BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

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

4.1.1. Perangkat Keras yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam pembangunan aplikasi *E-Commerce* berbasis web pada Van's Computer 354 tercantum pada tabel berikut:

Tabel 4.1 Perangkat keras yang digunakan

Perangkat	Spesifikasi
Processor	Intel Atom N450 1,66 GHz
RAM	DDR2 1 GB
Harddisk	160 GB
Monitor	LCD 10" 1024x600 piksel
VGA	Intel GMA 3150

4.1.2. Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak yang digunakan dalam membangun aplikasi *E-Commerce* berbasis *web* pada Van's Computer 354 sebagai berikut:

Tabel 4.2 Perangkat lunak yang digunakan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP Professional SP3
2	Bahasa Pemrograman	PHP 5.3.5
3	*DBMS*	MySQL 5.0.7
4	*Web browser*	Google Chrome 9.0 Beta
5	*Code editor*	Notepad++ 5.9
7	*Web server localhost*	XAMPP 1.7.4 (Apache 2.2.17)

4.1.3. Implementasi Basis Data

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

1. Tabel admin

```
CREATE TABLE IF NOT EXISTS `admin` (
     `id_admin` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `username` varchar(20) NOT NULL,
     `nama_depan` varchar(20) NOT NULL,
     `nama_belakang` varchar(40) NOT NULL,
     `jk` enum('L','P') NOT NULL,
     `tanggal_lahir` date NOT NULL,
     `email` varchar(40) NOT NULL,
     `telp` varchar(12) NOT NULL,
     `alamat` varchar(80) NOT NULL,
     `password` varchar(32) NOT NULL,
     `aktif` enum('Y','T') NOT NULL DEFAULT 'Y',
     `level` enum('A','O') NOT NULL,
     PRIMARY KEY (`id_admin`),
     UNIQUE KEY `email` (`email`),
     UNIQUE KEY `username` (`username`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

2. Tabel member

```
CREATE TABLE IF NOT EXISTS `member` (
     `id_member` int(10) unsigned NOT NULL AUTO_INCREMENT,
     `nama_depan` varchar(20) NOT NULL,
     `nama_belakang` varchar(40) NOT NULL,
     `jk` enum('L','P') NOT NULL,
     `tanggal_lahir` date NOT NULL,
     `email` varchar(40) NOT NULL,
     `telp` varchar(12) DEFAULT NULL,
     `id_provinsi` int(5) unsigned NOT NULL,
     `id_kota` int(5) unsigned NOT NULL,
     `alamat` varchar(80) NOT NULL,
     `kodepos` int(5) unsigned NOT NULL,
     `password` varchar(32) NOT NULL,
     `verifikasi` varchar(32) NOT NULL,
     `aktif` enum('Y','T') NOT NULL DEFAULT 'T',
     `tgl_aktif` datetime DEFAULT NULL,
     `blokir` enum('Y','T') NOT NULL DEFAULT 'T',
     `gagal` int(2) unsigned NOT NULL DEFAULT '0',
     PRIMARY KEY (`id_member`),
     UNIQUE KEY `email` (`email`),
     CONSTRAINT `member_ibfk_1` FOREIGN KEY (`id_kota`)
     REFERENCES `kota` (`id_kota`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

3. Tabel merk

```
CREATE TABLE IF NOT EXISTS `merk` (
     `id_merk` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `nama_merk` varchar(20) NOT NULL,
     `website` varchar(40) DEFAULT NULL,
     PRIMARY KEY (`id_merk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

4. Tabel kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
     `id_kategori` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `nama_kategori` varchar(20) NOT NULL,
     PRIMARY KEY (`id_kategori`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

5. Tabel sub_kategori

```
CREATE TABLE IF NOT EXISTS `sub_kategori` (
     `id_sub_kategori` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `id_kategori` int(5) unsigned NOT NULL,
     `nama_sub_kategori` varchar(20) NOT NULL,
     PRIMARY KEY (`id_sub_kategori`),
     CONSTRAINT `sub_kategori_ibfk_1` FOREIGN KEY (`id_kategori`)
     REFERENCES `kategori` (`id_kategori`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

6. Tabel produk

```
CREATE TABLE IF NOT EXISTS `produk` (
     `id_produk` int(10) unsigned NOT NULL AUTO_INCREMENT,
     `id_merk` int(5) unsigned NOT NULL,
     `id_kategori` int(5) unsigned NOT NULL,
     `id_sub_kategori` int(5) unsigned DEFAULT NULL,
     `nama_produk` varchar(80) NOT NULL,
     `deskripsi` varchar(600) DEFAULT NULL,
     `harga` float(10,0) unsigned NOT NULL,
     `berat` float(6,0) unsigned NOT NULL,
     `stok` int(5) unsigned NOT NULL,
     `diskon` float(2,0) unsigned NOT NULL DEFAULT '0',
     `avg_rating` float(4,2) unsigned NOT NULL DEFAULT '0.00',
     `count_rating` int(10) unsigned NOT NULL DEFAULT '0',
     `sum_rating` int(10) unsigned NOT NULL DEFAULT '0',
     `count_view` int(10) unsigned NOT NULL DEFAULT '0',
     `count_buy` int(10) unsigned NOT NULL DEFAULT '0',
     `tanggal_input` datetime NOT NULL,
```

```
     `seo` varchar(200) NOT NULL,
     `aktif` enum('Y','T') NOT NULL DEFAULT 'Y',
     PRIMARY KEY (`id_produk`),
     CONSTRAINT `produk_ibfk_2` FOREIGN KEY (`id_kategori`)
     REFERENCES `kategori` (`id_kategori`)
     ON DELETE NO ACTION ON UPDATE CASCADE,
     CONSTRAINT `produk_ibfk_1` FOREIGN KEY (`id_merk`)
     REFERENCES `merk` (`id_merk`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

7. Tabel rekening

```
CREATE TABLE IF NOT EXISTS `rekening` (
     `id_rekening` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `id_admin` int(5) unsigned NOT NULL,
     `no_rekening` varchar(40) NOT NULL,
     `nama` varchar(60) NOT NULL,
     `bank` varchar(20) NOT NULL,
     `cabang` varchar(20) NOT NULL,
     `logo` varchar(180) NOT NULL,
     PRIMARY KEY (`id_rekening`),
     CONSTRAINT `rekening_ibfk_1` FOREIGN KEY (`id_admin`)
     REFERENCES `admin` (`id_admin`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

8. Tabel yahoo

```
CREATE TABLE IF NOT EXISTS `yahoo` (
     `id_ym` varchar(20) NOT NULL,
     `id_admin` int(5) unsigned DEFAULT NULL,
     `aktif` enum('Y','T') NOT NULL DEFAULT 'Y',
     PRIMARY KEY (`id_ym`),
     CONSTRAINT `yahoo_ibfk_1` FOREIGN KEY (`id_admin`)
     REFERENCES `admin` (`id_admin`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

9. Tabel bantuan

```
CREATE TABLE IF NOT EXISTS `bantuan` (
     `id_bantuan` int(10) unsigned NOT NULL AUTO_INCREMENT,
     `id_member` int(10) unsigned DEFAULT NULL,
     `id_admin` int(5) unsigned DEFAULT NULL,
     `nama` varchar(60) NOT NULL,
     `email` varchar(40) NOT NULL,
     `tanggal` datetime NOT NULL,
     `topik` varchar(180) NOT NULL,
     `isi` varchar(600) NOT NULL,
     `status` enum('Baru','Ditangani','Selesai') NOT NULL DEFAULT
   'Baru',
     KEY `id_bantuan` (`id_bantuan`),
     CONSTRAINT `bantuan_ibfk_1` FOREIGN KEY (`id_member`)
     REFERENCES `member` (`id_member`)
     ON DELETE NO ACTION ON UPDATE CASCADE,
     CONSTRAINT `bantuan_ibfk_2` FOREIGN KEY (`id_admin`)
     REFERENCES `admin` (`id_admin`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

10. Tabel gambar

```
CREATE TABLE IF NOT EXISTS `gambar` (
     `id_gambar` int(10) unsigned NOT NULL AUTO_INCREMENT,
     `id_produk` int(10) unsigned NOT NULL,
     `path` varchar(255) DEFAULT NULL,
     PRIMARY KEY (`id_gambar`),
     CONSTRAINT `gambar_ibfk_1` FOREIGN KEY (`id_produk`)
     REFERENCES `produk` (`id_produk`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

11. Tabel beli

```
CREATE TABLE IF NOT EXISTS `beli` (
     `id_beli` int(10) unsigned NOT NULL AUTO_INCREMENT,
     `id_member` int(10) unsigned NOT NULL,
     `jenis_pembayaran` enum('Rekening','ATM','E-
   Banking','Paypal') DEFAULT NULL,
     `id_rekening` int(5) unsigned DEFAULT NULL,
     `id_tarif` int(10) unsigned DEFAULT NULL,
     `no_resi` varchar(40) DEFAULT NULL,
     `nama` varchar(60) NOT NULL,
     `email` varchar(40) NOT NULL,
     `telp` varchar(12) NOT NULL,
     `alamat` varchar(80) NOT NULL,
     `kodepos` int(5) unsigned NOT NULL,
     `pengirim` varchar(60) DEFAULT NULL,
     `total_kirim` float(12,2) unsigned DEFAULT NULL,
     `tanggal_beli` datetime NOT NULL,
     `tanggal_bayar` datetime DEFAULT NULL,
     `tanggal_terima` datetime DEFAULT NULL,
     `status_kiriman`
   enum('Dipesan','Dikonfirmasi','Dibayar','Dikirim','Diterima')
   NOT NULL DEFAULT 'Dipesan',
     `status_pesanan` enum('Aktif','Batal') NOT NULL DEFAULT
   'Aktif',
     `session_id` varchar(32) NOT NULL,
     PRIMARY KEY (`id_beli`),
     CONSTRAINT `beli_ibfk_3` FOREIGN KEY (`id_tarif`)
     REFERENCES `tarif` (`id_tarif`)
     ON DELETE NO ACTION ON UPDATE CASCADE,
     CONSTRAINT `beli_ibfk_1` FOREIGN KEY (`id_member`)
     REFERENCES `member` (`id_member`)
     ON DELETE NO ACTION ON UPDATE CASCADE,
     CONSTRAINT `beli_ibfk_2` FOREIGN KEY (`id_rekening`)
     REFERENCES `rekening` (`id_rekening`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

12. Tabel beli_detail

```
CREATE TABLE IF NOT EXISTS `beli_detail` (
     `id_beli_detail` int(10) unsigned NOT NULL AUTO_INCREMENT,
     `id_beli` int(10) unsigned NOT NULL,
     `id_produk` int(10) unsigned NOT NULL,
     `harga` float(10,0) unsigned NOT NULL,
     `berat` float(6,0) unsigned NOT NULL,
     `jumlah` int(5) unsigned NOT NULL,
     PRIMARY KEY (`id_beli_detail`),
     CONSTRAINT `beli_detail_ibfk_2` FOREIGN KEY (`id_produk`)
     REFERENCES `produk` (`id_produk`)
     ON DELETE NO ACTION ON UPDATE CASCADE,
     CONSTRAINT `beli_detail_ibfk_1` FOREIGN KEY (`id_beli`)
     REFERENCES `beli` (`id_beli`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

13. Tabel provinsi

```
CREATE TABLE IF NOT EXISTS `provinsi` (
     `id_provinsi` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `nama_provinsi` varchar(60) NOT NULL,
     PRIMARY KEY (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

14. Tabel kota

```
CREATE TABLE IF NOT EXISTS `kota` (
     `id_kota` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `id_provinsi` int(5) unsigned NOT NULL,
     `nama_kota` varchar(60) NOT NULL,
     PRIMARY KEY (`id_kota`),
     CONSTRAINT `kota_ibfk_1` FOREIGN KEY (`id_provinsi`)
     REFERENCES `provinsi` (`id_provinsi`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

15. Tabel kurir

```
CREATE TABLE IF NOT EXISTS `kurir` (
     `id_kurir` int(5) unsigned NOT NULL AUTO_INCREMENT,
     `jasa` varchar(20) NOT NULL,
     `jenis` varchar(20) NOT NULL,
     `keterangan` varchar(60) DEFAULT NULL,
     PRIMARY KEY (`id_kurir`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

16. Tabel retur

```
CREATE TABLE IF NOT EXISTS `retur` (
     `id_retur` int(10) unsigned NOT NULL AUTO_INCREMENT,
     `id_beli` int(10) unsigned NOT NULL,
     `tgl_kirim` datetime DEFAULT NULL,
     `resi_kirim` varchar(40) DEFAULT NULL,
     `kurir_kirim` int(5) unsigned DEFAULT NULL,
     `tgl_retur` datetime DEFAULT NULL,
     `resi_retur` varchar(40) DEFAULT NULL,
     `kurir_retur` int(5) unsigned DEFAULT NULL,
     `tgl_terima` datetime DEFAULT NULL,
     `status`
   enum('Baru','Ditolak','Disetujui','Dikembalikan','Diterima','Di
   kirim','Selesai') NOT NULL DEFAULT 'Baru',
     UNIQUE KEY `id_retur` (`id_retur`),
     CONSTRAINT `retur_ibfk_1` FOREIGN KEY (`id_beli`)
     REFERENCES `beli` (`id_beli`)
     ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

17. Tabel retur_detail

```
CREATE TABLE IF NOT EXISTS `retur_detail` (
     `id_retur` int(10) unsigned NOT NULL,
     `id_beli_detail` int(10) unsigned NOT NULL,
     `jumlah` int(5) unsigned NOT NULL,
     `alasan` varchar(255) DEFAULT NULL,
     `setuju` enum('Y','T') DEFAULT NULL,
```

```
    `img` varchar(255) DEFAULT NULL,
    CONSTRAINT `retur_detail_ibfk_2` FOREIGN KEY
  (`id_beli_detail`) REFERENCES `beli_detail` (`id_beli_detail`)
  ON DELETE NO ACTION ON UPDATE CASCADE,
    CONSTRAINT `retur_detail_ibfk_1` FOREIGN KEY (`id_retur`)
    REFERENCES `retur` (`id_retur`)
    ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

18. Tabel tarif

```
CREATE TABLE IF NOT EXISTS `tarif` (
    `id_tarif` int(10) unsigned NOT NULL AUTO_INCREMENT,
    `id_kurir` int(5) unsigned NOT NULL,
    `id_kota` int(5) unsigned NOT NULL,
    `ongkos_kirim` float(10,0) unsigned NOT NULL,
    PRIMARY KEY (`id_tarif`),
    CONSTRAINT `tarif_ibfk_2` FOREIGN KEY (`id_kota`)
    REFERENCES `kota` (`id_kota`)
    ON DELETE NO ACTION ON UPDATE NO ACTION,
    CONSTRAINT `tarif_ibfk_1` FOREIGN KEY (`id_kurir`)
    REFERENCES `kurir` (`id_kurir`)
    ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

19. Tabel kurs

```
CREATE TABLE IF NOT EXISTS `kurs` (
    `id_kurs` int(5) unsigned NOT NULL AUTO_INCREMENT,
    `id_admin` int(5) unsigned NOT NULL,
    `dollar` float(6,0) NOT NULL,
    PRIMARY KEY (`id_kurs`),
    CONSTRAINT `kurs_ibfk_1` FOREIGN KEY (`id_admin`)
    REFERENCES `admin` (`id_admin`)
    ON DELETE NO ACTION ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

4.2. 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 sistem penjualan *online* pada Van's Computer 354 akan dilampirkan pada halaman lampiran.

Tabel 4.3 Implementasi antarmuka *administrator*

Menu	Deskripsi	Nama File
Login *administrator*	Digunakan sebagai halaman *login administrator*	index.php
Ubah *password administrator*	Digunakan jika *administrator* ingin merubah *password*	akun_pw.php
Ubah profil *administrator*	Digunakan jika *administrator* ingin merubah data pribadi	akun_edit.php
Member	Digunakan untuk melihat, menambah, dan merubah data *member*	member.php
Kota	Digunakan untuk melihat, menambah, dan merubah data kota	kota. php
Provinsi	Digunakan untuk melihat, menambah, dan merubah data provinsi	provinsi. php
Produk	Digunakan untuk melihat, menambah, dan merubah data produk	produk. php
Gambar Produk	Digunakan untuk melihat, menambah, dan merubah data gambar produk	produk_imgedit.php
Kurir	Digunakan untuk melihat, menambah, dan merubah data kurir pengiriman	kurir. php
Tarif	Digunakan untuk melihat, menambah, dan merubah data tarif pengiriman	tarif. php
Kategori	Digunakan untuk melihat, menambah, dan merubah data kategori produk	kategori.php
Merk	Digunakan untuk melihat, menambah, dan merubah data merk produk	merk.php
Transaksi	Digunakan untuk melihat data pembelian, data detail pembelian, dan mengubah status pembelian.	beli.php
Retur	Digunakan untuk melihat data retur dan mengubah status retur	retur.php

Laporan produk	Digunakan untuk melihat laporan produk	lap_prod.php
Laporan penjualan	Digunakan untuk melihat laporan penjualan	lap_trans.php

Tabel 4.4 Implementasi antarmuka *member*

Menu	Deskripsi	Nama File
Halaman Depan	Digunakan untuk menampilkan halaman utama	home.php
Registrasi *member*	Digunakan untuk melakukan pendaftaran *member*	daftar.php
Login *member*	Digunakan sebagai halaman login *member*	index.php
Produk	Digunakan untuk menampilkan data produk	produk.php
Profil	Digunakan untuk menampilkan halaman profil perusahaan	profil.php
Lokasi	Digunakan untuk menampilkan halaman lokasi perusahaan	lokasi.php
Sitemap	Digunakan untuk menampilkan halaman peta/struktur situs *web*	sitemap.php
Jasa Pengiriman	Digunakan untuk menampilkan halaman jasa pengiriman yang didukung	pengiriman.php
Jasa Pembayaran	Digunakan untuk menampilkan halaman jasa pembayaran yang didukung	pembayaran.php
Cara Belanja	Digunakan untuk menampilkan halaman cara berbelanja.	belanja.php
Syarat & Ketentuan	Digunakan untuk menampilkan halaman syarat dan ketentuan yang berlaku	syarat.php
Hubungi Kami	Digunakan untuk menampilkan halaman hubungi kami	hubungi.php
Produk	Digunakan untuk melihat seluruh data barang yang ada	produk.php
Daftar Pembelian	Digunakan untuk melihat seluruh data pembelian yang ada secara detail.	history.php
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus detail transaksi yang akan dilakukan	keranjang.php

4.3. 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.

4.4. Pengujian Alpha

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

4.4.1. Skenario Pengujian Alpha

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 *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Logout	Menghapus *session user*	*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 produk	*Black Box*
	Menampilkan data detail produk	*Black Box*
	Cari barang	*Black Box*
Menu profil	Menampilkan profil perusahaan	*Black Box*
Menu lokasi	Menampilkan lokasi perusahaan	*Black Box*
Menu *sitemap*	Menampilkan peta/struktur situs	*Black Box*
Menu jasa pengiriman	Menampilkan jenis-jenis jasa pengiriman	*Black Box*
Menu jasa pembayaran	Menampilkan jenis-jenis jasa pembayaran	*Black Box*
Menu cara belanja	Menampilkan cara berbelanja	*Black Box*
Menu syarat & ketentuan	Menampilkan aturan yang berlaku	*Black Box*
Menu hubungi kami	Menampilkan kontak perusahaan	*Black Box*
Menu keranjang belanja	Menampilkan data detail pesanan	*Black Box*
Menu daftar pembelian	Menampilkan seluruh data pembelian yang ada secara detail.	*Black Box*

Tabel 4.6. Skenario pengujian aplikasi *administrator*

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Login	Isi data *login*	*Black Box*
	Verifikasi *username*	*Black Box*
	Verivikasi *password*	*Black Box*
Logout	Menghapus *session user*	*Black Box*
Ubah profil	Menampilkan *form* ubah profil *administrator*	*Black Box*
	Ubah data *administrator*	*Black Box*
	Simpan perubahan	*Black Box*
Ubah *password*	Menampilkan *form* ubah *password administrator*	*Black Box*
	Verifikasi *password* lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verfikasi konfirmasi *password*	*Black Box*

Member	Tampilkan data *member*	*Black Box*
Kota	Tampilkan data kota	*Black Box*
	Tambah data kota	*Black Box*
	Ubah data kota	*Black Box*
Provinsi	Tampilkan data provinsi	*Black Box*
	Tambah data provinsi	*Black Box*
	Ubah data provinsi	*Black Box*
Kurir	Tampilkan data kurir	*Black Box*
	Tambah data kurir	*Black Box*
	Ubah data kurir	*Black Box*
Tarif	Tampilkan data tarif	*Black Box*
	Tambah data tarif	*Black Box*
	Ubah data tarif	*Black Box*
Produk	Tampilkan data produk	*Black Box*
	Tambah data produk	*Black Box*
	Ubah data produk	*Black Box*
Gambar produk	Tampilkan data gambar produk	*Black Box*
	Tambah data gambar produk	*Black Box*
	Ubah data gambar produk	*Black Box*
Kategori	Tampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
Merk	Tampilkan data merk	*Black Box*
	Tambah data merk	*Black Box*
	Ubah data merk	*Black Box*
Data transaksi	Tampilkan data transaksi	*Black Box*
	Edit status data transaksi	*Black Box*
Retur	Tampilkan data retur	*Black Box*
	Ubah data retur	*Black Box*
Laporan penjualan	Tampilkan laporan penjualan	*Black Box*
Laporan produk	Tampilkan laporan produk	*Black Box*

4.4.2. Kasus dan Hasil Pengujian

4.4.2.1.Pengujian Registrasi *Member*

Tabel 4.7. Pengujian registrasi *member*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu registrasi *member*	Tampil form registrasi *member*	Dapat menampilkan form registrasi *member*	[√] Diterima [ ] Ditolak
Memasukan data pendaftaran, contoh masukan: Email: 79agus@gmail.com Password: 123456 Nama: Hernawa Alamat: Kubangsari Provinsi: Jawa Barat Kota: Bandung Kodepos: 45552	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka muncul pesan kesalahan bahwa *field* harus terisi.	Dapat memunculkan pesan kesalahan bahwa *field* harus terisi jika salah satu *field* tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Klik tombol daftar	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka muncul pesan kesalahan bahwa *field* harus terisi. Jika data masukan benar muncul pesan “registrasi sukses”	Tampil pesan “registrasi sukses” jika data masukan benar dan memunculkan pesan kesalahan bahwa *field* harus terisi jika salah satu *field* tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Memasukan data kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”data harus diisi”	[√] Diterima [ ] Ditolak

4.4.2.2.Pengujian *Login Member*

Tabel 4.8. Pengujian *login member*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat *email*: 79agus@gmail.com	Mengisikan data alamat *email* pada *field* alamat *email*. Jika salah maka muncul pesan "*email* salah"	Dapat mengisi *field* alamat *email* dan muncul pesan "*email* salah" jika salah dalam pengisian data	[√] Diterima [ ] Ditolak
Memasukan *password*: 123456	Mengisikan data *password* pada *field password*. Jika salah maka muncul pesan "*password* harus lebih dari 5 karakter"	Dapat mengisi *field password* dan muncul pesan "*password* harus lebih dari 5 karakter" jika salah dalam pengisian data	[√] Diterima [ ] Ditolak
Klik tombol *login*	Data *email* dan data *password* dicari	Tombol *login* dapat berfungsi	[√] Diterima [ ] Ditolak
Memasukan data *email* atau *password* yang tidak terdaftar	Tidak dapat *login*	Muncul pesan "*login* gagal, cek *username* dan *password* Anda"	[√] Diterima [ ] Ditolak
Mengosongkan data *email* atau *password*	Tidak dapat *login*	Dapat menampilkan pesan kesalahan "*field* harus diisi"	[√] Diterima [ ] Ditolak

4.4.2.3.Pengujian *Logout*

Tabel 4.9. Pengujian *logout*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik *logout*	Dapat keluar dan menghapus *session* pengguna	*User* keluar dari aplikasi	[√] Diterima [ ] Ditolak

4.4.2.4.Pengujian Ubah Profil *Member*

Tabel 4.10. Pengujian ubah profil *member*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik lihat profil *member*	Menampilkan *form* ubah profil	Dapat menampilkan *form* ubah profil	[√] Diterima [ ] Ditolak
Data profil yang akan diubah	Tercantum pada *textbox* data yang akan diubah	Dapat mengisi *field* data yang akan diubah dengan benar	[√] Diterima [ ] Ditolak
Klik *edit* pada data yang akan diubah	Data berhasil diubah di *database*	Data berhasil diubah.	[√] Diterima [ ] Ditolak
Mengosongkan salah satu *field*	Menampilkan pesan kesalahan bahwa *field* harus terisi.	Dapat menampilkan pesan kesalahan bahwa *field* harus terisi.	[√] Diterima [ ] Ditolak
Mengosongkan semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* harus diisi”	[√] Diterima [ ] Ditolak

4.4.2.5.Pengujian Ubah *Password Member*

Tabel 4.11. Pengujian ubah *password member*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik ubah *password*	Menampilkan *form* ganti *password*	Dapat menampilkan *form* ganti *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
Mengosongkan salah satu *field*	Muncul pesan kesalahan *field* harus diisi	Dapat menampilkan pesan kesalahan ”*field* harus diisi” bila ada *field* yang belum diisi	[√] Diterima [ ] Ditolak

Data *password* lama salah	muncul pesan "*password* lama tidak sesuai"	Dapat menampilkan pesan kesalahan ”*password* lama tidak sesuai”	[√] Diterima [ ] Ditolak
Data *password* baru dan konfirmasi *password* baru tidak sama	Muncul pesan "ulangi *password* baru "	Dapat menampilkan pesan "ulangi *password* baru"	[√] Diterima [ ] Ditolak
Mengosongkan semua *field*	Muncul pesan kesalahan *field* harus diisi	Dapat menampilkan pesan kesalahan *field* harus diisi	[√] Diterima [ ] Ditolak

4.4.2.6.Pengujian Menu Produk

Tabel 4.12. Pengujian menu produk

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu produk	Menampilkan data produk yang ada di *database*	Dapat menampilkan data produk yang ada di *database*	[√] Diterima [ ] Ditolak
Klik detail pada data produk	Menampilkan detail produk	Dapat menampilkan detail produk	[√] Diterima [ ] Ditolak
Klik beli	Menuju ke keranjang belanja	Dapat menampilkan keranjang belanja	[√] Diterima [ ] Ditolak

4.4.2.7.Pengujian Menu Profil

Tabel 4.13. Pengujian menu profil

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik profil	Menampilkan profil perusahaan	Tampil profil perusahaan	[√] Diterima [ ] Ditolak

4.4.2.8.Pengujian Menu Jasa Pengiriman

Tabel 4.14. Pengujian menu jasa pengiriman

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik jasa pengiriman	Menampilkan menu biaya pengiriman	Tampil menampilkan menu biaya pengiriman	[√] Diterima [ ] Ditolak

4.4.2.9.Pengujian Menu Jasa Pembayaran

Tabel 4.15. Pengujian menu jasa pembayaran

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik jasa pembayaran	Menampilkan menu jasa pembayaran	Tampil menampilkan menu jasa pembayaran	[√] Diterima [ ] Ditolak

4.4.2.10.Pengujian Menu Cara Belanja

Tabel 4.16. Pengujian menu cara belanja

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik cara belanja	Menampilkan menu cara belanja	Menampilkan cara belanja	[√] Diterima [ ] Ditolak

4.4.2.11.Pengujian Menu Syarat dan Ketentuan

Tabel 4.17. Pengujian menu syarat dan ketentuan

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik syarat dan ketentuan	Menampilkan menu syarat dan ketentuan	Tampil menu syarat dan ketentuan	[√] Diterima [ ] Ditolak

4.4.2.12.Pengujian Menu Keranjang Belanja

Tabel 4.18. Pengujian menu keranjang belanja

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu keranjang belanja	Menampilkan *form* keranjang belanja	Dapat menampilkan *form* keranjang belanja	[√] Diterima [ ] Ditolak
Menampilkan data jumlah pesanan	Tercantum pada *textbox* data yang akan diubah	Dapat mengisi data jumlah dengan benar	[√] Diterima [ ] Ditolak
Klik tombol ubah	Data jumlah yang diubah tersimpan di *database*	Dapat menampilkan perubahan data jumlah	[√] Diterima [ ] Ditolak
Klik tombol hapus	Data yang dihapus terhapus dari keranjang belanja	Dapat menghapus data pada keranjang belanja	[√] Diterima [ ] Ditolak

4.4.2.13.Pengujian Menu Daftar Pembelian

Tabel 4.19. Pengujian menu daftar pembelian

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu daftar pembelian	Menampilkan daftar pembelian	Dapat menampilkan daftar pembelian	[√] Diterima [ ] Ditolak
Klik detail	Menampilkan detail pembelian	Dapat menampilkan detail pembelian	[√] Diterima [ ] Ditolak

4.4.2.14.Pengujian *Login Administrator*

Tabel 4.20. Pengujian *login administrator*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Username: admin	Tercantum pada *textbox username*	Dapat mengisi *field username* dengan benar	[√] Diterima [ ] Ditolak
Password: qwerty	Tercantum pada *textbox password*	Dapat mengisi *field password* dengan benar	[√] Diterima [ ] Ditolak
Klik tombol *login*	Data *username* dan data *password* dicari	Tombol *login* dapat berfungsi	[√] Diterima [ ] Ditolak
Memasukan data *username* atau *password* yang tidak terdaftar	Tidak dapat *login*	Muncul pesan "*login* gagal, cek *username* dan *password* anda"	[√] Diterima [ ] Ditolak
Tidak memasukan data *username* atau *password*	Tidak dapat *login*	Dapat menampilkan pesan kesalahan "*login* gagal, cek *username* dan *password* anda"	[√] Diterima [ ] Ditolak

4.4.2.15.Pengujian Ubah *Password Administrator*

Tabel 4.21. Pengujian ubah *password administrator*

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

Memasukan data *password* lama yang salah	Muncul pesan "*password* salah"	Dapat menampilkan pesan kesalahan ”*password* salah”	[√] Diterima [ ] Ditolak
Memasukan data *password* baru dah 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

4.4.2.16.Pengujian Data Provinsi

Tabel 4.22. Pengujian data provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik provinsi	Menampilkan data provinsi	Dapat menampilkan data provinsi	[√] Diterima [ ] Ditolak
Klik *edit* pada data yang akan diubah	Menampilkan *form edit* provinsi	Dapat menampilkan *form* provinsi	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Muncul pesan kesalahan “*field* harus diisi”	Dapat muncul pesan kesalahan “*field* 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
Klik tombol *edit*	Data berhasil diubah di *database*	Tampil pesan “data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.2.17.Pengujian Tambah Data Provinsi

Tabel 4.23. Pengujian tambah provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah	Menampilkan *form* tambah provinsi	Dapat menampilkan *form* tambah provinsi	[√] Diterima [ ] Ditolak
Mengisi data provinsi Contoh: Nama: 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
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Klik tombol tambah	Data provinsiakan tersimpan didalam *database*	Data provinsi akan tersimpan didalam *database* dan muncul pesan ”data berhasil ditambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.2.18.Pengujian Data Kota

Tabel 4.24. Pengujian data kota

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik kota	Menampilkan data kota	Dapat menampilkan data kota	[√] Diterima [ ] Ditolak
Klik *edit* pada data yang akan diubah	Menampilkan *form edit* kota	Dapat menampilkan *form* kota	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu	Muncul pesan	Dapat muncul	[√] Diterima

field	kesalahan *field* harus diisi	pesan kesalahan *field* harus diisi	[ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Klik tombol *edit*	Data berhasil diubah di *database*	Tampil pesan “data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.2.19.Pengujian Tambah Data Kota

Tabel 4.25. Pengujian tambah kota

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah	Menampilkan *form* tambah kota	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data kota Nama kota: Bandung	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa “*field* harus terisi”	Dapat memunculkan pesan kesalahan bahwa “*field* harus terisi” jika salah satu *field* tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Klik tombol tambah	Data kota tersimpan didalam *database*	Data kota tersimpan didalam *database* dan muncul pesan ”data berhasil ditambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.2.20.Pengujian Data Kurir

Tabel 4.26. Pengujian data kurir

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik jenis kurir	Menampilkan data kurir	Dapat menampilkan data kurir	[√] Diterima [ ] Ditolak
Klik *edit* pada data yang akan diubah	Menampilkan *form edit* kurir	Dapat menampilkan *form* kurir	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* harus diisi	Dapat muncul pesan kesalahan *field* 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
Klik tombol *edit*	Data berhasil diubah di *database*	Tampil pesan “data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.2.21.Pengujian Tambah Data Kurir

Tabel 4.27. Pengujian tambah kurir

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah	Menampilkan *form* tambah kurir	Dapat menampilkan *form* tambah kurir	[√] Diterima [ ] Ditolak
Mengisi data pengiriman Contoh: Jasa: JNE Jenis: Reguler	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* harus terisi.	Dapat memunculkan pesan kesalahan bahwa *field* harus terisi jika salah satu *field* tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak

Klik tombol tambah	Data kurir tersimpan didalam *database*	Data kurir tersimpan didalam *database* dan muncul pesan ”data berhasil ditambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.2.22.Pengujian Data Tarif

Tabel 4.28. Pengujian data tarif

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tarif	Menampilkan data tarif	Dapat menampilkan data biaya pengiriman	[√] Diterima [ ] Ditolak
Klik *edit* pada data yang akan diubah	Menampilkan *form edit* tarif	Dapat menampilkan *form* tarif	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* harus diisi	Dapat muncul pesan kesalahan *field* 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
Klik tombol *edit*	Data berhasil diubah di *database*	Tampil pesan “data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.2.23.Pengujian Tambah Data Tarif

Tabel 4.29. Pengujian tambah tarif

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah	Menampilkan *form* tambah tarif	Dapat menampilkan *form* tambah tarif	[√] Diterima [ ] Ditolak
Mengisi data pengiriman Contoh:	Jika salah satu *field* tidak terisi atau salah dalam pengisian,	Dapat memunculkan pesan kesalahan bahwa *field* harus	[√] Diterima [ ] Ditolak

Provinsi: Jawa Barat Kota: Bandung Kurir: JNE - Reguler Tarif: 5000	maka akan muncul pesan kesalahan bahwa field harus terisi.	terisi jika salah satu *field* tidak terisi atau salah dalam pengisian.	
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Klik tombol tambah	Data tarif tersimpan didalam *database*	Data tarif tersimpan didalam *database* dan muncul pesan ”data berhasil ditambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.2.24.Pengujian Data Kategori

Tabel 4.30. Pengujian data kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik kategori	Menampilkan data kategori	Dapat menampilkan data kategori	[√] Diterima [ ] Ditolak
Klik *edit* pada data yang akan diubah	Menampilkan *form edit* kategori	Dapat menampilkan *form* kategori	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* harus diisi	Dapat muncul pesan kesalahan *field* 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
Klik tombol *edit*	Data berhasil diubah di *database*	Tampil pesan “data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.2.25.Pengujian Tambah Data Kategori

Tabel 4.31. Pengujian tambah kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah	Menampilkan *form* tambah kategori	Dapat menampilkan *form* tambah kategori	[√] Diterima [ ] Ditolak
Mengisi data Nama kategori: LGA 775	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* harus terisi.	Dapat memunculkan pesan kesalahan bahwa *field* harus terisi jika salah satu *field* tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Klik tombol tambah	Data kategori tersimpan didalam *database*	Data kategori tersimpan didalam *database* dan muncul pesan ”data berhasil ditambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.2.26.Pengujian Data Produk

Tabel 4.32. Pengujian data produk

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik produk	Menampilkan data produk	Dapat menampilkan data produk	[√] Diterima [ ] Ditolak
Klik *edit* pada data yang akan diubah	Menampilkan *form edit* produk	Dapat menampilkan *form* produk	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* harus diisi	Dapat muncul pesan kesalahan *field* harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi	Tampilkan	Dapat menampilkan pesan	[√] Diterima

semua *field*	pesan kesalahan	kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[ ] Ditolak
Klik tombol *edit*	Data berhasil diubah di *database*	Tampil pesan “data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.2.27.Pengujian Tambah Data Produk

Tabel 4.33. Pengujian tambah produk

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah	Menampilkan *form* tambah produk	Dapat menampilkan *form* tambah produk pengiriman	[√] Diterima [ ] Ditolak
Mengisi data produk Contoh: Nama Produk : ROG Formula Merk : Asus Kategori: Motherboard Sub Kategori: LGA 1366 Harga: 4250000 Stok: 20 Berat : 1100 Diskon : 10%	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* harus terisi.	Dapat memunculkan pesan kesalahan bahwa *field* harus terisi jika salah satu *field* tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] Diterima [ ] Ditolak
Klik tombol tambah	Data produk tersimpan didalam *database*	Data produk tersimpan didalam *database* dan muncul pesan ”data berhasil ditambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.2.28.Pengujian Data Transaksi

Tabel 4.34. Pengujian data transaksi

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik transaksi	Menampilkan data *list* transaksi	Dapat menampilkan data *list* transaksi	[√] Diterima [ ] Ditolak

4.4.2.29.Pengujian Data Retur

Tabel 4.35. Pengujian data retur

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik retur	Menampilkan data retur	Dapat menampilkan data retur	[√] Diterima [ ] Ditolak

4.4.2.30.Pengujian Data Laporan Produk

Tabel 4.36. Pengujian data laporan produk

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik laporan produk	Menampilkan data laporan produk	Dapat menampilkan data laporan stok produk	[√] Diterima [ ] Ditolak
Klik cetak	Menampilkan hasil laporan dalam bentuk pdf	Dapat menampilkan hasil laporan dalam bentuk pdf	[√] Diterima [ ] Ditolak

4.4.2.31.Pengujian Data Laporan Penjualan

Tabel 4.37. Pengujian data laporan penjualan

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik laporan penjualan	Menampilkan data laporan penjualan	Dapat menampilkan data laporan penjualan	[√] Diterima [ ] Ditolak
Klik cetak	Menampilkan hasil laporan dalam bentuk pdf	Dapat menampilkan hasil laporan dalam bentuk pdf	[√] Diterima [ ] Ditolak

4.4.3. Kesimpulan Pengujian Alpha

Berdasarkan hasi pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan sistem penjualan *online* pada Van's Computer 354 bebas kesalahan penulisan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.5. Pengujian Beta

4.5.1. Skenario Pengujian Beta

Pengujian beta ini dilakukan dengan cara mendistribusikan langsung aplikasi *e-commerce* kepada publik, dengan tujuan untuk mendapatkan respon seputar kekurangan dan kelebihan aplikasi *e-commerce* yang dibangun.

4.5.1.1. Pengujian Lingkungan Karyawan Toko

Penelitian di lingkungan karyawan toko untuk mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

Adapun beberapa poin yang diteliti adalah sebagai berikut :

a. Bagaimana tampilan antarmuka pengolahan data *website* ini.
b. Kemudahan penggunaan aplikasi dalam mengolah data transaksi.
c. Sejauh mana *website e-commerce* ini membantu dalam mengelola kegiatan operasional.
d. Apakah laporan yang disediakan pada *website* ini sudah lengkap.
e. Apakah *website* ini sudah layak untuk di-*online*-kan.

Kuisioner ini ditujukan kepada 1 orang *administrator* dan 3 orang operator sebagai pengguna sistem dimana kuisioner terdiri dari 5 pertanyaan.

Berdasarkan data hasil kuisioner tersebut, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus: Y = P/Q * 100%

Keterangan :

P = Banyaknya jawaban responden tiap soal.

Q = Jumlah responden

Y = Nilai persentase

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas.

1. Apakah Anda setuju bahwa tampilan *website* cukup informatif dan mudah dipahami?

Tabel 4. 38. Hasil pengujian kuisioner karyawan toko soal nomor 1

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Persentase (%)**
1	1	Sangat setuju	0	0
	2	Setuju	3	75
	3	Biasa saja	1	25
	4	Tidak setuju	0	0
	5	Sangat tidak setuju	0	0
Jumlah			4	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 75% menyatakan setuju *website* ini memiliki tampilan yang informatif dan mudah dipahami, dan 25% menyatakan cukup setuju.

2. Apakah Anda setuju bahwa pengolahan data transaksi menggunakan *website* ini menjadi lebih mudah?

Tabel 4. 39. Hasil pengujian kuisioner karyawan toko soal nomor 2

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Persentase (%)**
2	1	Sangat setuju	0	0
	2	Setuju	4	100
	3	Biasa saja	0	0
	4	Tidak setuju	0	0
	5	Sangat tidak setuju	0	0
Jumlah			4	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 100% menyatakkan setuju bahwa bahwa pengolahan data transaksi menggunakan *website* ini menjadi lebih mudah.

3. Sejauh mana *website* ini membantu Anda dalam mengelola kegiatan operasional toko?

Tabel 4. 40. Hasil pengujian kuisioner karyawan toko soal nomor 3

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Persentase (%)**
3	1	Sangat membantu	1	25
	2	Membantu	3	75
	3	Biasa saja	0	0
	4	Tidak membantu	0	0
	5	Sangat tidak membantu	0	0
Jumlah			4	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 75% menyatakan *website* ini membantu dalam mengelola kegiatan operasional toko, sedangkan 25% menyatakan sangat membantu.

4. Apakah laporan yang tersedia pada *website* ini sudah lengkap?

Tabel 4. 41. Hasil pengujian kuisioner karyawan toko soal nomor 4

Pertanyaan	No.	Keterangan	Responden	Persentase (%)
4	1	Sangat setuju	0	0
	2	Setuju	3	75
	3	Biasa saja	1	1
	4	Tidak setuju	0	0
	5	Sangat tidak setuju	0	0
Jumlah			4	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 75% menyatakan setuju bahwa laporan yang tersedia pada *website* ini sudah lengkap, sedangkan 25%-nya menyatakan biasa saja.

5. Apakah Anda setuju bahwa *website* ini layak untuk dipublikasikan secara *online*?

Tabel 4. 42. Hasil pengujian kuisioner karyawan toko soal nomor 5

Pertanyaan	No.	Keterangan	Responden	Persentase (%)
5	1	Sangat setuju	0	0
	2	Setuju	4	100
	3	Biasa saja	0	0
	4	Tidak setuju	0	0
	5	Sangat tidak setuju	0	0
Jumlah			4	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 100% menyatakan setuju *website* ini layak untuk dipublikasikan secara *online*.

4.5.1.2.Pengujian Lingkungan *Member*

Penelitian dilakukan di lingkungan *member* khususnya yang sudah terbiasa dengan teknologi internet, dilakukan untuk mengetahui sejauh mana sistem yang dibangun dapat memberikan kemudahan kepada pelanggan untuk melakukan transaksi pembelian barang

Adapun beberapa poin yang diteliti adalah :

a. Tampilan antarmuka yang *user friendly*.
b. *Website* dapat memenuhi kebutuhan informasi pelanggan dan memudahkan pelanggan dalam melakukan transaksi pembelian barang.
c. Sistem pembayaran yang mudah digunakan.
d. *Website* menjadi sarana memperluas pangsa pasar toko.
e. Kelayakan *website* untuk dipublikasikan.

Berdasarkan data hasil kusioner tersebut, dapat dicari persentase masing-masing jawaban dengan menggunakan rumus: Y = (P/Q) * 100%

Keterangan :

P = Banyaknya jawaban responden tiap soal.

Q = Jumlah responden

Y = Nilai persentase

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuisioner ini diujikan kepada 10 orang *member*.

1. Apakah Anda setuju bahwa tampilan *website* cukup menarik dan mudah dipahami?

Tabel 4.43. Hasil pengujian kuisioner *member* soal nomor 1

Pertanyaan	No.	Keterangan	Responden	Persentase(%)
1	1	Sangat Setuju	0	0
	2	Setuju	6	60
	3	Biasa Saja	3	30
	4	Tidak Setuju	1	10
	5	Sangat Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil presentase diatas maka dapat disimpulkan sebanyak 6 orang atau 60% menyatakan setuju, 3 orang atau 30% menyatakan biasa saja, dan 1 orang atau 10% menyatakan tidak setuju, bahwa tampilan *website* cukup menarik dan mudah untuk dipahami.

2. Apakah Anda setuju bahwa *website* ini lebih memudahkan pelanggan dalam melakukan transaksi pembelian produk?

Tabel 4.44. Hasil pengujian kuisioner *member* soal nomor 2

Pertanyaan	No.	Keterangan	Responden	Persentase(%)
2	1	Sangat Setuju	2	20
	2	Setuju	7	70
	3	Biasa Saja	1	10
	4	Tidak Setuju	0	0
	5	Sangat Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 2 orang atau 20% menyatakan sangat setuju, 7 orang atau 70% menyatakan setuju, dan 1 orang atau 10% menyatakan biasa saja bahwa *website* ini akan lebih memudahkan pelanggan dalam melakukan transaksi pembelian produk.

3. Apakah Anda setuju bahwa sistem pembayaran yang digunakan lebih memudahkan transaksi pembelian?

Tabel 4.45. Hasil pengujian kuisioner *member* soal nomor 3

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Persentase(%)**
3	1	Sangat Setuju	4	40
	2	Setuju	6	60
	3	Biasa Saja	0	0
	4	Tidak Setuju	0	0
	5	Sangat Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 4 atau 40% menyatakan sangat setuju dan 6 orang atau 60% menyatakan setuju bahwa sistem pembayaran yang digunakan lebih memudahkan transaksi pembelian.

4. Apakah Anda setuju bahwa *website* ini dapat memperluas pangsa pasar toko?

Tabel 4.46. Hasil pengujian kuisioner *member* soal nomor 4

Pertanyaan	No.	Keterangan	Responden	Persentase(%)
4	1	Sangat Setuju	3	30
	2	Setuju	5	50
	3	Biasa Saja	2	20
	4	Tidak Setuju	0	0
	5	Sangat Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 3 orang atau 30% menyatakan sangat setuju, 5 orang atau 50% menyatakan setuju, dan 2 atau 20% orang menyatakan biasa saja bahwa *website* ini dapat memperluas pangsa pasar toko.

5. Apakah Anda setuju bahwa *website* ini layak untuk dipublikasikan secara *online*?

Tabel 4.47. Hasil pengujian kuisioner *member* soal nomor 5

Pertanyaan	No.	Keterangan	Responden	Persentase(%)
5	1	Sangat Setuju	1	10
	2	Setuju	7	70
	3	Biasa Saja	2	20
	4	Tidak Setuju	0	0
	5	Sangat Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 1 orang atau 10% menyatakan sangat setuju, 7 orang atau 70 % menyatakan setuju,

dan 2 orang atau 20% menyatakan biasa saja bahwa *website* ini layak untuk dipublikasikan secara *online*.

4.5.2. Kesimpulan Pengujian Beta

Berdasarkan pengujian beta diatas, dapat diambil kesimpulan bahwa:

1. Aplikasi ini memudahkan pelanggan dalam mendapatkan informasi barang yang tersedia.
2. Pelanggan dapat melakukan transaksi pembelian barang secara *online* dengan mudah dan nyaman.
3. Proses pembayaran yang mudah dan cukup membantu para pelanggan dalam bertransaksi.
4. Aplikasi ini juga memudahkan pihak toko dalam melaksanankan kegiatan operasional dan mendokumentasikan data transaksi penjualan barang.
5. Aplikasi ini juga dapat memperluas pangsa pasar Van’s Computer 354 karena tidak terbatas secara geografis.