BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penciptaan perangkat lunak, tahap kelanjutan dari kegiatan perancangan sistem. Tahap ini merupakan menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya.

4.1.1 Perangkat keras yang digunakan

Spesifikasi perangkat keras (*hardware*) yang digunakan dalam pembangunan aplikasi *E-Commerce* di toko kamera DH FOTO adalah tercantum pada tabel 4.1 sebagai berikut:

Tabel 4. 1 Perangkat keras yang digunakan

Perangkat Keras	Spesifikasi
Processor	Dual Core T42000 @2.20GHZ
RAM	2.00 Gb
Harddisk	160 Gb
Monitor	LCD 12”

4.1.2 Perangkat Lunak yang digunakan

Spesifikasi perangkat lunak (*Software*) yang dipasang pada sistem komputer yang digunakan untuk mengembangkan sistem penerapan *E-commerce* untuk sistem penjualan ini adalah tercantum pada tabel 4.2 sebagai berikut :

Tabel 4. 2 Perangkat lunak yang digunakan

Perangkat Lunak	Keterangan
Sistem Operasi	Windows XP profesional
PHP	Bahasa Pemrograman
MySQL	Penyimpanan *database*
PhpMyAdmin	*interface database*
XAMPP	*web server*
Macromedia Dreamweaver 8	Kode dan perancangan tampilan
Mozilla Firefox	*Web browser*

4.2 Implementasi Basis Data

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

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,
  `type_admin` enum('superadmin','admin') NOT NULL,
  PRIMARY KEY (`id_admin`),
```

```
  UNIQUE KEY `email_admin` (`email_admin`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 COMMENT='Tabel
Administrator';
```

2. Tabel Barang

```
CREATE TABLE IF NOT EXISTS `barang` (
  `id_barang` int(11) NOT NULL AUTO_INCREMENT,
  `id_detailmerk` int(11) NOT NULL,
  `nama_barang` varchar(50) NOT NULL,
  `harga_barang` int(11) NOT NULL,
  `deskripsi_barang` text NOT NULL,
  `kondisi_barang` text NOT NULL,
  `berat_barang` float NOT NULL,
  `stok_barang` int(11) NOT NULL,
  `rating_barang` float NOT NULL,
  `voterrating_barang` int(11) NOT NULL,
  `viewcounter_barang` int(11) NOT NULL,
  `type_barang` enum('normal','auction') NOT NULL,
  PRIMARY KEY (`id_barang`),
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 COMMENT='Tabel
Barang' AUTO_INCREMENT=16 ;
ALTER TABLE `barang`
  ADD   CONSTRAINT   `barang_ibfk_1`   FOREIGN   KEY
(`id_detailmerk`)       REFERENCES       `detailmerk`
(`id_detailmerk`);
```

3. Tabel Bid

```
CREATE TABLE IF NOT EXISTS `bid` (
  `id_bid` int(11) NOT NULL AUTO_INCREMENT,
  `id_lelang` int(11) NOT NULL,
  `id_member` int(11) NOT NULL,
```

```
  `harga_bid` float NOT NULL,
  `waktu_bid` datetime NOT NULL,
  PRIMARY KEY (`id_bid`),
  KEY `id_lelang` (`id_lelang`),
  KEY `id_member` (`id_member`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Bid';
ALTER TABLE `bid`
  ADD CONSTRAINT `bid_ibfk_2` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`),
  ADD CONSTRAINT `bid_ibfk_1` FOREIGN KEY (`id_lelang`)
REFERENCES `lelang` (`id_lelang`);
```

4. Tabel Currency

```
CREATE TABLE IF NOT EXISTS `currency` (
  `id_currency` int(11) NOT NULL AUTO_INCREMENT,
  `kode_currency` varchar(5) NOT NULL,
  `symbol_currency` varchar(5) NOT NULL,
  `harga_currency` int(11) NOT NULL,
  `deskripsi_currency` text NOT NULL,
  `status_currency` enum('0','1') NOT NULL,
  PRIMARY KEY (`id_currency`),
  UNIQUE KEY `nama_currency`
(`kode_currency`,`symbol_currency`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Kurs';
```

5. Tabel Detail Lelang

```
CREATE TABLE IF NOT EXISTS `detaillelang` (
  `id_detaillelang` int(11) NOT NULL AUTO_INCREMENT,
  `id_lelang` int(11) NOT NULL,
```

```
  `id_member` int(11) NOT NULL,
  `waktu_detaillelang` datetime NOT NULL,
  `status_pengiriman`
enum('dipesan','dikonfirmasi','dibayar','dikirim','dite
rima') NOT NULL,
  `keterangan_detaillelang` text NOT NULL,
  `nama_pemesan` varchar(50) NOT NULL,
  `alamat_pemesan` text NOT NULL,
  `email_pemesan` varchar(50) NOT NULL,
  `no_telp_pemesan` varchar(20) NOT NULL,
  `kodepos_pemesan` varchar(6) NOT NULL,
  `kota_pemesan` varchar(100) NOT NULL,
  `provinsi_pemesan` varchar(100) NOT NULL,
  `no_resi_pemesan` varchar(50) NOT NULL,
  `hargafix_detaillelang` float NOT NULL,
  `id_ongkoskirim` int(11) NOT NULL,
  `hargaongkir_detaillelang` float NOT NULL,
  `status_pemesanan` enum('ok','cancel') NOT NULL,
  PRIMARY KEY (`id_detaillelang`),
  KEY `id_lelang` (`id_lelang`),
  KEY `id_member` (`id_member`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
detailpembelian';
ALTER TABLE `detaillelang`
  ADD  CONSTRAINT  `detaillelang_ibfk_2`  FOREIGN  KEY
(`id_member`) REFERENCES `detaillelang` (`id_member`),
  ADD  CONSTRAINT  `detaillelang_ibfk_1`  FOREIGN  KEY
(`id_lelang`) REFERENCES `lelang` (`id_lelang`);
```

6. Tabel Detail Merk

```
CREATE TABLE IF NOT EXISTS `detailmerk` (
  `id_detailmerk` int(11) NOT NULL AUTO_INCREMENT,
  `id_kategori` int(11) NOT NULL,
  `id_merk` int(11) NOT NULL,
  PRIMARY KEY (`id_detailmerk`),
  KEY `id_merk` (`id_merk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Detail Merk';
ALTER TABLE `detailmerk`
  ADD  CONSTRAINT  `detailmerk_ibfk_2`  FOREIGN  KEY
(`id_kategori`) REFERENCES `kategori` (`id_kategori`),
  ADD  CONSTRAINT  `detailmerk_ibfk_1`  FOREIGN  KEY
(`id_merk`) REFERENCES `merk` (`id_merk`);
```

7. Tabel Detail Pembelian

```
CREATE TABLE IF NOT EXISTS `detailpembelian` (
  `id_detailpembelian` int(11) NOT NULL AUTO_INCREMENT,
  `tanggal_detailpembelian` datetime NOT NULL,
  `status_pengiriman`
enum('dipesan','dikonfirmasi','dibayar','dikirim','dite
rima') NOT NULL,
  `keterangan_detailpembelian` text NOT NULL,
  `nama_pemesan` varchar(50) NOT NULL,
  `alamat_pemesan` text NOT NULL,
  `email_pemesan` varchar(50) NOT NULL,
  `no_telp_pemesan` varchar(20) NOT NULL,
  `kodepos_pemesan` varchar(6) NOT NULL,
  `kota_pemesan` varchar(100) NOT NULL,
  `provinsi_pemesan` varchar(100) NOT NULL,
```

```
  `no_resi_pemesan` varchar(50) NOT NULL,
  `totalpembayaran_detailpembelian` float NOT NULL,
  `hargaongkir_detailpembelian` float NOT NULL,
  `status_pemesanan` enum('ok','cancel') NOT NULL,
  `session_id` varchar(32) NOT NULL,
  `id_currency` int(11) NOT NULL,
  PRIMARY KEY (`id_detailpembelian`),
  KEY `id_currency` (`id_currency`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
detailpembelian';
ALTER TABLE `detailpembelian`
  ADD CONSTRAINT `detailpembelian_ibfk_1` FOREIGN KEY
(`id_currency`)      REFERENCES      `detailpembelian`
(`id_currency`);
```

8. Tabel Gambar

```
CREATE TABLE IF NOT EXISTS `gambar` (
  `id_gambar` int(11) NOT NULL AUTO_INCREMENT,
  `id_barang` int(11) NOT NULL,
  `path_gambar` varchar(200) NOT NULL,
  `profile_gambar` enum('0','1') NOT NULL,
  PRIMARY KEY (`id_gambar`),
  KEY `id_barang` (`id_barang`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Gambar';
ALTER TABLE `gambar`
  ADD    CONSTRAINT    `gambar_ibfk_1`    FOREIGN    KEY
(`id_barang`) REFERENCES `gambar` (`id_barang`);
```

9. Tabel Hubungi

```
CREATE TABLE IF NOT EXISTS `hubungi` (
  `id_hubungi` int(11) NOT NULL AUTO_INCREMENT,
```

```
  `id_admin` int(11) DEFAULT NULL,
  `nama_hubungi` varchar(100) NOT NULL,
  `email_hubungi` varchar(50) NOT NULL,
  `telp_hubungi` varchar(20) NOT NULL,
  `isi_hubungi` text NOT NULL,
  `tanggal_hubungi` datetime NOT NULL,
  PRIMARY KEY (`id_hubungi`),
  KEY `id_admin` (`id_admin`),
  KEY `id_admin_2` (`id_admin`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Hubungi Kami';
ALTER TABLE `hubungi`
  ADD   CONSTRAINT   `hubungi_ibfk_1`   FOREIGN   KEY
(`id_admin`) REFERENCES `hubungi` (`id_admin`);
```

10. Tabel Jasa Pengiriman

```
CREATE TABLE IF NOT EXISTS `jasapengiriman` (
  `id_jasapengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `nama_jasapengiriman` varchar(100) NOT NULL,
  `deskripsi_jasapengiriman` text NOT NULL,
  PRIMARY KEY (`id_jasapengiriman`),
  UNIQUE          KEY          `nama_jasapengiriman`
(`nama_jasapengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

11. Tabel Jenis Pengiriman

```
CREATE TABLE IF NOT EXISTS `jenispengiriman` (
  `id_jenispengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_jasapengiriman` int(11) NOT NULL,
  `nama_jenispengiriman` varchar(50) NOT NULL,
  `deskripsi_jenispengiriman` text NOT NULL,
```

```
  PRIMARY KEY (`id_jenispengiriman`),
  KEY `id_jasapengiriman` (`id_jasapengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Jenis Pengiriman';
ALTER TABLE `jenispengiriman`
  ADD CONSTRAINT `jenispengiriman_ibfk_1` FOREIGN KEY
(`id_jasapengiriman`)  REFERENCES  `jenispengiriman`
(`id_jasapengiriman`);
```

12. Tabel Kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL AUTO_INCREMENT,
  `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';
```

13. Tabel Konfirmasi lelang

```
CREATE TABLE IF NOT EXISTS `konfirmasilelang` (
  `id_konfirmasilelang`     int(11)     NOT     NULL
AUTO_INCREMENT,
  `id_detaillelang` int(11) NOT NULL,
  `payment_gateway` enum('TRANSFER','PAYPAL') NOT NULL,
  `bank_rekening` varchar(50) DEFAULT NULL,
  `no_rekening` varchar(50) DEFAULT NULL,
  `instansi_pembayaran` varchar(25) NOT NULL,
  `no_transaksi` varchar(25) NOT NULL,
  `totalbayar_konfirmasilelang` float NOT NULL,
  `kurs_konfirmasilelang` float DEFAULT NULL,
```

```
  `tanggal_konfirmasilelang` datetime NOT NULL,
  PRIMARY KEY (`id_konfirmasilelang`),
  KEY `id_detaillelang` (`id_detaillelang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Tabel
Konfirmasi';
ALTER TABLE `konfirmasilelang`
  ADD CONSTRAINT `konfirmasilelang_ibfk_1` FOREIGN KEY
(`id_detaillelang`)      REFERENCES      `detaillelang`
(`id_detaillelang`);
```

14. Tabel Konfirmasi Pembayaran

```
CREATE TABLE IF NOT EXISTS `konfirmasipembayaran` (
  `id_konfirmasipembayaran`     int(11)     NOT     NULL
AUTO_INCREMENT,
  `id_detailpembelian` int(11) NOT NULL,
  `payment_gateway` enum('TRANSFER','PAYPAL') NOT NULL,
  `bank_rekening` varchar(50) DEFAULT NULL,
  `no_rekening` varchar(50) DEFAULT NULL,
  `instansi_pembayaran` varchar(25) NOT NULL,
  `no_transaksi` varchar(25) NOT NULL,
  `totalbayar_konfirmasipembayaran` float NOT NULL,
  `kurs_konfirmasipembayaran` float DEFAULT NULL,
  `tanggal_konfirmasipembayaran` datetime NOT NULL,
  PRIMARY KEY (`id_konfirmasipembayaran`),
  KEY `id_detailpembelian` (`id_detailpembelian`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Konfirmasi';
ALTER TABLE `konfirmasipembayaran`
  ADD CONSTRAINT `konfirmasipembayaran_ibfk_1` FOREIGN
KEY (`id_detailpembelian`) REFERENCES `detailpembelian`
(`id_detailpembelian`);
```

15. Tabel Kota

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(10) NOT NULL AUTO_INCREMENT,
  `id_provinsi` int(10) DEFAULT NULL,
  `nama_kota` varchar(50) DEFAULT NULL,
  `kabkota` varchar(20) DEFAULT NULL,
  UNIQUE KEY `kota#PX` (`id_kota`),
  KEY `id_provinsi` (`id_provinsi`),
  KEY `id_provinsi_2` (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `kota`
  ADD    CONSTRAINT    `kota_ibfk_1`    FOREIGN    KEY
(`id_provinsi`) REFERENCES `provinsi` (`id_provinsi`);
```

16. Tabel Lelang

```
CREATE TABLE IF NOT EXISTS `lelang` (
  `id_lelang` int(11) NOT NULL AUTO_INCREMENT,
  `id_barang` int(11) NOT NULL,
  `nama_lelang` varchar(100) NOT NULL,
  `countbid_lelang` int(11) NOT NULL,
  `currentbid_lelang` float NOT NULL,
  `minbid_lelang` float NOT NULL,
  `starttime_lelang` datetime NOT NULL,
  `endtime_lelang` datetime NOT NULL,
  PRIMARY KEY (`id_lelang`),
  KEY `id_barang` (`id_barang`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Lelang';
ALTER TABLE `lelang`
  ADD    CONSTRAINT    `lelang_ibfk_1`    FOREIGN    KEY
(`id_barang`) REFERENCES `barang` (`id_barang`);
```

17. Tabel 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','2') 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';
ALTER TABLE `member`
  ADD   CONSTRAINT   `member_ibfk_1`   FOREIGN   KEY
(`id_kota`) REFERENCES `kota` (`id_kota`);
```

18. Tabel Merk

```
CREATE TABLE IF NOT EXISTS `merk` (
  `id_merk` int(11) NOT NULL AUTO_INCREMENT,
  `nama_merk` varchar(50) NOT NULL,
  PRIMARY KEY (`id_merk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Merk';
```

19. Tabel Ongkos Kirim

```
CREATE TABLE IF NOT EXISTS `ongkoskirim` (
  `id_ongkoskirim` int(11) NOT NULL AUTO_INCREMENT,
  `id_kota` int(11) NOT NULL,
  `id_jenispengiriman` int(11) NOT NULL,
  `harga_ongkoskirim` int(11) NOT NULL,
  PRIMARY KEY (`id_ongkoskirim`),
  KEY `id_kota` (`id_kota`),
  KEY `id_jenispengiriman` (`id_jenispengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Ongkos Kirim';
ALTER TABLE `ongkoskirim`
  ADD  CONSTRAINT  `ongkoskirim_ibfk_2`  FOREIGN  KEY
(`id_jenispengiriman`)    REFERENCES    `ongkoskirim`
(`id_jenispengiriman`),
  ADD  CONSTRAINT  `ongkoskirim_ibfk_1`  FOREIGN  KEY
(`id_kota`) REFERENCES `kota` (`id_kota`);
```

20. Tabel 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_barang` int(11) NOT NULL,
  `harga_temp` int(11) NOT NULL,
  `stok_temp` int(11) NOT NULL,
  `berat_temp` float NOT NULL,
  `session_id` varchar(32) NOT NULL,
  `id_admin` int(11) NOT NULL,
  PRIMARY KEY (`id_pembelian`),
  KEY `id_detailpembelian` (`id_detailpembelian`),
```

```
  KEY `id_member` (`id_member`),
  KEY `id_barang` (`id_barang`),
  KEY `id_admin` (`id_admin`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Pembelian';
ALTER TABLE `pembelian`
  ADD  CONSTRAINT  `pembelian_ibfk_4`  FOREIGN  KEY
(`id_admin`) REFERENCES `pembelian` (`id_admin`),
  ADD  CONSTRAINT  `pembelian_ibfk_1`  FOREIGN  KEY
(`id_detailpembelian`)  REFERENCES  `detailpembelian`
(`id_detailpembelian`),
  ADD  CONSTRAINT  `pembelian_ibfk_2`  FOREIGN  KEY
(`id_member`) REFERENCES `member` (`id_member`),
  ADD  CONSTRAINT  `pembelian_ibfk_3`  FOREIGN  KEY
(`id_barang`) REFERENCES `barang` (`id_barang`);
```

21. Tabel Provinsi

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(10) NOT NULL AUTO_INCREMENT,
  `nama_provinsi` varchar(30) DEFAULT NULL,
  KEY `id_prov` (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

22. Tabel 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`),
  KEY `id_admin` (`id_admin`),
  KEY `id_admin_2` (`id_admin`),
  KEY `id_admin_3` (`id_admin`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 COMMENT='Tabel
Rekening';
ALTER TABLE `rekening`
  ADD   CONSTRAINT   `rekening_ibfk_1`   FOREIGN   KEY
(`id_admin`) REFERENCES `rekening` (`id_admin`);
```

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 pembangunan aplikasi *E-commerce* di toko kamera DH FOTO berbasis *Web* akan dilampirkan pada halaman lampiran.

Tabel 4. 3 Implementasi antarmuka Pengunjung dan Member

Menu	Deskripsi	Nama File
Registarsi member	Digunakan untuk melakukan pendaftaran *member* baru.	Register.html
Login *member*	Digunakan Digunakan sebagai halaman login *member*	Account.html
Lupa *password*	Digunakan jika *member* lupa dengan *password*	forgotpassword.html

Cara Pembelian dan Lelang	Digunakan untuk melihat cara pembelian dan lelang	Carapembelian.html
Ubah Akun	Digunakan untuk mengubah profil *member.*	account.php
Ganti password	Digunakan jika *member* ingin merubah *password*	ubahpass.php
Kategori	Digunakan untuk melihat barang perkategori	kategori. Php
Lelang	Digunakan untuk melihat barang yang di lelang	Lelang.html
Hubungi	Digunakan untuk melakukan tanya jawab secara langsung	hubungi.html
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus data pembelian yang akan dilakukan.	cart.html
Member History	Digunakan untuk melihat *history* pembelian yang pernah dilakukan.	history.php
Logout	Digunakan *member* untuk keluar dari akunnya.	logout.html

Tabel 4. 4 Implementasi Antarmuka Admin

Menu	Deskripsi	Nama File
Login administrator	Digunakan sebagai halaman login administrator	index.php
Lupa Password	Digunakan jika admin lupa terhadap passwordnya	Forgotpassword.php
Ubah Biodata Admin	Digunakan jika admin ingin merubah profil	Editprofile.php
Pengolahan Data Admin	Digunakan untuk melihat,menambah dan menghapus admin	Admin.Php
Ubah Kata Sandi Admin	Digunakan jika administrator ingin merubah *password*	editpassword.Php
Kota	Digunakan untuk melihat, menambah, menghapus dan merubah data kota	Kota.Php
Provinsi	Digunakan untuk melihat, menambah, menghapus dan merubah data provinsi	provinsi.Php
Kategori	Digunakan untuk melihat, menambah, menghapus dan merubah data kategori	kategori.Php
Barang	Digunakan untuk melihat, menambah, menghapus dan merubah data barang	barang.Php
Jasa Pengiriman	Digunakan untuk melihat, menambah, menghapus dan	Jasapengiriman.Php

	merubah data Jasa Pengiriman	
Jenis Pengiriman	Digunakan untuk melihat, menambah, menghapus dan merubah data Jenis Pengiriman	Jenispengiriman.Php
Member	Digunakan untuk melihat, menghapus, dan mengaktifkan status member	Member.Php
Rekening	Digunakan untuk melihat, menambah, menghapus dan merubah data rekening	Rekening.Php
Merk	Digunakan untuk melihat, menambah, menghapus dan merubah data Merk	Merk.Php
Detail Merk	Digunakan untuk melihat, menambah, menghapus dan merubah data Detail Merk	Detailmerk.Php
Kurs	Digunakan untuk melihat, menambah, menghapus, merubah dan mengaktifkan status data kurs	Currency.Php
Lelang	Digunakan untuk melihat, menambah, menghapus dan merubah data Lelang	Lelang. Php
Data Transaksi	Digunakan untuk melihat data transaksi yang masuk dan merubah statusnya.	datatransaksi. Php
Konfirmasi Pembayaran	Digunakan untuk melihat data konfirmasi pembayaran	Konfirmasipembayaran.Php
Data Lelang	Digunakan untuk melihat data lelang	

Konfirmasi lelang	Digunakan untuk melihat data konfirmasi pembayaran lelang	Konfirmasilelang.Php
Ongkos Kirim	Digunakan untuk melihat data onkos kirim, merubah, menghapus onkos kirim	ongkoskirim. Php
Laporan	Digunakan untuk melihat data laporan harian,bulanan maupun tahunan kemudian dicetak ke pdf	laporan.php
Hubungi	Digunakan untuk melihat data hubungi yang akan dibaca.	hubungi.php
Logout	Digunakan administrator Untuk dapat keluar dari aplikasi	logout.php

4.2 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan – kesalahan atau kekurangan – kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

Pengujian perangkat lunak ini menggunakan pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

4.3 Pengujian Alpa

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

4.5.1 Skenario Pengujian Alpa

Tabel 4.5 Skenario pengujian aplikasi *member*

Kelas Uji	Butir Uji	Jenis Pengujian
Registrasi *member*	Menampilkan *form* user baru	*Black Box*
	Isi *form* pendaftaran	*Black Box*
	Simpan data baru	*Black Box*
Login	Isi data login	*Black Box*
	Verifikasi *Email*	*Black Box*
	Verifikasi *password*	*Black Box*
Ubah profil	Menampilkan *form* ubah profil	*Black Box*
	Ubah data *member*	*Black Box*
	Simpan perubahan	*Black Box*
ubah *password*	Menampilkan *form* ubah *password*	*Black Box*
	Verifikasi *password* lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verifikasi konfirmasi *password*	*Black Box*
Menu Produk	Menampilkan data barang	*Black Box*
	Menampilkan data barang	*Black Box*
	Cari barang	*Black Box*
Menu Profil Toko	Menampilkan profil perusahaan	*Black Box*
keranjang belanja	Menampilkan data pembelian	*Black Box*
	Mengupdate data pembelian	*Black Box*
	Menghapus data pembelian	*Black Box*
Form konfirmasi	Menampilkan konfirmasi transaksi	*Black Box*
Form pengiriman	Menampilkan form pengiriman	*Black Box*
	Pilih jenis pengiriman	*Black Box*
Logout	Menghapus *session user*	*Black Box*

Tabel 4.6 Menjelaskan Skenario pengujian aplikasi administrator.

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Isi data login	*Black Box*
	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Lupa *password*	Menampilkan *form* Lupa *password*	*Black Box*
Data admin	Tampilkan data admin	*Black Box*
	Tambah data admin	*Black Box*
Kategori	Tampilkan data kategori	*Black Box*
	Tambah Data katgori	*Black Box*
	Ubah data kategori	*Black Box*
Provinsi	Tampilkan data Provinsi	*Black Box*
	Tambah data Provinsi	*Black Box*
	Ubah data Provinsi	*Black Box*
Kota	Tampilkan data kota	*Black Box*
	Tambah data kota	*Black Box*
	Ubah data kota	*Black Box*
Lelang	Tampilkan data Lelang	*Black Box*
	Tambah data Lelang	*Black Box*
	Ubah data ukuran	*Black Box*
Data Laporan	Tampilkan data laporan	*Black Box*
	Cari data laporan	*Black Box*
	Cetak data laporan	*Black Box*
Data Transaksi	Tampilkan data Transaksi	*Black Box*
	Ubah data transaksi	*Black Box*
Member	Tampilan Data Member	*Black Box*
Data kategori	Tampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
Detail Barang	Tampilkan detail barang	*Black Box*

	Tambah detail barang	*Black Box*
	Ubah detail barang	*Black Box*
Data Barang	Tampilkan data barang	*Black Box*
	Tambah data barang	*Black Box*
	Ubah data barang	*Black Box*
Logout	Menghapus *session user*	*Black Box*

4.5.2 Kasus dan Hasil Pengujian

4.5.2.1 Pengujian Registrasi Member

Berikut dapat dilihat pengujian dari registrasi *member* yang terdapat pada tabel 4.7 di halaman berikut :

Tabel 4.7. Pengujian Registrasi *member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data pendaftaran, contoh masukan : Nama : Hario Alamat Email : if.10106115@yahoo.ci.id Alamat : Jl. Kamper no. 28 Telepon : 081320335957 Provinsi : Jawa Barat Kota : Bandung Kodepos : 40623 Password : yamaha Konfirmasi password : yamaha	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses registrasi.	Form registrasi di disi semua maka sesuai yang di harapkan	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data kosong	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.2 Pengujian Login Member

Berikut dapat dilihat pengujian login *member* pada tabel 4.8 dihalaman berikut:

Tabel 4.8 Pengujian login *member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat *email* : if.10106115@yahoo.co.id	Mengisikan data alamat *email* pada *field* alamat *email*.	Dapat mengisi *field* alamat *email* dan dapat mengisi data login dengan benar	[√] Diterima [ ] Ditolak
Memasukan *Password* : yamaha	Mengisikan data *password* pada *field password*. Jika benar akan masuk ke	Mengisikan data *password* pada *field password*. Jika benar akan masuk ke menu utama	[√] Diterima [ ] Ditolak

	menu utama		
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data *email* atau *password* yang tidak terdaftar	Tidak dapat login	Muncul pesan "Maaf account tidak di temukan "	[√] Diterima [ ] Ditolak
mengosongkan data *email* atau *password*	Tidak dapat login	Dapat menampilkan pesan kesalahan " Maaf account tidak di temukan"	[√] Diterima [ ] Ditolak

4.5.2.3 Pengujian Ubah Profile Member

Berikut dapat dilihat pengujian dari ubah profil member pada tabel 4.9. dibawah ini:

Tabel 4.9 Pengujian ubah profil member

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih ubah profil member	Menampilkan *form* ubah profil	Dapat menampilkan *form* ubah profil	[√] Diterima [ ] Ditolak
Data profil yang akan diubah	Tercantum pada text box data yang akan diubah	Dapat mengisi *field* data yang akan diubah dengan benar	[√] Diterima [ ] Ditolak
Pilih tombol update	Data berhasil di update di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			

Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan ada yang tidak diisi	Menampilkan pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan bahwa field tersebut harus terisi.	[√] Diterima [ ] Ditolak
Data masukan semuanya tidak diisi	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak
Data kosong	Muncul pesan kesalahan field harus diisi	Dapat menampilkan pesan kesalahan field harus diisi	[√] Diterima [ ] Ditolak

4.5.2.4 Pengujian ganti *password member*

Berikut dapat dilihat pengujian dari ganti *password member* pada tabel 4.10 dibawah ini:**Tabel 4.10 Pengujian ganti password member**

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih ubah *password*	Menampilkan *form* ubah *password*	Dapat menampilkan *form* ubah *password*	[√] Diterima [ ] Ditolak
data yang dimasukkan semuanya benar	*Password* baru berhasil disimpan di *database*	Dapat menyimpan data *password* yang baru dengan muncul pesan “ Data berhasil diubah”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu field belum terisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi” bila ada field yang	[√] Diterima [ ] Ditolak

		belum diisi	
Data *password* lama salah	muncul pesan "password salah"	Dapat menampilkan pesan kesalahan ”*Password* salah”	[√] Diterima [ ] Ditolak
Data *password* baru dan konfirmasi *password* baru tidak sama	Muncul pesan "password baru dan konfirmasi password harus sama"	Dapat menampilkan pesan "password baru dan konfirmasi password harus sama"	[√] Diterima [ ] Ditolak
Data kosong	Muncul pesan kesalahan field harus diisi	Dapat menampilkan pesan kesalahan field harus diisi	[√] Diterima [ ] Ditolak

4.5.2.5 Pengujian Menu Kategori

Berikut dapat dilihat pengujian dari menu kategori pada tabel 4.11 dibawah ini:

Tabel 4.11 Pengujian Menu kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu kategori	Menampilkan data kategori yang ada di *database*	Dapat menampilkan data kategori yang ada di *data base*	[√] Diterima [ ] Ditolak
Pilih beli pada data produk	Menampilkan detail produk	Dapat menampilkan detail produk	[√] Diterima [ ] Ditolak
Pilih beli	Menuju ke keranjang belanja	Dapat menampilkan keranjang belanja	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih merk	Menampilkan pilih merk	Pesan “ukuran harus disi”	[√] Diterima [ ] Ditolak

4.5.2.6 Pengujian Menu Lelang

Berikut dapat dilihat pengujian dari menu kategori pada tabel 4.12 dibawah ini:

Tabel 4.12 Pengujian Menu Lelang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu lelang	Menampilkan data kategori yang ada di *database*	Dapat menampilkan data kategori yang ada di*data base*	[√] Diterima [ ] Ditolak
Pilih barang lelang	Menampilkan detail lelang	Dapat menampilkan detail lelang	[√] Diterima [ ] Ditolak
Isi bid	Data bid telah tersimpan di database	Bid telah terisi	[√] Diterima [ ] Ditolak

4.5.2.7 Pengujian *Form CheckOut*

Berikut dapat dilihat pengujian dari *Form Checkout* pada tabel 4.13 dibawah ini:

Tabel 4.13 Pengujian *Form CheckOut*

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

4.5.2.8 Pengujian Form Pengiriman

Berikut dapat dilihat pengujian dari Form pengiriman pada tabel 4.14 dibawah ini:

Tabel 4.14 Pengujian form Pengiriman

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

Data masukan benar lalu mengPilih tombol ganti	Data pengiriman baru akan disimpan di dalam *database*	Dapat data pengirimah baru akan disimpan di dalam *database*	[√] Diterima [ ] Ditolak
Salah satu *field* kosong	Menampilkan pesan kesalahan field tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak
Pilih tombol batal	Kembali ke menu sebelumnya	Dapat kembali ke menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.9 Pengujian Menu History Pembelian

Berikut dapat dilihat pengujian dari menu history pembayaran pada tabel 4.15 dibawah ini:

Tabel 4.15 Pengujian Menu *History* Pembelian

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu *history* Pembelian	Menampilkan *history*	Dapat menampilkan form *history*	[√] Diterima [ ] Ditolak
Pilih view	Menampilkan detail transaksi	Dapat menampilkan detail transaksi	[√] Diterima [ ] Ditolak

4.5.2.10 Pengujian Menu History Lelang

Berikut dapat dilihat pengujian dari menu history lelang pada tabel 4.16 dibawah ini:

Tabel 4.16 Pengujian Menu *History* Lelang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih menu *history* Pembelian	Menampilkan *history*	Dapat menampilkan form *history*	[√] Diterima [ ] Ditolak
Pilih view	Menampilkan detail transaksi	Dapat menampilkan detail transaksi	[√] Diterima [ ] Ditolak

4.5.2.11 Pengujian Form Konfirmasi Pembayaran

Berikut dapat dilihat pengujian dari *Form* konfirmasi pembayaran pada tabel 4.17 dibawah ini:

Tabel 4.17 Pengujian Konfirmasi Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh No. invoice : 2 Nama Bank : Mandiri Atas Nama : kharisma Jumlah Transfer : 5100	Menampilkan *form* konfirmasi	Dapat memunculkan pesan form konfirmasi.	[√] Diterima [ ] Ditolak
Memasukan data kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.12 Pengujian Form Konfirmasi Pembayaran Lelang

Berikut dapat dilihat pengujian dari *Form* konfirmasi pembayaran pada tabel 4.18 dibawah ini:

Tabel 4.18 Pengujian Konfirmasi Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh No. invoice lelang : 4 Nama Bank : BNI Atas Nama : Rudi Jumlah Transfer : 1700000	Menampilkan *form* konfirmasi	Dapat memunculkan pesan form konfirmasi.	[√] Diterima [ ] Ditolak
Memasukan data kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.13 Pengujian Login Administrator

Berikut dapat dilihat pengujian dari login administrator pada tabel 4.19 dibawah ini:

Tabel 4.19 Pengujian login Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
E-mail : admin@dhfotostore.com	Tercantum pada text box *username*	Dapat mengisi field *username* dengan benar	[√] Diterima [ ] Ditolak
Password : admin123	Tercantum pada text box *password*	Dapat mengisi field *password* dengan benar	[√] Diterima [ ] Ditolak

Pilih tombol login	jika sama dengan data login maka admin masuk ke menu utama admin	Tombol login dapat berfungsi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *email* atau *password* tidak terdaftar	Tidak dapat login	Muncul pesan "maaf account tidak ditemukan."	[√] Diterima [ ] Ditolak
Data *emaikl* atau *password* kosong	Tidak dapat login	Muncul pesan kesalahan "account tidak ditemukan."	[√] Diterima [ ] Ditolak

4.5.2.14 Pengujian Ganti Password Admin

Berikut dapat dilihat pengujian dari ganti *password* administrator pada tabel 4.20 dibawah ini:

Tabel 4.20 Pengujian ganti *password* administrator

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

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

4.5.2.15 Pengujian Ubah Profil Administrator

Berikut dapat dilihat pengujian dari ubah profil administrator pada tabel 4.21 dibawah ini:

Tabel 4.21 Pengujian Ubah Profil Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih edit profil Administrator	Menampilkan *form* ubah profil	Dapat menampilkan *form* ubah profil	[√] Diterima [ ] Ditolak
Data profil yang akan diubah	Tercantum pada text box data yang akan diubah	Dapat mengisi *field* data yang akan diubah dengan benar	[√] Diterima [ ] Ditolak
Pilih ubah	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan ada	Menampilkan	Dapat	[√] Diterima

yang tidak diisi	pesan kesalahan bahwa field tersebut harus terisi.	menampilkan pesan kesalahan bahwa field tersebut harus terisi.	[ ] Ditolak
Data masukan semuanya tidak diisi	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.16 Pengujian Pengolahan Data Kategori

Berikut dapat dilihat pengujian dari pengolahan data kategori pada tabel 4.22 dibawah ini:

Tabel 4.22 Pengujian lihat kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih kategori	Menampilkan *form* lihat kategori	Dapat menampilkan *form* lihat kategori	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah kategori	Dapat menampilkan *form* ubah kategori	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan	[√] Diterima [ ] Ditolak

		pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	

4.5.2.17 Pengujian Tambah Kategori

Berikut dapat dilihat pengujian dari tambah kategori pada tabel 4.23 dibawah ini :

Tabel 4.23 Pengujian Tambah kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Tambah	Menampilkan *form* tambah kategori	Dapat menampilkan *form* tambah kategori	[√] Diterima [ ] Ditolak
Mengisi data kategori Contoh: Nama kategori: Feminim	Jika semua field terisi maka akan muncul pesan “data berhasil ditambah”	Dapat memunculkan pesan data berhasil ditambah	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data kategori masih kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.20 Pengujian Pengolahan Data Barang

Berikut dapat dilihat pengujian dari pengolahan data Produk pada tabel 4.24 dibawah ini :

Tabel 4.24 Pengujian Lihat Barang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Barang	Menampilkan data Barang	Dapat menampilkan data Barang	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah Barang	Dapat menampilkan *form* ubah Barang	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Data produk kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "ada kesalahan didalam pengisian data, silahkan cek kembali"	[√] Diterima [ ] Ditolak

4.5.2.21 Pengujian Tambah Data Barang

Berikut dapat dilihat pengujian dari pengolahan data barang pada tabel 4.25 dibawah ini :

Tabel 4.25 Pengujian Tambah Barang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Tambah	Menampilkan *form* tambah barang	Dapat menampilkan *form* tambah barang	[√] Diterima [ ] Ditolak
Mengisi data barang Contoh: kategori: Feminim Nama : Fem C 01 Harga : 120000 Deskripsi: Gambar tampak depan: Gambar tampak samping: Gambar tampak belakang:	Jika semua field terisi semua maka akan muncul pesan " data berhasil ditambah"	Dapat memunculkan pesan data berhasil ditambah	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data barang akan tersimpan didalam *database*	Data barang akan tersimpan didalam *database* dan muncul pesan "data berhasil di tambah"	[√] Diterima [ ] Ditolak
Pilih tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukkan	Yang diharapkan	Pengamatan	Kesimpulan
Data barang masih kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "ada kesalahan didalam pengisian data, silahkan cek kembali"	[√] Diterima [ ] Ditolak

4.5.2.22 Pengujian Pengolahan Data Provinsi

Berikut dapat dilihat pengujian dari pengolahan data provinsi pada tabel 4.26 dibawah ini:

Tabel 4.26 Pengujian Lihat provinsi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih provinsi	Menampilkan data provinsi	Dapat menampilkan data provinsi	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah provinsi	Dapat menampilkan *form* ubah provinsi	[√] Diterima [ ] Ditolak
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Data provinsi kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.23 Pengujian Tambah Provinsi

Berikut dapat dilihat pengujian dari tambah provinsi pada tabel 4.27 dibawah ini:

Tabel 4.27 Pengujian Tambah Provinsi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih tambah provinsi	Menampilkan *form* tambah provinsi	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data provinsi Contoh: Nama Provinsi : Jawa Barat	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
Data provinsi masih kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.24 Pengujian Pengolahan Data Kota

Berikut dapat dilihat pengujian dari pengolahan data kota pada tabel 4.28 dibawah ini:

Tabel 4.28 Pengujian Lihat Kota

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih kota	Menampilkan data kota	Dapat menampilkan data kota	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah kota	Dapat menampilkan *form* ubah kota	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Data kota kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "ada kesalahan didalam pengisian data, silahkan cek kembali"	[√] Diterima [ ] Ditolak

4.5.2.25 Pengujian Tambah Kota

Berikut dapat dilihat pengujian dari tambah kota pada tabel 4.29 dibawah ini:

Tabel 4.29 Pengujian Tambah kota

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih baru	Menampilkan *form* tambah kota	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota : Cikarang	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan 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
Pilih tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data kota masih kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.26 Pengujian Pengolahan Data Merk

Berikut dapat dilihat pengujian dari pengolahan data merk pada tabel 4.30 dibawah ini:

Tabel 4.30 Pengujian Pengolahan Data Merk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih merk	Menampilkan data merk	Dapat menampilkan data merk	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah merk	Dapat menampilkan *form* ubah merk	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "ada kesalahan didalam pengisian data, silahkan cek kembali"	[√] Diterima [ ] Ditolak

4.5.2.27 Pengujian Tambah Merk

Berikut dapat dilihat pengujian dari tambah merk pada tabel 4.31 dibawah ini:

Tabel 4.31 Pengujian Tambah Merk

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih tambah merk	Menampilkan *form* tambah merk	Dapat menampilkan *form* tambah merk	[√] Diterima [ ] Ditolak
Mengisi data merk Contoh: Nama merk : Nikon	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
Pilih tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.28 Pengujian Data Member

Berikut dapat dilihat pengujian dari pengolahan data member pada tabel 4.32 dibawah ini:

Tabel 4.32 Pengujian Lihat Member

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih member	Menampilkan data member	Dapat menampilkan data member	[√] Diterima [ ] Ditolak
Pilih *check box* pada data yang akan dihapus hapus	Memilih data yang akan dihapus	Data yang akan dihapus telah dipilih	[√] Diterima [ ] Ditolak
Pilih hapus	Data akan dihapus dari *database*. Jika tidak ada data yang dipilih akan muncul pesan kesalahan	Data telah dihapus dari database. Muncul pesan "pilih data yang akan dihapus" bila data belum dipilih	[√] Diterima [ ] Ditolak

4.5.2.29 Pengujian Pengolahan Data Detail Merk

Berikut dapat dilihat pengujian dari pengolahan data detail merk pada tabel 4.33 dibawah ini:

Tabel 4.33 Pengujian Pengolahan Data Detail Merk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih detail merk	Menampilkan data detail merk	Dapat menampilkan data detail merk	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah merk	Dapat menampilkan *form* ubah merk	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak

Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.30 Pengujian Tambah detail Merk

Berikut dapat dilihat pengujian dari tambah detail merk pada tabel 4.34 dibawah ini:

Tabel 4.34 Pengujian Tambah Detail Merk

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih tambah merk	Menampilkan *form* tambah detail merk	Dapat menampilkan *form* tambah detail merk	[√] Diterima [ ] Ditolak
Mengisi data detail merk Contoh: Kategori: Lensa Merk : Nikon	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi	[√] Diterima [ ] Ditolak

	pesan kesalahan bahwa field tersebut harus terisi.	atau salah dalam pengisian.	
Pilih tombol simpan	Data akan tersimpan didalam *database*	Data akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.31 Pengujian Pengolahan Data Rekening

Berikut dapat dilihat pengujian dari pengolahan data rekening pada tabel 4.35 dibawah ini:

Tabel 4.35 Pengujian Pengolahan Data Rekening

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih rekening	Menampilkan data rekening	Dapat menampilkan data rekening	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah rekening	Dapat menampilkan *form* ubah rekening	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil	Tampil pesan “Data	[√] Diterima

	diubah di*database*	berhasil diubah"	[ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "ada kesalahan didalam pengisian data, silahkan cek kembali"	[√] Diterima [ ] Ditolak

4.5.2.32 Pengujian Tambah Rekening

Berikut dapat dilihat pengujian dari tambah rekening pada tabel 4.36 dibawah ini:

Tabel 4.36 Pengujian Tambah Rekening

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih tambah rekening	Menampilkan *form* tambah rekening	Dapat menampilkan *form* tambah rekening	[√] Diterima [ ] Ditolak
Mengisi data rekening Contoh: Nama bank : Bni Nama Pemilik : hario Rekening : 12345678 Cabang : bandung Gambar : gambar	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam	[√] Diterima [ ] Ditolak

	bahwa field tersebut harus terisi.	pengisian.	
Pilih tombol simpan	Data akan tersimpan didalam *database*	Data akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.33 Pengujian Pengolahan Data Lelang

Berikut dapat dilihat pengujian dari pengolahan data master lelang pada tabel 4.37 dibawah ini:

Tabel 4.37 Pengujian Pengolahan Data Master Lelang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih lelang	Menampilkan data lelang	Dapat menampilkan data lelang	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah lelang	Dapat menampilkan *form* ubah lelang	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.34 Pengujian Tambah Data Master Lelang

Berikut dapat dilihat pengujian dari tambah data master lelang pada tabel 4.38 dibawah ini:

Tabel 4.38 Pengujian Tambah Data Master lelang

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih tambah lelang	Menampilkan *form* tambah lelang	Dapat menampilkan *form* tambah lelang	[√] Diterima [ ] Ditolak
Mengisi data lelang Contoh: Barang : kamera canon eos 20 D Nama lelang : lelang kamera Harga start lelang : 850000 Waktu akhir : 23 mei 2011 07:13:45 Waktu awal : : 23 mei 2011 23:13:45	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 Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih tombol simpan	Data akan tersimpan didalam *database*	Data akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.35 Pengujian Pengolahan Data Lelang

Berikut dapat dilihat pengujian dari lihat transaksi lelang pada tabel 4.39 dibawah ini:

Tabel 4.39 Pengujian Pengolahan Data Lelang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih pengolahan data lelang	Menampilkan *form* data lelang	Dapat menampilkan *form* lihat data lelang	[√] Diterima [ ] Ditolak
Pilih lihat pengolahan data lelang	Menampilkan *form* detail data lelang	Dapat menampilkan *form* transaksi lelang pembayaran lelang	[√] Diterima [ ] Ditolak

4.5.2.36 Pengujian Pengolahan Data Konfirmasi Lelang

Berikut dapat dilihat pengujian dari lihat data konfirmasi lelang pada tabel 4.40 dibawah ini:

Tabel 4.40 Pengujian Pengolahan Data Konfirmasi Lelang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih pengolahan data konfirmasi lelang	Menampilkan *form* konfirmasi lelang	Dapat menampilkan *form* lihat konfirmasi lelang	[√] Diterima [ ] Ditolak
Pilih lihat konfirmasi lelang	Menampilkan data konfirmasi lelang	Dapat menampilkan data konfirmasi lelang	[√] Diterima [ ] Ditolak

4.5.2.37 Pengujian Pengolahan Data Transaksi

Berikut dapat dilihat pengujian dari lihat transaksi pada tabel 4.41 dibawah ini:

Tabel 4.41 Pengujian Pengolahan Data Transaksi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih pengeolahan data transaksi	Menampilkan *form* transaksi	Dapat menampilkan *form* lihat transaksi	[√] Diterima [ ] Ditolak
Pilih lihat transaksi	Menampilkan data detail transaksi	Dapat menampilkan data detail transaksi	[√] Diterima [ ] Ditolak

4.5.2.38 Pengujian Pengolahan Data Konfirmasi pembayaran

Berikut dapat dilihat pengujian dari lihat transaksi pada tabel 4.42 dibawah ini:

Tabel 4.42 Pengujian Pengolahan Data Konfirmasi Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih pengolahan data konfirmasi pembayaran	Menampilkan *form* konfirmasi pembayaran	Dapat menampilkan *form* lihat konfirmasi pembayaran	[√] Diterima [ ] Ditolak
Pilih lihat konfirmasi pembayaran	Menampilkan data detail konfirmasi pembayaran	Dapat menampilkan data detail konfirmasi pemabayaran	[√] Diterima [ ] Ditolak

4.5.2.39 Pengujian Pengolahan Data Ongkos Kirim

Berikut dapat dilihat pengujian pengolahan data ongkos kirim pada tabel 4.43 dibawah ini:

sTabel 4.43 Pengujian Pengolahan Data ongkos kirim

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih pengolahan data ongkos kirim	Menampilkan data ongkos kirirm	Dapat menampilkan data kota ongkos kirim	[√] Diterima [ ] Ditolak
Pilih ubah	Menampilkan *form* ubah ongkos kirim	Dapat menampilkan *form*ongkos kirim	[√] Diterima [ ] Ditolak
Pilih tombol simpan	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu *field*	Muncul pesan	Dapat muncul pesan	[√] Diterima

kosong	kesalahan *field* tersebut harus diisi	kesalahan *field* tersebut harus diisi	[ ] Ditolak
Data ongkos kirim kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.2.40 Pengujian Tambah Ongkos Kirim

Berikut dapat dilihat pengujian dari tambah ongkos kirim pada tabel 4.44 dibawah ini:

Tabel 4.44 Pengujian Tambah Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih Tambah	Menampilkan *form* tambah ongkos kirim	Dapat menampilkan *form* tambah ongkos kirim	[√] Diterima [ ] Ditolak
Mengisi data ongkos kirim Contoh: Provinsi : Jawa Barat Kota : Bandung Jenis pengiriman : JNE harga: Rp.8000	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
Pilih tombol simpan	Data ongkos kirim akan tersimpan didalam *database*	Data ongkos kirim akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Pilih batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data ongkos kirim masih kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

4.5.3 Keimpulan Hasil Pengujian Alpa

Berdasarkan hasil pengujian sistem yang telah dilakukan secara keseluruhan memberikan kesimpulan bahwa pada proses pada situs *e-commerce* telah melalui tahap perbaikan dan sudah dimaksimalkan terhadap proses-proses tersebut dan secara fungsional sistem sudah dapat digunakan dan menghasilkan *output* yang diharapkan.

4.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan dengan membuat kuesioner mengenai kepuasan user dengan kandungan poin syarat user friendly untuk selanjutnya dibagikan kepada sebagian user dengan mengambil *sample* sebanyak 10 orang dan 1 orang yang nantinya sebagai admin. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru.

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

1. Lingkungan Penjual di Toko Kamera DH Foto

   Penelitian di lingkungan penjual di Toko Kamera DH Foto dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

2. Lingkungan Masyarakat Umum

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

4.6.1 Kuesioner Pengguna

Kuesioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 10 pengguna biasa dan 1 pengguna yang nantinya sebagai admin. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuesioner ini terdiri dari 5 pertanyaan (contoh kuesioner dapat diliihat pada lampiran) dengan menggunakan skala likert dengan skala 1 sampai 5, dengan ketentuan skala untuk tiap pertanyaan sebagai berikut :

4.6.1.1 Kuesioner Pengujian Beta Untuk Admin

1. Apakah anda setuju tampilan perangkat lunak untuk pengolahan data sudah bagus?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju

3	Cukup Setuju
4	Setuju
5	Sangat Setuju

2. Apakah anda setuju instruksi yang terdapat dalam *website* tersebut mudah dimengerti?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

3. Apakah anda setuju *website* ini membantu dalam pengolahan data barang?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

4. Apakah anda setuju website ini membantu dalam pengolahan data penjualan secara *online?*

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

5. Apakah anda setuju *website* ini dapat membantu dalam membuat laporan penjualan ?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju

3	Cukup Setuju
4	Setuju
5	Sangat Setuju

Berdasarkan data hasil kusioner tersebut, dapat dicari rata-rata hasil setiap jawaban dengan menggunakan rumus : Y = P/R

Keterangan :

P = Jumlah point jawaban responden tiap soal.

R = Jumlah responden

Y = Nilai rata-rata

Berikut ini adalah hasil rata-rata jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuisioner ini diujikan kepada 1 orang.

No	Responden	Pertanyaan 1					Pertanyaan 2					Pertanyaan 3					Pertanyaan 4					Pertanyaan 5				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	Ardiansyah				√					√				√						√						√	
	Jumlah Poin	4					4					3					4					4					
	Rata-rata	4/1= 4					4/1=4					3/1=3					4/1=4					4/1=4				

4.6.1.2 Kuesioner Pengujian Beta Untuk Pengguna

1. Apakah Anda setuju bahwa tampilan *website* Toko Kamera DH Foto cukup menarik dan mudah dipahami / *user friendly*?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

2. Apakah Anda setuju bahwa *website* E-commerce ini dapat membantu Anda untuk mendapatkan informasi terbaru tentang produk yang ada di Toko DH Foto?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

3. Apakah Anda setuju bahwa *website* ini akan lebih mempermudah Anda dalam melakukan transaksi pembelian di bandingkan transaksi secara manual?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

4. Apakah Anda setuju dengan adanya *website* ini pemilik dapat memperluas pangsa pasar?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

5. Apakah anda setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan anda?

No	Keterangan
1	Tidak Setuju
2	Kurang Setuju
3	Cukup Setuju
4	Setuju
5	Sangat Setuju

Berdasarkan data hasil kusioner tersebut, dapat dicari rata-rata hasil setiap jawaban dengan menggunakan rumus : Y = P/R

Keterangan :

P = Jumlah point jawaban responden tiap soal.

R = Jumlah responden

Y = Nilai rata-rata

Berikut ini adalah hasil rata-rata setiap jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuisioner ini diujikan kepada 10 orang.

No	Responden	Pertanyaan 1					Pertanyaan 2					Pertanyaan 3					Pertanyaan 4					Pertanyaan 5				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	Asti				√					√				√						√				√		
2	Arya					√				√					√						√			√		
3	Andri					√					√			√							√			√		
4	Anissa				√						√				√					√				√		
5	Farid				√						√				√					√					√	
6	Harya			√						√				√						√					√	
7	Yanto					√				√				√						√				√		
8	Eka				√						√				√						√				√	
9	Octa				√					√				√						√					√	
10	Novan					√				√					√					√					√	
	Jumlah Poin	43					44					40					43					35				
	Rata-rata	43/10 = 4.3					44/10 = 4.4					40 /10 = 4					43/10 = 4.3					35/10 = 3.5				

4.6.2 Kesimpulan Pengujian Beta

Dengan beberapa aturan pembulatan data statistika sebagai berikut :

- Angka lebih dari 5 dibulatkan jadi 10 pada tempatnya. Artinya angka yang mendahuluinya ditambah 1 pada tempatnya.
- Angka kurang dari 5 dibulatkan jadi 0 pada tmpatnya. Artinya angka yang mendahuluinya tetap.
- Angka sama dengan 5 dibulatkan jadi 0 jika angka yang mendahuluinya genap, dan dibulatkan jadi 10 jika angka yang mendahuluinya ganjil. Dengan demikian hasil pembulatan selalu genap, aturan terakhir ini disebut aturan genap terdekat.

Berikut ini adalah hasil kesimpulan pengujian beta untuk pengguna admin terhadap 1 responden dan pengguna biasa terhadap 10 responden.

Kesimpulan Pengujian untuk Admin

Pertanyaan	Poin Akhir	Kesimpulan
Pertanyaan ke 1	4	Setuju tampilan perangkat lunak untuk pengolahan data sudah bagus
Pertanyaan ke 2	4	Setuju instruksi yang terdapat dalam website tersebut mudah dimengerti
Pertanyaan ke 3	3	Cukup setuju website ini membantu dalam pengolahan data barang
Pertanyaan ke 4	4	Setuju website ini membantu dalam pengoalahan data penjualan secara online
Pertanyaan ke 5	4	Setuju website ini dapat membantu dalam membuat laporan penjualan

Kesimpulan pengujian untuk Pengguna Biasa

Pertanyaan	Poin Akhir	Kesimpulan
Pertanyaan ke 1	4	Setuju bahwa tampilan website Toko Kamera DH Foto cukup menarik dan mudah dipahami / user friendly
Pertanyaan ke 2	4	Setuju bahwa website E-commerce ini dapat membantu untuk mendapatkan informasi terbaru tentang produk yang ada di Toko Kamera DH Foto
Pertanyaan ke 3	4	Setuju bahwa website ini akan lebih mempermudah dalam melakukan transaksi pembelian di bandingkan transaksi secara manual
Pertanyaan ke 4	4	Setuju dengan adanya website ini pemilik dapat memperluas pangsa pasar
Pertanyaan ke 5	3	Cukup setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan anda