONSTRAINT `received_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.22 Rekening

```
CREATE TABLE IF NOT EXISTS `rekening` (
  `id_rekening` int(11) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `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 COMMENT='Tabel
Rekening' AUTO_INCREMENT=4 ;
ALTER TABLE `rekening`
```

```
  ADD CONSTRAINT `rekening_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.23 Retur

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(11) NOT NULL auto_increment,
  `id_member` 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 `id_member` (`id_member`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1
AUTO_INCREMENT=13 ;
LTER TABLE `retur`
  ADD CONSTRAINT `retur_ibfk_1` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`) ON DELETE CASCADE ON
UPDATE CASCADE;
```

4.24 Ukuran

```
CREATE TABLE IF NOT EXISTS `ukuran` (
  `id_ukuran` int(11) NOT NULL auto_increment,
  `id_admin` int(11) NOT NULL,
  `nama_ukuran` varchar(10) NOT NULL,
  `deskripsi_ukuran` text NOT NULL,
  PRIMARY KEY  (`id_ukuran`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Ukuran' AUTO_INCREMENT=6 ;
```

```
ALTER TABLE `ukuran`
 ADD CONSTRAINT `ukuran_ibfk_1` FOREIGN KEY (`id_admin`) REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON UPDATE CASCADE;
```

4.25 Warna

```
CREATE TABLE IF NOT EXISTS `warna` (
 `id_warna` int(11) NOT NULL auto_increment,
`id_admin` int(11) NOT NULL,
 `nama_warna` varchar(50) NOT NULL,
 `format_warna` varchar(10) default NULL,
 PRIMARY KEY (`id_warna`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Tabel Warna' AUTO_INCREMENT=14 ;
ALTER TABLE `warna`
 ADD CONSTRAINT `warna_ibfk_1` FOREIGN KEY (`id_admin`) REFERENCES `admin` (`id_admin`) ON DELETE CASCADE ON UPDATE CASCADE;
```

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 pengguna. Adapun untuk tampilan implementasi akan dilampirkan pada halaman lampiran.

Tabel 4.3 Implementasi antarmuka admin

Menu	Deskripsi	Nama file
login Admin	Login sebagai admin	login.php
Edit Akun: Ubah Biodata	Digunakan jika administrator ingin merubah biodata	edit.account.php

Ganti Kata Sandi	Digunakan jika administrator ingin merubah *password*	
Pengolahan Data Kategori	Digunakan untuk melihat, menambah, merubah dan menghapus data kategori.	kategori.php
Pengolahan Data Barang	Digunakan untuk melihat, menambah, merubah dan menghapus data barang.	barang.php
Pengolahan Data Ukuran	Digunakan untuk melihat, menambah, merubah dan menghapus data ukuran.	ukuran.php
Pengolahan Data Warna	Digunakan untuk melihat, menambah, merubah dan menghapus data warna.	warna.php
Pengolahan Data Barang Detail	Digunakan untuk melihat, menambah, merubah dan menghapus data barang detail.	detailbarang.php
Pengolahan Data Provinsi	Digunakan untuk melihat, menambah, merubah dan menghapus data provinsi.	provinsi.php
Pengolahan Data Kota	Digunakan untuk melihat, menambah, merubah dan menghapus data kota.	kota.php
Pengolahan Data Ongkos Kirim	Digunakan untuk melihat, menambah, merubah dan menghapus data ongkos kirim.	ongkir.php
Pengolahan Data Member	Digunakan untuk melihat, dan	member.php

	menghapus data member.	
Pengolahan Data Jenis Pengiriman	Digunakan untuk melihat, dan menghapus data jenis pengiriman	jenispengiriman.php
Pengolahan Data Rekening	Digunakan untuk melihat, dan menghapus data rekening.	rekening.php
Pengolahan Data Transaksi	Digunakan untuk melihat data transaksi yang masuk dan untuk merubah status	datatransaksi.php
Pengolahan Laporan: - Laporan retur -	Digunakan untuk membuat laporan barang yang di retur	laporanretur.php
Keluar	Digunakan admin untuk keluar dari menu admin	logout.php

Tabel 4.4 Implementasi antarmuka pengunjung dan *member*

Menu	Deskripsi	Nama file
Registrasi *member*	Digunakan untuk melakukan pendaftaran *member* baru.	register.php
Masuk *member*	Digunakan sebagai halaman login *member*	left.php
Ubah Akun: - Ubah Biodata - Ganti kata sandi	 Digunakan untuk mengubah profil *member*. Digunakan jika *member* ingin merubah kata sandi	 edit.account.php edit.account.php
Kategori	Digunakan untuk melihat barang perkategori	kategori.php
Produk : -Produk terbaru -Katalog Produk terlaris	 Digunakan untuk melihat seluruh barang terbaru. Digunakan untuk	barang.php

-Produk paling banyak dilihat	melihat seluruh barang terlaris Digunakan untuk melihat seluruh barang paling banyak dilihat	
Bantuan : Cara pembelian Data Kontak Cara pembayaran	 Digunakan untuk melihat cara melakukan pembelian Digunakan untuk melihat data kontak Digunakan untuk melihat cara pembayaran	 carapembelian.php kontak.php carapembayaran.php
Konfirmasi Pembayaran	Digunakan untuk melakukan konfirmasi pembayaran	konfirmasi.php
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus detail transaksi yang akan dilakukan.	cart.php
Keluar	Digunakan *member* untuk keluar dari akunnya.	logout.php

4.4 Pengujian

Pengujian akan dilakukan sebelum *website e-commerce* akan mulai beroperasi. Tahap pengujian akan terdiri dari 3 tahap, yaitu pengujian *black box*, pengujian alpha, dan pengujian betha. *Black box testing* merupakan pengujian pertama yang dilakukan sebelum berinteraksi dengan salah satu pengguna untuk memastikan setiap *form* berfungsi dengan baik. Disusul dengan pengujian alpha yang dimana pengembang akan menguji *website e-commerce* dengan memandang dari sisi penggguna dan merekam semua kesalahan dan masalah pemakaian. Pengujian terakhir yaitu pengujian betha dilakukan tanpa kehadiran pengembang. Pengguna akan menguji dan melaporkan pada pengembang pada interval waktu tertentu. Pengujian ini berupa penyebaran kuesioner kepada pengguna seperti pengujung atau *member* dan melakukan wawancara kepada admin.

4.5 Pengujian Black Box

Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak. Dapat dilihat padatabel 4.5.

Tabel 4.5 Rencana Pengujian Aplikasi Admin

Item Uji	Detail Pengujian	Jenis Pengujian
Masuk	Input Login Admin	*Black Box*
	Verifikasi Login Admin	*Black Box*
Pengolahan data kategori	Tambah Kategori	*Black Box*
	Ubah Kategori	*Black Box*
	Hapus Kategori	*Black Box*
	Pencarian	*Black Box*
Pengolahan data barang	Tambah Barang	*Black Box*
	Ubah Barang	*Black Box*
	Hapus Barang	*Black Box*
	Pencarian	*Black Box*
Pengolahan data barang detail	Tambah Barang detail	*Black Box*
	Ubah Barang detail	*Black Box*
	Hapus Barang detail	*Black Box*
	Pencarian	*Black Box*
Pengolahan data ukuran	Tambah ukuran	*Black Box*
	Ubah ukuran	*Black Box*
	Hapus ukuran	*Black Box*
	Pencarian	*Black Box*
	Tambah ukuran	*Black Box*
Pengolahan data warna	Tambah warna	*Black Box*
	Ubah warna	*Black Box*
	Hapus warna	*Black Box*
	Pencarian	*Black Box*
	Tambah warna	*Black Box*
Pengolahan data barang detail	Tambah barang detail	*Black Box*
	Ubah barang detail	*Black Box*
	Hapus barang detail	*Black Box*
	Pencarian	*Black Box*
Pengolahan data provinsi	Tambah barang provinsi	*Black Box*
	Ubah barang provinsi	*Black Box*
	Hapus barang provinsi	*Black Box*
	Pencarian	*Black Box*
Pengolahan data kota	Tambah barang kota	*Black Box*
	Ubah barang kota	*Black Box*
	Hapus barang kota	*Black Box*
	Pencarian	*Black Box*

Pengolahan data kota	Tambah forwarder	*Black Box*
	Ubah forwarder	*Black Box*
	Hapus forwarder	*Black Box*
	Pencarian forwader	*Black Box*
Pengolahan data jenis pengiriman	Tambah jenis pengiriman	*Black Box*
	Ubah jenis pengiriman	*Black Box*
	Hapus jenis pengiriman	*Black Box*
	Pencarian	*Black Box*
Pengolahan data ongkir	Tambah ongkir	*Black Box*
	Ubah ongkir	*Black Box*
	Hapus ongkir	*Black Box*
	Pencarian	*Black Box*
Pengolahan data rekening	Tambah rekening	*Black Box*
	Ubah rekening	*Black Box*
	Hapus rekening	*Black Box*
	Pencarian	*Black Box*

Tabel 4.6 Skenario pengujian aplikasi *member*

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Registrasi	Input Registrasi	*Black Box*
	Simpan Data	*Black Box*
	Verifikasi Email	*Black Box*
Masuk	Input Login	*Black Box*
	Verifikasi Login	*Black Box*
Keluar	Menghapus *session user*	*Black Box*
Ubah Biodata	Ubah profile	*Black Box*
Ganti kata sandi	Menampilkan *form* ubah kata sandi	*Black Box*
	Verifikasi password lama	*Black Box*
	Verifikasi password baru	*Black Box*
	Verifikasi konfirmasi password	*Black Box*
Kategori	Menampilkan data kategori	*Black Box*
	Menampilkan data detail kategori	*Black Box*
Produk	Menampilkan data barang	*Black Box*
	Menampilkan data detai l barang	*Black Box*
Menu keranjang belanja	Menampilkan data detail transaksi	*Black Box*
	Mengubah data detail Transaksi	*Black Box*

	Menghapus data detail transaksi	*Black Box*
Menu Bantuan	Menampilkan cara pemesanan dan cara pembelian	*Black Box*
Form Konfirmasi Pembayaran	Menampilkan form Konfirmasi Pembayaran	*Black Box*
	Isi form pengiriman	*Black Box*

4.6 Pengujian Alpha

Hasil implementasi perangkat lunak sistem informasi ini diuji pada pemilik toko maupun berbasis *online* melalui spesifikasi perangkat sistem agar sistem dapat berjalan dengan baik.

Pengujian perangkat lunak sistem penerapan *e-commerce* menggunakan data uji berdasarkan form yang terdapat didalam aplikasi.

1. Pengujian Pendaftaran *Member*

Berikut dapat dilihat pengujian dari pendaftaran*member* pada tabel 4.7.

Tabel 4.7 Pengujian Pendaftaran *Member*

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama : Ridwanto Purba Password : ******* Konfirmasi Password: ******* Email : kaki_bagus@yahoo.com Alamat : Rancaekek Provinsi : Jawa Barat Kota/kab : Bandung Kodepos : 40394 No Telpon : 085295306416 Masukan Kode : 10108334	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftaran. Akan muncul pesan “Terima Kasih Anda Telah Melakukan Pendaftaran. Silahkan cek email untuk melakukan Aktifasi”	Form registrasi terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftran. . Akan muncul pesan “Terima Kasih Anda Telah Melakukan Pendaftaran. Silahkan cek email untuk melakukan Aktifasi”	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji(Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama : tidak diisi (kosong)	Tampilkan pesan kesalahan”Nama harus diisi”	Dapat menampilkan Pesan kesalahan “Nama harus diisi”	[√] Diterima [ ] Ditolak
Password : tidak diisi (kosong)	Tampilkan pesan kesalahan ”password harus diisi”	Dapat menampilkan pesan kesalahan ”password harus diisi”	[√] Diterima [ ] Ditolak
Konfirmasi password : tidak diisi (kosong)	Tampilkan pesan kesalahan ”Ulangi password harus diisi”	Dapat menampilkan pesan kesalahan ” Ulangi passwors harus diisi”	[√] Diterima [ ] Ditolak

Email : tidak diisi (kosong) /tidak valid	Tampilkan pesan kesalahan "Email harus diisi"	Dapat menampilkan pesan kesalahan "Email harus diisi"	[√] Diterima [ ] Ditolak
Alamat : tidak diisi (kosong)	Tampilkan pesan kesalahan"Alamat harus diisi"	Dapat menampilkan pesan kesalahan "Alamat harus diisi"	[√] Diterima [ ] Ditolak
Provinsi : tidak diisi (kosong)	Tampilkan pesan kesalahan "Provinsi harus diisi"	Dapat menampilkan pesan kesalahan "Provinsi harus diisi"	[√] Diterima [ ] Ditolak
kota : tidak diisi (kosong)	Tampilkan pesan kesalahan "kota harus diisi"	Dapat menampilkan Pesan kesalahan "kota harus diisi"	[√] Diterima [ ] Ditolak
Kode pos : tidak diisi (kosong)	Tampilkan pesan kesalahan " kode pos harus diisi"	Dapat menampilkan pesan kesalahan "kode pos harus diisi"	[√] Diterima [ ] Ditolak
No Telepon : tidak diisi (kosong)	Tampilkan pesan Tampilkan "data harus diisi""	Dapat menampilkan pesan kesalahan "no telepon haris diisi	[√] Diterima [ ] Ditolak

2. Pengujian Masuk *Member*

Berikut dapat dilihat pengujian masuk *member* pada tabel 4.8.

Tabel 4.8 Pengujian Masuk *Member*

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat email : kaki_bagus@yahoo.com	Mengisikan data alamat email pada field alamat email.	Dapat mengisi field alamat email dan dapat mengisi data masuk 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
Email atau password yang tidak terdaftar	Tampilkan pesan kesalahan”Maaf Username dan Password Salah!!”	Muncul Pesan ”Maaf Username dan Password Salah!!”	[√] Diterima [ ] Ditolak
Mengosongkan data email atau password	Tampilkan pesan kesalahan”Maaf Username dan Password Salah!!”	Muncul Pesan ”Maaf Username dan Password Salah!!”	[√] Diterima [ ] Ditolak

3. Pengujian Pengaturan Akun *Member*

Berikut dapat dilihat pengujian dari pengaturan akun *member* pada tabel 4.9.

Tabel 4.9 Pengujian Pengaturan Akun *Member*

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama : Ridwanto Purba Email : kaki_bagus@yahoo.com Alamat : Rancaekek Provinsi : Jawa Barat Kota/kab : Bandung Kodepos : 40394 No Telpon : 085295306416	Mengubah isi field ubah profil yang akan diubah dan dapat melanjutkan proses ubah profile	Dapat mengubah isi field ubah profil yang akan diubah dan dapat melanjutkan proses ubah profile	[√] Diterima [ ] Ditolak

Password lama: ******* Password baru:******** Konfirmasi Password: ********			
Kasus dan Hasil Uji (Data salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama : tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Email : tidak diisi (kosong) /tidak valid	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Alamat : tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Provinsi : tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Kota : tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak

Kode pos : tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
No Telepon : tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Password lama: tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Password lama: tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak
Password lama: tidak diisi (kosong)	Tampilkan pesan kesalahan"Data Tidak Boleh ada yang kosong"	Dapat menampilkan Pesan kesalahan"Data Tidak Boleh ada yang kosong"	[√] Diterima [ ] Ditolak

4. Pengujian Menu Keranjang Belanja

Berikut dapat dilihat pengujian menu keranjang belanja pada tabel 4.10.

Tabel 4.10 Pengujian Menu Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
gambar produk : Sepatu (yang diinginkan)	Masuk ke detail Produk	Masuk ke detail Produk	[√] Diterima [ ] Ditolak

Warna: hitam (yangdiinginkan)	Dapat memilih warna	Dapat memilih warna	[√] Diterima [ ] Ditolak
Ukuran : 36 (yang diinginkan)	Dapat memilih ukuran	Dapat memilih ukuran	[√] Diterima [ ] Ditolak
Jumlah produk : 5 (yang diinginkan)	Dapat mengisi jumlah produk yang akan dibeli	Dapat mengisi jumlah produk yang akan dibeli	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
gambar produk : tidak diisi (kosong)	Anda belum mengisi warna, ukuran, , ulangi kembali	Anda belum mengisi warna, ukuran, maupun jumlah, silahkan ulangi kembali	[√] Diterima [ ] Ditolak
Warna:tidak diisi (kosong)	Anda belum mengisi warna, ukuran, maupun jumlah, silahkan ulangi kembali	Anda belum mengisi warna, ukuran, maupun jumlah, silahkan ulangi kembali	[√] Diterima [ ] Ditolak
Ukuran : tidak diisi (kosong)	Anda belum mengisi warna, ukuran, maupun jumlah, silahkan ulangi kembali	Anda belum mengisi warna, ukuran, maupun jumlah, silahkan ulangi kembali	[√] Diterima [ ] Ditolak
Jumlah produk : tidak diisi (kosong)	Anda belum mengisi warna, ukuran, maupun jumlah, silahkan ulangi kembali	Anda belum mengisi warna, ukuran, maupun jumlah, silahkan ulangi kembali	[√] Diterima [ ] Ditolak

4.7 Pengujian Admin

1. Pengujian Masuk Admin

Berikut dapat dilihat pengujian dari pendaftaran member pada tabel 4.11:

Tabel 4.11 Pengujian Masuk Admin

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat email : kaki_bagus@yahoo.com	Tercantum pada text box username	Dapat mengisi field username dengan benar	[√] Diterima [ ] Ditolak
Memasukan Password : *******	Tercantum pada text box password	Dapat mengisi field password dengan benar	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Email atau password yang tidak terdaftar	Tampilkan pesan kesalahan”Maaf Account tidak ditemukan”	Muncul Pesan ”Maaf Account tidak ditemukan”	[√] Diterima [ ] Ditolak
Mengosongkan data email atau password	Tampilkan pesan kesalahan”Maaf Account tidak ditemukan”	Tampilkan pesan kesalahan”Maaf Account tidak ditemukan”	[√] Diterima [ ] Ditolak

2. Pengujian Ubah Biodata Admin

Berikut ini adalah tabel pengujian Ubah biodata admin pada tabel 4.12.

Tabel 4.12 Pengujian Ubah Biodata Admin

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama : Ridwanto Purba Email : kaki_bagus@yahoo.com Telepon : 085295306416	Data berhasil diedit. Jika Mengosongkan salah satu field Menampilkan pesan kesalahan	Tampil pesan “Data berhasil disimpan”. Dapat menampilkan pesan	[√] Diterima [ ] Ditolak

Alamat : Rancaekek	bahwa field tersebut harus terisi.	kesalahan bahwa field tersebut harus terisi.	
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan ada yang tidak diisi	Tampilkan pesan kesalahan"Maaf Account tidak ditemukan"	Muncul Pesan "Maaf Account tidak ditemukan"	[√] Diterima [ ] Ditolak
Email : tidak diisi (kosong)	Muncul pesan kesalahan email harus diisi	Muncul pesan kesalahan email harus diisi	[√] Diterima [ ] Ditolak
Nama : tidak diisi (kosong)	Muncul pesan kesalahan nama harus diisi	Muncul pesan kesalahan nama harus diisi"	[√] Diterima [ ] Ditolak
Alamat : tidak diisi (kosong)	Muncul pesan kesalahan alamat harus diisi	Muncul pesan kesalahan alamat harus diisi	[√] Diterima [ ] Ditolak

3. Pengujian Pengolahan Kategori

Berikut dapat dilihat pengujian pengelolaan kategori pada tabel 4.13.

Tabel 4.13 Pengujian Pengolahan Kategori

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Kategori : Sepatu wanita	Mengisi field nama kategori yang kemudian akan ditambahkan dan tersimpan didalam database dan muncul pesan " data sedang disimpan"	Dapat menambah data kategori yang kemudian akan tersimpan didalam database dan muncul pesan " data sedang disimpan"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan

Nama Kategori : tidak diisi (kosong)	Menampilkan pesan ”nama kategori harus diisi”	Menampilkan pesan ”nama kategori harus diisi”	[√] Diterima [ ] Ditolak
Nama Kategori : sudah ada sebelumnya	Menamplikan pesan “ Maaf, kategori tidak boleh sama”	Menamplikan pesan “ Maaf, kategori tidak boleh sama”	[√] Diterima [ ] Ditolak

4. **Pengujian Pengolahan Barang**

Berikut dapat dilihat pengujian dari data barang pada tabel 4.14.

Tabel 4.14 Pengujian Pengolahan Data Barang

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama kategori: sepatu wanita Nama Barang: jeany Harga : 350.000 Diskon : 0 % Deskrispsi: sendal yang sangat trendi dan sangat nyaman dipaki oleh kaum hawa Gambar :	Jika semua field terisi semua maka akan muncul pesan “ data sedang disimpan	Jika semua field terisi semua maka akan muncul pesan “ data sedang disimpan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Kategori : tidak diisi (kosong)	Tampilkan pesan kesalahan “nama harus diisi”	Dapat menampilkan pesan kesalahan” nama harus diisi”	[√] Diterima [ ] Ditolak
Nama Barang : tidak diisi (kosong)	Muncul pesan kesalahan“nama barang harus diisi”	Dapat menampilkan pesan kesalahan “nama barang harus diisi”	[√] Diterima [ ] Ditolak
Harga : tidak diisi (kosong)	Muncul pesan kesalahan“harga harus diisi”	Dapat menampilkan pesan kesalahan “nama barang	[√] Diterima [ ] Ditolak

		harus diisi”	
Diskon : tidak diisi (kosong)	Muncul pesan kesalahan“diskon harus diisi”	Muncul pesan kesalahan“diskon harus diisi”	[√] Diterima [ ] Ditolak
Deskripsi : tidak diisi (kosong)	Muncul pesan kesalahan “deskripsi wajib diisi”	Muncul pesan kesalahan “deskripsi wajib diisi”	[√] Diterima [ ] Ditolak
Gambar : tidak ada berkas yang dipilih	Muncul pesan kesalahan “ wajib memeilih berkas gambar”	Muncul pesan kesalahan “ wajib memeilih berkas gambar”	[√] Diterima [ ] Ditolak

5. Pengujian Pengolahan Provinsi

Berikut dapat dilihat pengujian dari tambah provinsi pada tabel 4.15.

Tabel 4.15 Pengujian Pengolahan Provinsi

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
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”.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama provinsi : tidak diisi (kosong)	Menampilkan pesan ”Provinsi harus diisi”	Menampilkan pesan ”Provinsi harus diisi”	[√] Diterima [ ] Ditolak
Nama provinsi : sudah ada sebelumnya	Menamplikan pesan “ Maaf,provinsi tidak boleh sama”	Menamplikan pesan “ Maaf, provinsi tidak boleh sama”	[√] Diterima [ ] Ditolak

6. Pengujian Pengolahan Kota

Berikut dapat dilihat pengujian dari data kota pada tabel 4.16.

Tabel 4.16 Pengujian Pengolahan Kota

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama provinsi : Jawa Barat Nama Kota : Bandung Jenis : Kota	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.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama provinsi : tidak dipilih (kosong)	Menampilkan pesan ”Provinsi harus diisi”	Menampilkan pesan ”Provinsi harus diisi”	[√] Diterima [ ] Ditolak
Nama Kota : Tidak diisi (kosong)	Menamplikan pesan “ kota harus diisi”	Menamplikan pesan “ kota harus diisi”	[√] Diterima [ ] Ditolak
Jenis : tidak dipilih (kosong)	Menamplikan pesan “ Maaf,jenisharus diisi”	Menamplikan pesan “ Maaf, jenis harus diisi”	[√] Diterima [ ] Ditolak

7. Pengujian Pengolahan Ukuran

Berikut dapat dilihat pengujian dari tambah kota pada tabel 4.17.

Tabel 4.17 Pengujian Pengolahan Ukuran

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama ukuran : 38 Deskripsi ukuran : Small	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul	Dapat memunculkan pesan kesalahan bahwa field	[√] Diterima [ ] Ditolak

	pesan kesalahan bahwa field tersebut harus terisi.	tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama ukuran : tidak dipilih (kosong)	Menampilkan pesan ”Ukuran harus diisi”	Menampilkan pesan ”Ukuran harus diisi”	[√] Diterima [ ] Ditolak
Deskripsi ukuran : Tidak diisi (kosong)	Menamplikan pesan “ deskripsi harus diisi”	Menamplikan pesan “ deskripsi harus diisi”	[√] Diterima [ ] Ditolak

8. Pengujian Pengolahan Ongkos Kirim

Berikut dapat dilihat pengujian dari ongkos kirim pada tabel 4.18.

Tabel 4.18 Pengujian Pengolahan Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Provinsi : Jawa Barat Nama Kota : Bandung Jenis :JNE- Regular Harga: Rp.150	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah ongkos kirim	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah onkos kirim	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama provinsi : tidak dipilih (kosong)	Menampilkan pesan ”Provinsi harus diisi”	Menampilkan pesan ”Provinsi harus diisi”	[√] Diterima [ ] Ditolak
Nama Kota: tidak dipilih	Menamplikan pesan “ kota harus diisi”	Menamplikan pesan “ kota harus diisi”	[√] Diterima [ ] Ditolak

Jenis : tidak dipilih	Menampilkan pesan ”jenis harus diisi”	Menampilkan pesan ”jenis harus diisi”	[√] Diterima [ ] Ditolak
harga: Tidak diisi (kosong)	Menamplikan pesan “ harga harus diisi”	Menamplikan pesan “ harga harus diisi”	[√] Diterima [ ] Ditolak

9. Pengujian Pengolahan Data forwarder

Berikut dapat dilihat pengujian pengolahan data Jasa Pengiriman pada tabel 4.19.

Tabel 4.19 Pengujian Pengolahan Data Forwarder

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama forwarder: JNE	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Jasa Pengiriman: tidak diisi (kosong)	Menampilkan pesan ”jenis forwarder harus diisi”	Menampilkan pesan ”jenis forwarder harus diisi”	[√] Diterima [ ] Ditolak
Nama Jasa Pengiriman: diisi dengan data yang sudah ada	Menamplikan pesan “ jenis forwarder sudah ada”	Menamplikan pesan “ jenis forwarder sudah ada”	[√] Diterima [ ] Ditolak

10. Pengujian Pengolahan Jenis Pengiriman

Berikut dapat dilihat pengujian pengolahan jenis pengiriman pada tabel 4.20.

Tabel 4.20 Pengujian Pengolahan Jenis Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Jasa Pengiriman: JNE Nama Jenis Pengiriman:	Jika semua field terisi atau benar dalam pengisian,	Jika semua field terisi atau benar	[√] Diterima [ ] Ditolak

Reguler Deskripsi : Waktu pengiriman diperkirakan max 6 hari	maka akan dapat melanjutkan proses	dalam pengisian, maka akan dapat melanjutkan proses	
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Jasa Pengiriman: tidak dipilih	Menampilkan pesan ”jenis Jasa Pengiriman harus dipilih”	Menampilkan pesan ”jenis Jasa Pengiriman harus dipilih”	[√] Diterima [ ] Ditolak
Nama Jenis Pengiriman: tidak diisi (kosong)	Menamplikan pesan “ nama jenis pengiriman harus diisi”	Menamplikan pesan “ nama jenis pengiriman harus diisi”	[√] Diterima [ ] Ditolak
Deskripsi: tidak diisi (kosong)	Menamplikan pesan “ nama Jasa Pengiriman sudah ada”	Menamplikan pesan “ nama Jasa Pengiriman sudah ada”	[√] Diterima [ ] Ditolak

11. Pengujian Pengolahan Warna

Berikut dapat dilihat pengujian pengolahan warna pada tabel 4.21.

Tabel 4.21 Pengujian Pengolahan Warna

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Warna: Hitam Format Warna :	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan

Nama Warna: tidak diisi (kosong)	Menampilkan pesan ”warna harus diisi”	Menampilkan pesan ”warna harus diisi”	[√] Diterima [ ] Ditolak
Format warna: tidak diisi (kosong)	Menamplikan pesan “Format warna harus diisi”	Menamplikan pesan “Format warna harus diisi”	[√] Diterima [ ] Ditolak

12. Pengujian Pengolahan Data Rekening

Berikut dapat dilihat pengujian dari data rekening pada tabel 4.22.

Tabel 4.22 Pengujian Rekening

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Bank : Mandiri Nama Pemilik: Budi Aditia Koeswana Nomor Rekening : 2332391039 Cabang : Bandung	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama Bank: tidak diisi (kosong)	Menampilkan pesan ”bank harus diisi”	Menampilkan pesan ”bank harus diisi”	[√] Diterima [ ] Ditolak
Nama Pemilik: tidak diisi (kosong)	Menamplikan pesan “Nama pemilik harus diisi”	Menamplikan pesan “Nama pemilik harus diisi”	[√] Diterima [ ] Ditolak
No rekening: tidak diisi (kosong)	Menampilkan pesan ”No rekening harus diisi”	Menampilkan pesan ”No rekening harus diisi””	[√] Diterima [ ] Ditolak
Cabang: tidak diisi (kosong)	Menamplikan pesan “Cabangharus diisi”	Menamplikan pesan “Cabang harus diisi”	[√] Diterima [ ] Ditolak

4.8 Pengujian Beta

Pengujian beta ini dilakukan dengan dua cara, yaitu melakukan wawancara langsung di tempat penelitian, dan menyebar kuesioner di kalangan yang sudah terbiasa dengan pemakaian internet. Kuesioner akan disebar berdasarkan 2 tipe narasumber yaitu yang berperan sebagai pengunjung dan yang berperan sebagai *member*. *Sample* yang berupa kuisioner disebarkan untuk pengunjung atau *member* yang masing-masing berjumlah 20 orang. Data *sample* ini akan diuji dengan menggunakan *Skala Likert*(*Skala Likert* adalah suatu skala psikometrik yang umum digunakan dalam kuesioner, dan merupakan skala yang paling banyak digunakan dalam riset berupa survei). Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru.

Berdasarkan data hasil kuesioner tersebut, dicari prosentase masing-masing jawaban dengan menggunakan rumus :

Y = P/Q * 100%(1)

Keterangan : P = Banyaknya jawaban responden tiap soal

Q = Jumlah Responden

Y = Nilai Prosentase

4.9 Hasil Kuesioner

Kuesioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 20 pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuesioner ini terdiri dari 6 pertanyaan (contoh kuesioner dapat diliihat pada lampiran) dengan menggunakan skala likert dengan skala 1 sampai 6, dengan ketentuan skala untuk tiap pertanyaan sebagai berikut :

1. Apakah anda setuju bahwa website ini mudah dioperasikan dan menarik?

Tabel 4.23 Kuesioner Pertanyaan No.1

No.	Keterangan	Responden	Prosentase
1	Sangat setuju	5	25%
2	Setuju	11	55%
3	Cukup setuju		
4	Biasa Saja	3	15%
5	Tidak setuju	1	5%

Berdasarkan hasil prosentase, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 5 orang atau 25% mengatakan sangat setuju , 11 orang atau 55% mengatakan setuju, 3 orang atau 15% mengatakan biasa aja, dan 1 orang atau 5% menyatakan Tidak setuju. Sehingga secara keseluruhan *website* tersebut sangat mudah dioperasikan dan sangat menarik .

2. Apakah anda setuju bahwa website ini lebih memudahkan anda dalam melakukan transaksi penjualan barang?

Tabel 4.24 Kuesioner Pertanyaan No.2

No.	Keterangan	Responden	Prosentase
1	Sangat setuju	4	20%
2	Setuju	12	60%
3	Cukup setuju	3	15%
4	Biasa Saja	1	5%
5	Tidak setuju		

Berdasarkan hasil prosentase, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 4 orang atau 20% mengatakan sangat setuju bahwa website ini mudah untuk malakukan transaksi penjualan barang , 12 orang atau 60% mengatakan bahwa setuju, 3 orang atau 15% cukup setuju, dan 1 orang atau 5% mengatakan Biasa aja. Sehingga secara keseluruhan, bahasa yang digunakan pada website sangat mudah

3. Apakah anda setuju bahwa sistem pembayaran online yang digunakan lebih memudahkan dalam proses transaksi?

Tabel 4.25 Kuesioner Pertanyaan No.3

No.	Keterangan	Responden	Prosentase
1	Sangat sangat setuju	2	10%
2	Setuju	11	55%
3	Cukup setuju	6	30%
4	Biasa Saja	1	5%
5	Tidak setuju		

Berdasarkan hasil prosentase, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 2 orang atau 10% mengatakan bahwa sistem pembayaran online yang digunakan lebih memudahkan pengunjung , sementara 11 orang atau 55% mengatakan bahwa sistem pembayran online yang digunakan lebih memudahkan pengunjung, 6 orang atau 30% mengatakan cukup setuju, dan 1 orang atau 5% mengatakan biasa saja. Sehingga secara keseluruhan sistem pembayran online yang digunakan lebih memudahkan member pada *website* tersebut .

4. Apakah anda setuju bahwa informasi produk yang diberikan oleh website DUA SAUDARA ini sudah sesuai dengan informasi yang anda butuhkan?

Tabel 4.26 Kuesioner Pertanyaan No.4

No.	Keterangan	Responden	Prosentase
1	Sangat setuju	5	25%
2	Setuju	6	30%
3	Cukup setuju	6	30%
4	Biasa Saja	3	15%
5	Tidak setuju		

Berdasarkan hasil prosentase, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 5 orang atau 25% mengatakan bahwa informasi produk yang diberikan telah sesuai dengan informasi yang dibutuhkan, sementara 6 orang atau 30% mengatakan setuju bahwa informasi yang diberikan telah sesuai, 6 orang atau 30% mengatakan

cukup setuju bahwa informasi website yang diberikan oleh website ini telah sesuai, dan 3 orang atau 15% menyatakan biasa saja bahwa informasi yang diberikan sessuai Sehingga secara keseluruhan bahwa informasi produk yang diberikan oleh website DUA SAUDARA ini sudah sesuai dengan informasi yang dibutuhkan.

5. Apakah anda setuju sistem pembayaran yang digunakan website DUA SAUDARA dapat memudahkan anda dalam melakukan transaksi ?

Tabel 4.27 Kuesioner Pertanyaan No.5

No.	Keterangan	Responden	Prosentase
1	Sangat setuju	2	10%
2	Setuju	11	55%
3	Cukup setuju	5	25%
4	Biasa Saja	2	10%
5	Tidak setuju		

Berdasarkan hasil prosentase, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 2 orang atau 10% mengatakan sangat setuju bahwa sistem pembayaran yang digunakan sangat mudah, sementara 11 orang atau 55% mengatakan setuju bahwa sistem pembayaran yang digunakan sangat mudah dan dimengerti,5 orang atau 25% mengatakan cukup setuju , dan 2 orang atau 10% mengatakan biasa saja. Sehingga secara keseluruhan bahwa sistem pembayaran yang dimiliki pada *website* DUA SAUDARA ini sangat mudah untuk dimengerti.

6. Apakah anda setuju bahwa website DUA SAUDARA ini telah layak di onlinekan?

Tabel 4.28 Kuesioner Pertanyaan No.6

No.	Keterangan	Responden	Prosentase
1	Sangat setuju	4	20%
2	Setuju	13	65%
3	Cukup setuju	2	10%
4	Biasa Saja	1	5%
5	Tidak setuju		

Berdasarkan hasil prosentase, telah dihitung jawaban kuesioner dengan menggunakan rumus 1 maka dapat disimpulkan bahwa sebanyak 4 orang atau 20% mengatakan sangat setuju bahwa DUA SAUDARA ini telah layak di onlinekan, sementara 13 orang atau 65% mengatakan setuju bahwa website DUA SAUDARA ini telah layak di onlinekan, 2 orang atau 10% mengatakan cukup setuju bahwa website DUA SAUDARA ini telah layak di onlinekan, dan 1 orang atau 5% mengatakan biasa saja. Sehingga secara keseluruhan bahwa website DUA SAUDARA ini telah layak di onlinekan.

4.10 Hasil Wawancara

Wawancara dilakukan kepada pemilik (sebagai admin) yang telah mencoba sistem dan melakukan proses-proses yang ada dalam sistem. Wawancara dilakukan dengan mangajukan 5 pertanyaan, hasil dari wawancara dapat dilihat sebagai berikut :

1. Apakah dengan adanya *website e-commerce* ini dapat memudahkan dalam melakukan promosi produk di Toko Dua Saudara?

   Jawaban : Ya, *website* ini mampu membantu mempromosikan produk yang dijual yang ada di Toko Dua Saudara.
2. Apakah *website* ini mempermudah anda dalam mengelola data produk?

   Jawaban : Ya, *website* ini mampu membantu dalam mengelola data produk yang ada di toko
3. Apakah *website* ini dapat memudahkan dalam mengelola data transaksi di toko Dua Saudara?

   Jawaban: Ya, karena proses transaksi lebih mudah karena terdapat sistem pembayaran *online* dan *offline*
4. Apakah aplikasi yang dibangun dapat memudahkan dalam membuat laporan penjualan produk?

   Jawaban : Ya, laporan lebih mudah dibuat sehingga mudah untuk dicek dan mengurangi kesalahan yang terjadi

4.11 Kesimpulan dari Hasil Pengujian Beta

Berdasarkan pengujian beta diatas, bahwa dapat diambil kesimpulan sebagai berikut.

1. *Website* ini mudah dioperasikan dan menarik
2. *Website* ini membantu dalam melakukan pembelian.
3. Informasi produk cukup sesuai dengan informasi yang dibutuhkan oleh pengguna.
4. Sistem pembayaran pada *website* penjualan Toko DUA SAUDARA mempermudah pembeli..
5. Mempermudah Toko DUA SAUDARA dalam mengolah data produk.
6. Mempermudah pengolahan data transaksi.