BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

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 (*hardware*) yang digunakan dalam pembangunan aplikasi *E-Commerce* di Toko Gayaku adalah tercantum pada tabel 4.1 sebagai berikut:

Tabel 4.1 Spesifikasi Perangkat Keras Yang Digunakan

Perangkat Keras	Spesifikasi
Prosesor	Intel Core Duo 1.86 Ghz
RAM	1024 MB
VGA	256 MB
Harddisk	160 GB
Monitor	15” Resolusi 1028 x 800

4.1.2 Perangkat lunak yang digunakan

Spesifikasi perangkat lunak (*software*) yang digunakan dalam pembangunan layanan *e-Commerce* di Toko Gayaku adalah tercantum pada tabel 4.2 sebagai berikut:

Tabel 4.2 Spesifikasi Perangkat Lunak Yang Digunakan

Perangkat Lunak	Keterangan
Sistem Operasi	Microsoft Windows 7 Professional
Bahasa Pemrograman	PHP
DBMS	MySQL 5.1
Web Browser	Mozilla Firefox 4
Code Editor	Adobe Dreamweaver CS4
DFD modeler	Microsoft Office Visio 2007

4.1.3 Implementasi *Database*

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

1. Tabel tbl_admin

```
CREATE TABLE IF NOT EXISTS `tbl_admin` (
  `id_admin` char(5) NOT NULL,
  `nama_lengkap` varchar(50) NOT NULL,
  `username` varchar(20) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(70) NOT NULL,
  `level_admin` enum('Admin','Superadmin','Gudang') NOT NULL,
  `status_admin` tinyint(1) NOT NULL,
  PRIMARY KEY  (`id_admin`)
) ENGINE=InnoDB;
```

2. Tabel tbl_pelanggan

```
CREATE TABLE IF NOT EXISTS `tbl_pelanggan` (
  `id_pelanggan` char(6) NOT NULL,
  `nama_lengkap` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(70) NOT NULL,
  `ulang_password` varchar(70) NOT NULL,
  `alamat_pelanggan` text NOT NULL,
  `id_propinsi` int(11) NOT NULL,
  `id_kabkota` int(11) NOT NULL,
  `kodepos` int(5) NOT NULL,
  `no_hp` varchar(15) NOT NULL,
  `status` enum('Aktif','Tidak Aktif') NOT NULL,
  PRIMARY KEY  (`id_pelanggan`),
  KEY `id_propinsi` (`id_propinsi`),
  KEY `id_kabkota` (`id_kabkota`)
) ENGINE=InnoDB;
```

3. Tabel tbl_kategori

```
CREATE TABLE IF NOT EXISTS `tbl_kategori` (
  `id_kategori` char(5) NOT NULL,
  `nama_kategori` varchar(50) NOT NULL,
  PRIMARY KEY  (`id_kategori`)
) ENGINE=InnoDB;
```

4. Tabel tbl_subkategori

```
CREATE TABLE IF NOT EXISTS `tbl_subkategori` (
  `id_subkategori` char(5) NOT NULL,
  `id_kategori` char(5) NOT NULL,
  `nama_subkategori` varchar(50) NOT NULL,
  PRIMARY KEY  (`id_subkategori`),
  KEY `id_kategori` (`id_kategori`)
) ENGINE=InnoDB;
```

5. Tabel tbl_ukuran

```
CREATE TABLE IF NOT EXISTS `tbl_ukuran` (
  `id_ukuran` char(6) NOT NULL,
  `id_kategori` char(6) NOT NULL,
  `nama_ukuran` varchar(30) NOT NULL,
  `no_ukuran` varchar(10) NOT NULL,
  PRIMARY KEY  (`id_ukuran`),
  KEY `id_kategori` (`id_kategori`)
) ENGINE=InnoDB;
```

6. Tabel tbl_produk

```
CREATE TABLE IF NOT EXISTS `tbl_produk` (
  `id_produk` char(6) NOT NULL,
  `id_subkategori` char(5) NOT NULL,
  `id_kategori` char(5) NOT NULL,
  `id_ukuran` char(6) NOT NULL,
  `nama_produk` varchar(50) NOT NULL,
  `harga` int(20) NOT NULL,
  `stok` int(20) NOT NULL,
  `warna` varchar(30) NOT NULL,
  `berat` float NOT NULL,
  `diskon` int(5) NOT NULL,
  `file_gambar` varchar(200) NOT NULL,
  `promo` enum('Ya','Tidak') NOT NULL,
  `deskripsi` text NOT NULL,
  `tgl_diinputkan` timestamp NOT NULL default CURRENT_TIMESTAMP
on update CURRENT_TIMESTAMP,
  `counter` int(11) NOT NULL,
  PRIMARY KEY  (`id_produk`),
  KEY `id_subkategori` (`id_subkategori`),
  KEY `id_kategori` (`id_kategori`),
  KEY `id_ukuran` (`id_ukuran`)
) ENGINE=InnoDB;
```

7. Tabel tbl_propinsi

```
CREATE TABLE IF NOT EXISTS `tbl_propinsi` (
  `id_propinsi` int(10) NOT NULL,
  `nama` varchar(100) NOT NULL,
  PRIMARY KEY  (`id_propinsi`)
) ENGINE=InnoDB;
```

8. Tabel tbl_kabkota

```
CREATE TABLE IF NOT EXISTS `tbl_kabkota` (
  `id_propinsi` int(10) NOT NULL,
  `id_kabkota` int(10) NOT NULL,
  `nama_kabkota` varchar(100) NOT NULL,
  PRIMARY KEY  (`id_kabkota`),
  KEY `id_propinsi` (`id_propinsi`)
) ENGINE=InnoDB;
```

9. Tabel tbl_jenis_pengiriman

```
CREATE TABLE IF NOT EXISTS `tbl_jenis_pengiriman` (
  `id_jenis_pengiriman` char(5) NOT NULL,
  `nama_jenis_pengiriman` varchar(50) NOT NULL,
  UNIQUE KEY `id_jenis_pengiriman` (`id_jenis_pengiriman`)
) ENGINE=InnoDB;
```

10. Tabel tbl_ongkos

```
CREATE TABLE IF NOT EXISTS `tbl_ongkos` (
  `id_ongkos` char(5) NOT NULL,
  `id_jenis_pengiriman` char(5) NOT NULL,
  `id_kabkota` int(10) NOT NULL,
  `biaya` int(20) NOT NULL,
  PRIMARY KEY  (`id_ongkos`),
  KEY `id_jenis_pengiriman` (`id_jenis_pengiriman`),
  KEY `id_kabkota` (`id_kabkota`)
) ENGINE=InnoDB;
```

11. Tabel matauang

```
CREATE TABLE IF NOT EXISTS `matauang` (
  `kode` varchar(3) NOT NULL,
  `kurs` double NOT NULL
) ENGINE=InnoDB;
```

12. Tabel tbl_retur

```
CREATE TABLE IF NOT EXISTS `tbl_retur` (
  `id_retur` char(8) NOT NULL,
  `no_pesanan` char(8) NOT NULL,
  `id_pelanggan` char(6) NOT NULL,
  `penerima` varchar(50) NOT NULL,
  `alamat` varchar(100) NOT NULL,
  `keterangan` text NOT NULL,
  `tgl_retur` date NOT NULL,
  `status` enum('Dikirim','Diterima','Ditolak') NOT NULL,
  PRIMARY KEY  (`id_retur`),
  UNIQUE KEY `no_pesanan`
(`no_pesanan`,`id_pelanggan`,`penerima`,`alamat`),
  KEY `id_pelanggan` (`id_pelanggan`)
) ENGINE=InnoDB;
```

13. Tabel tbl_detail_retur

```
CREATE TABLE IF NOT EXISTS `tbl_detail_retur` (
  `id_retur` char(8) NOT NULL,
  `id_produk` char(6) NOT NULL,
  `no_ukuran` varchar(10) NOT NULL,
  `quantity` int(20) NOT NULL,
  `harga` int(30) NOT NULL,
  PRIMARY KEY  (`id_retur`,`id_produk`),
  KEY `id_produk` (`id_produk`)
) ENGINE=InnoDB;
```

14. Tabel tbl_keranjang

```
CREATE TABLE IF NOT EXISTS `tbl_keranjang` (
  `id_pelanggan` char(6) NOT NULL,
  `id_produk` char(6) NOT NULL,
  `nama_produk` varchar(30) NOT NULL,
  `ukuran` varchar(10) NOT NULL,
  `berat` float NOT NULL,
  `harga` int(20) NOT NULL,
  `qty` int(20) NOT NULL,
  `subtotal` int(20) NOT NULL,
  KEY `id_pelanggan` (`id_pelanggan`),
  KEY `id_produk` (`id_produk`)
) ENGINE=InnoDB;
```

15. Tabel tbl_pesanan

```
CREATE TABLE IF NOT EXISTS `tbl_pesanan` (
  `no_pesanan` char(8) NOT NULL,
  `id_pelanggan` char(6) NOT NULL,
  `tgl_dipesan` date NOT NULL,
  `tgl_expired` date NOT NULL,
  `tgl_terima` date NOT NULL,
  `id_ongkos` char(5) default NULL,
  `tarif_ongkos_kirim` int(20) NOT NULL,
  `ukuran` varchar(10) NOT NULL,
  `berat` float NOT NULL,
  `nama_penerima` varchar(50) NOT NULL,
  `alamat_penerima` text NOT NULL,
  `no_telp_penerima` varchar(15) NOT NULL,
  `status_pengiriman` enum('Belum Dikirim','Sudah
Dikirim','Diterima','Retur') NOT NULL,
  `status_pesanan` enum('Dipesan','Dibatalkan') NOT NULL,
  `total_pembayaran` int(15) NOT NULL,
  `no_resi` varchar(20) NOT NULL,
  PRIMARY KEY  (`no_pesanan`),
  KEY `id_pelanggan` (`id_pelanggan`),
```

```
  KEY `id_ongkos` (`id_ongkos`)
) ENGINE=InnoDB;
```

16. Tabel tbl_detail_pesanan

```
CREATE TABLE IF NOT EXISTS `tbl_detail_pesanan` (
  `no_pesanan` char(8) NOT NULL,
  `id_produk` char(6) NOT NULL,
  `qty` int(20) NOT NULL,
  `ukuran` varchar(10) NOT NULL,
  `harga` int(20) NOT NULL,
  `diskon` int(5) NOT NULL,
  PRIMARY KEY  (`no_pesanan`,`id_produk`),
  KEY `id_produk` (`id_produk`)
) ENGINE=InnoDB;
```

17. Tabel tbl_pembayaran

```
CREATE TABLE IF NOT EXISTS `tbl_pembayaran` (
  `id_pembayaran` char(8) NOT NULL,
  `no_pesanan` char(8) NOT NULL,
  `tgl_bayar` date NOT NULL,
  `no_rekening_pengirim` varchar(25) NOT NULL,
  `no_rekening_tujuan` varchar(25) NOT NULL,
  `atas_nama` varchar(50) NOT NULL,
  `status_pembayaran` enum('Belum Bayar','Sudah Bayar','Lunas')
NOT NULL,
  `total_pembayaran` int(20) NOT NULL,
  `jumlah_dibayar` int(20) NOT NULL,
  `id_paypal` varchar(20) NOT NULL,
  `no_transaksi` varchar(30) NOT NULL,
  PRIMARY KEY  (`id_pembayaran`),
  KEY `no_pesanan` (`no_pesanan`)
) ENGINE=InnoDB;
```

4.1.4 Impelementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna.

Tabel 4.3 Implementasi Antarmuka *Backend*

Menu	Deskripsi	Nama File
Login	Digunakan sebagai halaman login administrator atau gudang	index.php
Beranda	Merupakan tampilan beranda administrator atau gudang	beranda.html
Edit data profil	Digunakan untuk merubah profil administrator atau gudang	administrator_profil.php
Ubah password	Digunakan untuk merubah password administrator atau gudang	administrator_ubah_password.php
Data admin	Digunakan administrator untuk melihat, menambah, menghapus dan mencari data gudang	admin_list.php
Data pelanggan	Digunakan administrator untuk melihat, menambah, menghapus dan mencari data pelanggan	pelanggan_list.php
Data kategori	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data kategori	kategori_list.php
Data subkategori	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data subkategori	subkategori_list.php
Data ukuran	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data ukuran	ukuran_list.php
Data produk	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data produk serta melihat data detail produk	produk_list.php
Data Propinsi	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data propinsi	propinsi_list.php
Data Kab/kota	Digunakan untuk melihat, menambah, merubah,	kabkota_list.php

	menghapus dan mencari data kab/kota	
Data jenis pengiriman	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data jenis pengiriman	jenis_pengiriman_list.php
Data ongkos kirim	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data ongkos kirim	ongkoskirim_list.php
Data mata uang	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data mata uang	matauang_list.php
Data retur barang	Digunakan untuk mengolah data retur barang	returbarang_list.php
Data transaksi	Digunakan untuk mengolah data transaksi yaitu data pemesanan dan data pembayaran	pesanan_list.php
Laporan produk	Digunakan untuk melihat laporan produk per-tanggal, per-bulan dan mencetak laporan	laporan_semua_produk.php
Laporan pesanan	Digunakan untuk melihat laporan pesanan per-tanggal, per-bulan dan mencetak laporan	laporanpenjualanpertanggal.php
Logout	Digunakan administrator atau gudang untuk keluar dari aplikasi	logout.php

Tabel 4.4 Implementasi Antarmuka *Frontend*

Menu	Deskripsi	Nama File
Login	Digunakan pelanggan sebagai halaman login	login.php
Lupa password	Digunakan jika pelanggan lupa password untuk login	lupa_pass_login.php
Beranda	Merupakan tampilan beranda pelanggan dan pengunjung	index.php
Edit data profil	Digunakan untuk merubah profil pelanggan	edit_data_pelanggan.php
Ubah password	Digunakan untuk merubah password pelanggan	ubah_password_pelanggan.php
Tentang kami	Digunakan pelanggan dan pengunjung untuk melihat profil perusahaan	tentangkami.php
Cara pesan dan bayar	Digunakan pelanggan dan pengunjung untuk melihat cara pemesanan dan pembayaran	pesanbayar.php
History pemesanan dan	Digunakan pelanggan untuk melihat history transaksi dan	history_pesanan.php

pembayaran	digunakan untuk melakukan pembayaran	
Klaim retur barang	Digunakan pelanggan untuk mereturkan pesanan	returbarang.php
Daftar	Digunakan pengunjung yang ingin mendaftar menjadi *member*	register.php
Info Produk	Digunakan pelanggan dan pengunjung untuk melihat data detail produk	detail_produk.php
Beli Produk	Digunakan pelanggan untuk memilih ukuran dan warna produk yang dibeli	beli_produk.php
Keranjang Belanja	Digunakan pelanggan untuk melihat, merubah, menghapus data produk yang dipesan	list_keranjang_belanja.php
Isi data pengiriman	Digunakan pelanggan untuk mengisi data penerima pesanan	isidatatransaksi.php
Selesai Transaksi	Digunakan pelanggan untuk menampilkan data pesanan, mencetak nota pesanan dan pembayaran *PayPal*	selesai_transaksi.php
Logout	Digunakan pelanggan untuk keluar dari halaman pelanggan	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.

4.2.1 Pengujian *Alpha*

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

4.2.1.1 Skenario Pengujian *Alpha*

Tabel 4.5 Skenario Pengujian *Alpha* Administrator

Kelas Uji	Butir Uji	Jenis Pengujian
Login administrator	Isi data login	*Black Box*
	Verifikasi login	*Black Box*
Edit profil administrator	Tampil profil gudang	*Black Box*
	Edit data profil gudang	*Black Box*
	Simpan perubahan	*Black Box*
Ubah password administrator	Tampil *form* ubah password administrator	*Black Box*
Data pelanggan	Tampil data pelanggan	*Black Box*
	Tambah data pelanggan	*Black Box*
	Hapus data pelanggan	*Black Box*
	Cari data pelanggan	*Black Box*
Data propinsi	Tampil data propinsi	*Black Box*
	Tambah data propinsi	*Black Box*
	Edit data propinsi	*Black Box*
	Hapus data propinsi	*Black Box*
	Cari data propinsi	*Black Box*
Data kabkota	Tampil data kab/kota	*Black Box*
	Tambah data kab/kota	*Black Box*
	Edit data kab/kota	*Black Box*
	Hapus data kab/kota	*Black Box*
	Cari data kab/kota	*Black Box*
Data jenis pengiriman	Tampil data jenis pengiriman	*Black Box*
	Tambah data jenis pengiriman	*Black Box*
	Edit data jenis pengiriman	*Black Box*
	Hapus data jenis pengiriman	*Black Box*
	Cari data jenis pengiriman	*Black Box*
Data ongkos kirim	Tampil data ongkos kirim	*Black Box*
	Tambah data ongkos kirim	*Black Box*
	Edit data ongkos kirim	*Black Box*
	Hapus data ongkos kirim	*Black Box*
	Cari data ongkos kirim	*Black Box*
Data mata uang	Tampil data mata uang	*Black Box*
	Tambah data mata uang	*Black Box*
	Edit data mata uang	*Black Box*
	Hapus data mata uang	*Black Box*
	Cari data mata uang	*Black Box*

Data retur barang	Tampil data retur barang	*Black Box*
	Ubah status retur barang	*Black Box*
	Ubah status pengiriman retur barang	*Black Box*
Data transaksi	Tampil data pesanan	*Black Box*
	Ubah status pengiriman	*Black Box*
	Ubah status pesanan	*Black Box*
	Ubah status pembayaran	*Black Box*
	Tampil data detail pesanan	*Black Box*
Laporan produk	Tampil laporan produk	*Black Box*
	Cari laporan produk	*Black Box*
	Cetak laporan produk	*Black Box*
Laporan pesanan	Tampil laporan pesanan	*Black Box*
	Cari laporan pesanan	*Black Box*
	Cetak laporan pesanan	*Black Box*
Logout administrator	Hapus *session* administrator	*Black Box*

Tabel 4.6 Skenario Pengujian *Alpha* Bagian Gudang

Kelas Uji	Butir Uji	Jenis Pengujian
Login gudang	Isi data login	*Black Box*
	Verifikasi login	*Black Box*
Edit profil gudang	Tampil profil gudang	*Black Box*
	Edit data profil gudang	*Black Box*
	Simpan perubahan	*Black Box*
Ubah password gudang	Tampil *form* ubah password gudang	*Black Box*
Data kategori	Tampil data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Edit data kategori	*Black Box*
	Hapus data kategori	*Black Box*
	Cari data kategori	*Black Box*
Data subkategori	Tampil data subkategori	*Black Box*
	Tambah data subkategori	*Black Box*
	Edit data subkategori	*Black Box*
	Hapus data subkategori	*Black Box*
	Cari data subkategori	*Black Box*
Data ukuran	Tampil data ukuran	*Black Box*
	Tambah data ukuran	*Black Box*
	Edit data ukuran	*Black Box*
	Hapus data ukuran	*Black Box*
	Cari data ukuran	*Black Box*
Data produk	Tampil data produk	*Black Box*
	Tambah data produk	*Black Box*
	Edit data produk	*Black Box*
	Hapus data produk	*Black Box*
	Cari data produk	*Black Box*
Logout gudang	Hapus *session* gudang	*Black Box*

Tabel 4.7 Skenario Pengujian *Alpha* Pelanggan

Kelas Uji	Butir Uji	Jenis Pengujian
Login pelanggan	Isi data login	*Black Box*
	Verifikasi login	*Black Box*
Edit profil pelanggan	Tampil profil pelanggan	*Black Box*
	Edit data profil pelanggan	*Black Box*
	Simpan perubahan	*Black Box*
Ubah password pelanggan	Tampil *form* ubah password pelanggan	*Black Box*
	Verifikasi username	*Black Box*
	Verifikasi password lama	*Black Box*
	Verifikasi password baru	*Black Box*
Daftar	Tampil *form* pendaftaran	*Black Box*
	Isi form pendaftaran	*Black Box*
	Simpan data pendaftaran	*Black Box*
	Aktifasi ke *email*	*Black Box*
Lupa password	Tampil *form* lupa password	*Black Box*
	Verifikasi *email*	*Black Box*
	Kirim informasi password ke email	*Black Box*
Produk	Tampil data produk	*Black Box*
	Tampil data detail produk	*Black Box*
Tentang kami	Tampil informasi profil perusahaan	*Black Box*
Cara pesan dan bayar	Tampil informasi cara pemesanan dan pembayaran	*Black Box*
History pesanan dan pembayaran	Tampil data pesanan yang pernah dilakukan	*Black Box*
	Tampil *form* pembayaran	*Black Box*
Klaim retur barang	Tampil data retur barang sebelumnya	*Black Box*
	Pilih nomor pesanan yang diretur	
	Tampil data pesanan yang diretur	*Black Box*
	Simpan retur barang	*Black Box*
Beli produk	Pilih ukuran dan warna produk	*Black Box*
Keranjang belanja	Tampil data pesanan yang sementara disimpan di keranjang belanja	*Black Box*
	Ubah jumlah produk yang dipesan	*Black Box*
	Hapus produk yang dipesan	*Black Box*
Isi data pengiriman	Tampil *form* data penerima pesanan	*Black Box*
	Isi jenis pengiriman dan ongkos kirim	*Black Box*
Selesai transaksi	Tampil data pesanan	*Black Box*
Pencarian	Cari produk	*Black Box*
	Tampil produk yang dicari	*Black Box*
Logout	Hapus *session* pelanggan	*Black Box*

4.2.1.2 Kasus dan Hasil Pengujian

1. Pengujian Data Administrator

Tabel 4.8 Pengujian Data Administrator

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data gudang, contoh : nama lengkap : Luvian Yoga Pratama *username* : admingayaku *email* : luvian@indosat.blackberry.com password : admingayaku *level* admin : Administrator	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah administrator	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah administrator	[√] Diterima [ ] Ditolak
Data nama lengkap kosong	Muncul pesan kesalahan "Nama lengkap harus diisi"	Dapat menampilkan pesan kesalahan "Nama lengkap harus diisi"	[√] Diterima [ ] Ditolak
Data *username* kosong	Muncul pesan kesalahan "*Username* harus diisi"	Dapat menampilkan pesan kesalahan "Username harus diisi"	[√] Diterima [ ] Ditolak
Data *email* kosong	Muncul pesan kesalahan "*Email* harus diisi"	Dapat menampilkan pesan kesalahan "*Email* harus diisi"	[√] Diterima [ ] Ditolak
Data password kosong	Muncul pesan kesalahan "Password harus diisi"	Dapat menampilkan pesan kesalahan "Password harus diisi"	[√] Diterima [ ] Ditolak

Data *level* admin kosong	Muncul pesan kesalahan "Level admin harus diisi"	Dapat menampilkan pesan kesalahan "Level admin harus diisi"	[√] Diterima [ ] Ditolak

2. Pengujian Data Bagian Gudang

Tabel 4.9 Pengujian Data Bagian Gudang

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data gudang, contoh : nama lengkap : Abang Gudang *username* : gudanggayaku *email* : luvian@indosat.blackberry.com password : gudanggayaku *level* admin : gudang	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah gudang	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah gudang	[√] Diterima [ ] Ditolak
Data nama lengkap kosong	Muncul pesan kesalahan "Nama lengkap harus diisi"	Dapat menampilkan pesan kesalahan "Nama lengkap harus diisi"	[√] Diterima [ ] Ditolak
Data *username* kosong	Muncul pesan kesalahan "*Username* harus diisi"	Dapat menampilkan pesan kesalahan "Username harus diisi"	[√] Diterima [ ] Ditolak
Data *email* kosong	Muncul pesan kesalahan "*Email* harus diisi"	Dapat menampilkan pesan kesalahan "*Email* harus diisi"	[√] Diterima [ ] Ditolak
Data password kosong	Muncul pesan kesalahan "Password harus diisi"	Dapat menampilkan pesan kesalahan "Password harus diisi"	[√] Diterima [ ] Ditolak

Data *level* admin kosong	Muncul pesan kesalahan "Level admin harus diisi"	Dapat menampilkan pesan kesalahan "Level admin harus diisi"	[√] Diterima [ ] Ditolak

3. Pengujian Cari Data Bagian Gudang

Tabel 4.10 Pengujian Cari Data Bagian Gudang

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data gudang yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Tidak boleh kosong"	Dapat menampilkan pesan "Tidak boleh kosong"	[√] Diterima [ ] Ditolak

4. Pengujian Data Kategori

Tabel 4.11 Pengujian Data Kategori

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data kategori, contoh : nama kategori : tas	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah kategori	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah kategori	[√] Diterima [ ] Ditolak
Data nama kategori kosong	Muncul pesan kesalahan "Nama kategori harus diisi"	Dapat menampilkan pesan kesalahan "Nama kategori harus diisi"	[√] Diterima [ ] Ditolak

5. Pengujian Edit Data Kategori

Tabel 4.12 Pengujian Edit Data Kategori

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama kategori kosong	Muncul pesan kesalahan "Nama kategori harus diisi"	Dapat menampilkan pesan kesalahan "Nama kategori harus diisi"	[√] Diterima [ ] Ditolak

6. Pengujian Cari Data Kategori

Tabel 4.13 Pengujian Cari Data Kategori

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data kategori yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Tidak boleh kosong"	Dapat menampilkan pesan "Tidak boleh kosong"	[√] Diterima [ ] Ditolak

7. Pengujian Data Subkategori

Tabel 4.14 Pengujian Data Subkategori

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data subkategori, contoh : nama kategori : pakaian nama subkategori : kemeja lengan panjang	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah subkategori	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah subkategori	[√] Diterima [ ] Ditolak
Data nama kategori kosong	Muncul pesan kesalahan "Nama kategori harus diisi"	Dapat menampilkan pesan kesalahan "Nama kategori harus diisi"	[√] Diterima [ ] Ditolak

Data nama subkategori kosong	Muncul pesan kesalahan “Nama subkategori harus diisi”	Muncul pesan kesalahan “Nama subkategori harus diisi”	[√] Diterima [ ] Ditolak

8. Pengujian Edit Data Subkategori

Tabel 4.15 Pengujian Edit Data Subkategori

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama kategori kosong	Muncul pesan kesalahan “Nama kategori harus diisi”	Dapat menampilkan pesan kesalahan “Nama kategori harus diisi”	[√] Diterima [ ] Ditolak
Data nama subkategori kosong	Muncul pesan kesalahan “Nama subkategori harus diisi”	Muncul pesan kesalahan “Nama subkategori harus diisi”	[√] Diterima [ ] Ditolak

9. Pengujian Cari Data Subkategori

Tabel 4.16 Pengujian Cari Data Subkategori

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data subkategori yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan “Tidak boleh kosong”	Dapat menampilkan pesan “Tidak boleh kosong”	[√] Diterima [ ] Ditolak

10. Pengujian Data Ukuran

Tabel 4.17 Pengujian Data Ukuran

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data ukuran, contoh : besar ukuran : L	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah ukuran	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah ukuran	[√] Diterima [ ] Ditolak

Data besar ukuran kosong	Muncul pesan kesalahan “Besar ukuran harus diisi”	Dapat menampilkan pesan kesalahan “No ukuran harus diisi”	[√] Diterima [ ] Ditolak

11. Pengujian Edit Data Ukuran

Tabel 4.18 Pengujian Edit Data Ukuran

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data besar ukuran kosong	Muncul pesan kesalahan “Besar ukuran harus diisi”	Dapat menampilkan pesan kesalahan “Besar ukuran harus diisi”	[√] Diterima [ ] Ditolak

12. Pengujian Cari Data Ukuran

Tabel 4.19 Pengujian Cari Data Ukuran

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data ukuran yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan “Tidak boleh kosong”	Dapat menampilkan pesan “Tidak boleh kosong”	[√] Diterima [ ] Ditolak

13. Pengujian Data Produk

Tabel 4.20 Pengujian Data Produk

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data produk, contoh : kategori produk : pakaian subkategori produk : T-Shirt nama produk : T-shirt Garuda harga : 55.000 gambar :	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah produk	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah produk	[√] Diterima [ ] Ditolak

kaos c.jpg deskripsi : T-shirt bergambar Sang Garuda			
Data kategori produk	Muncul pesan kesalahan “Kategori harus diisi”	Dapat menampilkan pesan kesalahan “Kategori harus diisi”	[√] Diterima [ ] Ditolak
Data subkategori produk	Muncul pesan kesalahan “Subkategori harus diisi”	Dapat menampilkan pesan kesalahan “Subkategori harus diisi”	[√] Diterima [ ] Ditolak
Data nama produk	Muncul pesan kesalahan “Nama produk harus diisi”	Dapat menampilkan pesan kesalahan “Nama produk harus diisi”	[√] Diterima [ ] Ditolak
Data harga	Muncul pesan kesalahan “Harga harus diisi”	Dapat menampilkan pesan kesalahan “Harga harus diisi”	[√] Diterima [ ] Ditolak
Data gambar	Muncul pesan kesalahan “Gambar harus diisi”	Dapat menampilkan pesan kesalahan “Gambar harus diisi”	[√] Diterima [ ] Ditolak
Data deskripsi	Muncul pesan kesalahan “Deskripsi harus diisi”	Dapat menampilkan pesan kesalahan “Deskripsi harus diisi”	[√] Diterima [ ] Ditolak

14. Pengujian Edit Data Produk

Tabel 4.21 Pengujian Edit Data Produk

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data kategori produk	Muncul pesan kesalahan "Kategori harus diisi"	Dapat menampilkan pesan kesalahan "Nama warna harus diisi"	[√] Diterima [ ] Ditolak
Data subkategori produk	Muncul pesan kesalahan "Subkategori harus diisi"	Dapat menampilkan pesan kesalahan "Subkategori harus diisi"	[√] Diterima [ ] Ditolak
Data nama produk	Muncul pesan kesalahan "Nama produk harus diisi"	Dapat menampilkan pesan kesalahan "Nama produk harus diisi"	[√] Diterima [ ] Ditolak
Data harga	Muncul pesan kesalahan "Harga harus diisi"	Dapat menampilkan pesan kesalahan "Harga harus diisi"	[√] Diterima [ ] Ditolak
Data gambar	Muncul pesan kesalahan "Gambar harus diisi"	Dapat menampilkan pesan kesalahan "Gambar harus diisi"	[√] Diterima [ ] Ditolak
Data deskripsi	Muncul pesan kesalahan "Deskripsi harus diisi"	Dapat menampilkan pesan kesalahan "Deskripsi harus diisi"	[√] Diterima [ ] Ditolak

15. Pengujian Cari Data Produk

Tabel 4.22 Pengujian Cari Data Produk

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data Produk yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Produk Tidak boleh kosong"	Dapat menampilkan pesan "Produk Tidak boleh kosong"	[√] Diterima [ ] Ditolak

16. Pengujian Data Propinsi

Tabel 4.23 Pengujian Data Propinsi

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data propinsi, contoh : nama propinsi : jawa barat	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah propinsi	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah propinsi	[√] Diterima [ ] Ditolak
Data nama propinsi	Muncul pesan kesalahan "Nama propinsi harus diisi"	Dapat menampilkan pesan kesalahan "Nama propinsi harus diisi"	[√] Diterima [ ] Ditolak

17. Pengujian Edit Data Propinsi

Tabel 4.24 Pengujian Edit Data Propinsi

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama propinsi	Muncul pesan kesalahan "Nama propinsi harus diisi"	Dapat menampilkan pesan kesalahan "Nama propinsi harus diisi"	[√] Diterima [ ] Ditolak

18. Pengujian Cari Data Propinsi

Tabel 4.25 Pengujian Cari Data Propinsi

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data Propinsi yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Propinsi Tidak boleh kosong"	Dapat menampilkan pesan "Propinsi Tidak boleh kosong"	[√] Diterima [ ] Ditolak

19. Pengujian Data Kabkota

Tabel 4.26 Pengujian Data Kabkota

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data kab/kota, contoh : nama propinsi : jawa barat nama kab/kota : garut	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah kab/kota	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah kab/kota	[√] Diterima [ ] Ditolak
Data nama propinsi	Muncul pesan kesalahan "Nama propinsi harus diisi"	Dapat menampilkan pesan kesalahan "Nama propinsi harus diisi"	[√] Diterima [ ] Ditolak
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama kab/kota	Muncul pesan kesalahan "Nama kab/kota harus diisi"	Dapat menampilkan pesan kesalahan "Nama kab/kota harus diisi"	[√] Diterima [ ] Ditolak

20. Pengujian Edit Data Kabkota

Tabel 4.27 Pengujian Edit Data Kabkota

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama propinsi	Muncul pesan kesalahan "Nama propinsi harus diisi"	Dapat menampilkan pesan kesalahan "Nama propinsi harus diisi"	[√] Diterima [ ] Ditolak
Data nama kab/kota	Muncul pesan kesalahan "Nama kab/kota harus diisi"	Dapat menampilkan pesan kesalahan "Nama kab/kota harus diisi"	[√] Diterima [ ] Ditolak

21. Pengujian Cari Data Kabkota

Tabel 4.28 Pengujian Cari Data Kabkota

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data kategori yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Kabkota Tidak boleh kosong"	Dapat menampilkan pesan "Kabkota Tidak boleh kosong"	[√] Diterima [ ] Ditolak

22. Pengujian Data Jenis Pengiriman

Tabel 4.29 Pengujian Data Jenis Pengiriman

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data jenis pengiriman, contoh : nama jenis pengiriman : JNE Reguler	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah jenis pengiriman	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah jenis pengiriman	[√] Diterima [ ] Ditolak
Data nama jenis pengiriman	Muncul pesan kesalahan "Nama jenis pengiriman harus diisi"	Dapat menampilkan pesan kesalahan "Nama jenis pengiriman harus diisi"	[√] Diterima [ ] Ditolak

23. Pengujian Edit Data Jenis Pengiriman

Tabel 4.30 Pengujian Edit Data Jenis Pengiriman

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama jenis pengiriman	Muncul pesan kesalahan "Nama jenis pengiriman harus diisi"	Dapat menampilkan pesan kesalahan "Nama jenis pengiriman harus diisi"	[√] Diterima [ ] Ditolak

24. Pengujian Cari Jenis Pengiriman

Tabel 4.31 Pengujian Cari Data Jenis Pengiriman

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data Jenis Pengiriman yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Jenis Pengiriman Tidak boleh kosong"	Dapat menampilkan pesan "Jenis Pengiriman Tidak boleh kosong"	[√] Diterima [ ] Ditolak

25. Pengujian Data Ongkos Kirim

Tabel 4.32 Pengujian Data Ongkos Kirim

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data ongkos kirim, contoh : nama jenis pengiriman : JNE Reguler kab/kota : garut biaya ongkos : 18.000	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah ongkos kirim	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah ongkos kirim	[√] Diterima [ ] Ditolak
Data nama jenis pengiriman	Muncul pesan kesalahan "Nama jenis pengiriman	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

	harus diisi”	“Nama jenis pengiriman harus diisi”	
Data kab/kota	Muncul pesan kesalahan “Kab/kota harus diisi”	Dapat menampilkan pesan kesalahan “Kab/kota harus diisi”	[√] Diterima [ ] Ditolak
Data biaya ongkos	Muncul pesan kesalahan “Biaya ongkos harus diisi”	Dapat menampilkan pesan kesalahan “Biaya ongkos harus diisi”	[√] Diterima [ ] Ditolak

26. Pengujian Edit Data Ongkos Kirim

Tabel 4.33 Pengujian Edit Data Ongkos Kirim

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama jenis pengiriman	Muncul pesan kesalahan “Nama jenis pengiriman harus diisi”	Dapat menampilkan pesan kesalahan “Nama jenis pengiriman harus diisi”	[√] Diterima [ ] Ditolak
Data kab/kota	Muncul pesan kesalahan “Kab/kota harus diisi”	Dapat menampilkan pesan kesalahan “Kab/kota harus diisi”	[√] Diterima [ ] Ditolak
Data biaya ongkos	Muncul pesan kesalahan “Biaya ongkos harus diisi”	Dapat menampilkan pesan kesalahan “Biaya ongkos harus diisi”	[√] Diterima [ ] Ditolak

27. Pengujian Cari Data Ongkos Kirim

Tabel 4.34 Pengujian Cari Data Ongkos Kirim

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data Ongkos Kirim yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Ongkos Kirim Tidak boleh kosong"	Dapat menampilkan pesan "Ongkos Kirim Tidak boleh kosong"	[√] Diterima [ ] Ditolak

28. Pengujian Data Mata Uang

Tabel 4.35 Pengujian Data Matauang

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data mata uang, contoh : kode : USD	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah mata uang	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah mata uang	[√] Diterima [ ] Ditolak
Data Kode	Muncul pesan kesalahan "Kode harus diisi"	Dapat menampilkan pesan kesalahan "Kode harus diisi"	[√] Diterima [ ] Ditolak

29. Pengujian Edit Data Mata Uang

Tabel 4.36 Pengujian Edit Data Mata uang

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data Kode	Muncul pesan kesalahan "Kode harus diisi"	Dapat menampilkan pesan kesalahan "Kode harus diisi"	[√] Diterima [ ] Ditolak

30. Pengujian Cari Data Mata Uang

Tabel 4.37 Pengujian Cari Data Mata Uang

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data kategori yang dicari	Jika semua *form* terisi, maka data yang dicari akan tampil	Jika semua *form* terisi, maka data yang dicari akan tampil	[√] Diterima [ ] Ditolak
Salah satu *form* pencarian kosong	Muncul pesan "Tidak boleh kosong"	Dapat menampilkan pesan "Tidak boleh kosong"	[√] Diterima [ ] Ditolak

31. Pengujian Data Pelanggan

Tabel 4.38 Pengujian Data Pelanggan

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data pelanggan, contoh : nama lengkap : Luvian Yoga Pratama *username* : luvanyoga *email* : luvian@indosat.blackberry.com password : luvian ulang password : luvian alamat : Jl. Encep Kartawiria no.25 propinsi : Jawa Barat kab/kota : Cimahi kodepos : 40512 nomor *handphone* : 08562010299	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftaran	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftaran	[√] Diterima [ ] Ditolak
Data nama lengkap kosong	Muncul pesan kesalahan "Nama lengkap harus diisi"	Dapat menampilkan pesan kesalahan "Nama	[√] Diterima [ ] Ditolak

		lengkap harus diisi”	
Data *username* kosong	Muncul pesan kesalahan “*Username* harus diisi”	Dapat menampilkan pesan kesalahan “*Username* harus diisi”	[√] Diterima [ ] Ditolak
Data *email* kosong	Muncul pesan kesalahan “*Email* harus diisi”	Dapat menampilkan pesan kesalahan “*Email* harus diisi”	[√] Diterima [ ] Ditolak
Data password kosong	Muncul pesan kesalahan “Password harus diisi”	Dapat menampilkan pesan kesalahan “Password harus diisi”	[√] Diterima [ ] Ditolak
Data ulang password	Muncul pesan kesalahan “Ulang password harus diisi”	Dapat menampilkan pesan kesalahan “Ulang password harus diisi”	[√] Diterima [ ] Ditolak
Data alamat kosong	Muncul pesan kesalahan “Alamat harus diisi”	Dapat menampilkan pesan kesalahan “Alamat harus diisi”	[√] Diterima [ ] Ditolak
Data propinsi kosong	Muncul pesan kesalahan “Propinsi harus diisi”	Dapat menampilkan pesan kesalahan “Propinsi harus diisi”	[√] Diterima [ ] Ditolak
Data kab/kota kosong	Muncul pesan kesalahan “Kab/kota harus diisi”	Dapat menampilkan pesan kesalahan “Kab/kota harus diisi”	[√] Diterima [ ] Ditolak
Data kodepos kosong	Muncul pesan	Dapat	[√] Diterima

	kesalahan "Kodepos harus diisi"	menampilkan pesan kesalahan "Kodepos harus diisi"	[ ] Ditolak
Data nomor *handphone* kosong	Muncul pesan kesalahan "Nomor *handphone* harus diisi"	Dapat menampilkan pesan kesalahan "Nomor *handphone* harus diisi"	[√] Diterima [ ] Ditolak

32. Pengujian Data Login Pelanggan

Tabel 4.39 Pengujian Data Login Pelanggan

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi *email* : luvian@indosat.blackberry.com	Dapat terisi pada *textbox email*	Dapat mengisi *field email* dengan benar	[√] Diterima [ ] Ditolak
Isi password : luvian	Dapat terisi pada *textbox* password	Dapat mengisi *field* password dengan benar	[√] Diterima [ ] Ditolak
Data *email* kosong	Tidak dapat menampilkan menu pelanggan dan menampilkan pesan "Data Login Invalid"	Muncul pesan kesalahan "Data Login Invalid"	[√] Diterima [ ] Ditolak
Data password kosong	Tidak dapat menampilkan menu pelanggan dan menampilkan pesan "Data Login Invalid"	Muncul pesan kesalahan "Data Login Invalid"	[√] Diterima [ ] Ditolak
Data *email* salah	Tidak dapat menampilkan menu pelanggan dan menampilkan pesan "*Email* salah"	Muncul pesan kesalahan "*Email* salah"	[√] Diterima [ ] Ditolak
Data password salah	Tidak dapat menampilkan menu	Muncul pesan	[√] Diterima [ ] Ditolak

	pelanggan dan menampilkan pesan "Password salah"	kesalahan "Password salah"	

33. Pengujian Lupa Data Password Pelanggan

Tabel 4.40 Pengujian Lupa Data Password Pelanggan

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi *email* : luvian@indosat.blackberry.com	Dapat terisi pada *textbox username*	Dapat mengisi *field username* dengan benar	[√] Diterima [ ] Ditolak
Data *email* salah	Tidak dapat menyimpan data password baru menampilkan pesan "*Email* salah"	Muncul pesan kesalahan "*Email* salah"	[√] Diterima [ ] Ditolak

34. Pengujian Data Profil Pelanggan

Tabel 4.41 Pengujian Data Profil Pelanggan

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data *profil* pelanggan kosong	Data profil pelanggan gagal dirubah dan menampilkan pesan kesalahan "Data pelanggan gagal dirubah"	Data profil pelanggan gagal dirubah dan menampilkan pesan kesalahan "Data pelanggan gagal dirubah"	[√] Diterima [ ] Ditolak

35. Pengujian Ubah Data Password Pelanggan

Tabel 4.42 Pengujian Ubah Data Password Pelanggan

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data *usernmame* salah	Menampilkan pesan kesalahan "Data *username* anda salah"	Dapat menampilkan pesan kesalahan "Data *username* anda salah"	[√] Diterima [ ] Ditolak
Data password lama salah	Menampilkan pesan kesalahan "Data password	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

	lama anda salah"	"Data password lama anda salah"	
Data *username* kosong	Muncul pesan kesalahan "*Username* harus diisi"	Dapat menampilkan pesan kesalahan "*Username* harus diisi"	[√] Diterima [ ] Ditolak
Data password lama kosong	Muncul pesan kesalahan "Password lama harus diisi"	Dapat menampilkan pesan kesalahan "Password harus diisi"	[√] Diterima [ ] Ditolak
Data password baru kosong	Muncul pesan kesalahan "Password baru harus diisi"	Dapat menampilkan pesan kesalahan "Password baru harus diisi"	[√] Diterima [ ] Ditolak

36. Pengujian Data Keranjang Belanja

Tabel 4.43 Pengujian Data Keranjang Belanja

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data *Quantity*	Tercantum pada *textbox* data yang akan diubah	Dapat mengisi data jumlah dengan benar	[√] Diterima [ ] Ditolak
Data *quantity* produk bukan diisi angka	Menampilkan pesan kesalahan "*Quantity* hanya boleh diisi angka"	Dapat menampilkan pesan kesalahan "*Quantity* hanya boleh diisi angka"	[√] Diterima [ ] Ditolak
Data *quantity* produk melebihi stok	Menampilkan pesan kesalahan "Maaf produk yang dibeli melebihi stok yang ada"	Dapat menampilkan pesan kesalahan "Maaf produk yang dibeli melebihi stok yang ada"	[√] Diterima [ ] Ditolak

37. Pengujian Data Pengiriman

Tabel 4.44 Pengujian Data Pengiriman

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data penerima pesanan contoh : nama penerima : Andriana propinsi : jawa barat kab/kota : garut alamat : jl. suryasumantri no.1 nomor telepon penerima : 08562010299 jenis pengiriman : JNE Reguler	Jika salah satu *field* tidak terisi atau salah dalam pengisian maka data pesanan tidak akan tersimpan	Jika salah satu *field* tidak terisi atau salah dalam pengisian maka data pesanan tidak akan tersimpan	[√] Diterima [ ] Ditolak

38. Pengujian Data Pembayaran

Tabel 4.45 Pengujian Data Pembayaran

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Isi data pembayaran contoh : transfer dari bank : Bank BCA jenis transfer : rekening nomor rekening : 0131444304 tujuan nomor rekening : BCA No. Rekening 280-354-1302 atas nama : Andriana uang yang di transfer : 100.000	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* tersebut harus terisi.	Dapat memunculkan jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* tersebut harus terisi.	[√] Diterima [ ] Ditolak

Data nomor rekening kosong	Muncul pesan kesalahan "Nomor rekening harus diisi"	Dapat menampilkan pesan kesalahan "Nomor rekening harus diisi"	[√] Diterima [ ] Ditolak
Data atas nama kosong	Muncul pesan kesalahan "Atas nama harus diisi"	Dapat menampilkan pesan kesalahan "Atas nama harus diisi"	[√] Diterima [ ] Ditolak
Data uang yang di transfer kosong	Muncul pesan kesalahan "Uang yang ditransfer harus diisi"	Dapat menampilkan pesan kesalahan "Uang yang ditransfer harus diisi"	[√] Diterima [ ] Ditolak

39. Pengujian Data Retur

Tabel 4.46 Pengujian Data Retur

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data *quantity* retur melebihi *quantity* pesanan	Menampilkan pesan kesalahan "Maaf produk yang diretur PR0009 melebihi *quantity* pesanan"	Dapat menampilkan pesan kesalahan "Maaf produk yang diretur melebihi *quantity* pesanan"	[√] Diterima [ ] Ditolak

4.2.1.3 Kesimpulan Pengujian Alpha

Berdasarkan hasil pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan aplikasi *e-commerce* berbasis web di Toko Gayaku bebas dari kesalahan yang disebabkan oleh data kosong atau data sama dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.2.2 Pengujian *Beta*

4.2.2.1 Skenario Pengujian *Beta*

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

1. Toko Gayaku

   Penelitian di lingkungan Toko Gayaku dilakukan agar 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*.

Adapun beberapa poin yang diteliti adalah :

1. Pengetahuan mengenai *e-commerce*.
2. Antarmuka *website* Toko Gayaku.
3. Informasi yang didapat diperoleh dari *website*.
4. Proses Pemesanan.
5. Proses pengolahan data dan laporan.

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

$$\mathbf{Y = P/Q * 100\%}$$

Keterangan :

P = Banyaknya jawaban responden tiap soal.

Q = Jumlah responden

Y = Nilai prosentase

Berikut ini adalah hasil prosentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuisioner ini diujikan kepada 20 orang.

1. Apakah Anda mengetahui tentang *e-commerce*?

Tabel 4.47 Pertanyaan mengenai *e-commerce*

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
1	1	Sangat Tahu	5	25%
	2	Tahu	13	65%
	3	Cukup Tahu	2	10%
	4	Kurang Tahu	0	0%
	5	Tidak Tahu	0	0%
			20	100%

Berdasarkan hasil prosentase pada tabel 4.47 maka dapat disimpulkan bahwa sebanyak 5 orang atau 25% menyatakan sangat tahu, 13 orang atau 65% menyatakan tahu, 2 orang atau 10% menyatakan cukup tahu. Sehingga secara keseluruhan responden sudah mengenal *e-commerce.*

2. Apakah menurut Anda tampilan *website* Toko Gayaku menarik dan mudah dipahami?

Tabel 4.48 Pertanyaan mengenai tampilan *website*

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
2	1	Sangat Menarik	4	20%
	2	Menarik	12	60%
	3	Cukup Menarik	4	20%
	4	Kurang Menarik	0	0%
	5	Tidak Menarik	0	0%
			20	100%

Berdasarkan hasil prosentase pada tabel 4.48 maka dapat disimpulkan bahwa sebanyak 4 orang atau 20% menyatakan sangat menarik, 12 orang atau 60% menyatakan menarik, 4 orang atau 20% menyatakan cukup menarik mengenai tampilan antarmuka *website* Toko Gayaku.

3. Apakah Anda setuju, dengan adanya aplikasi *e-commerce* ini dapat memudahkan pelanggan baik dalam maupun luar kota Bandung?

Tabel 4.49 Pertanyaan mengenai kemudahan penggunaan *website*

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
3	1	Sangat Setuju	4	20%
	2	Setuju	12	60%
	3	Cukup Setuju	4	20%
	4	Kurang Setuju	0	0%
	5	Tidak Setuju	0	0%
			20	100%

Berdasarkan hasil prosentase pada tabel 4.49 maka dapat disimpulkan bahwa sebanyak 4 orang atau 20% mengatakan sangat setuju, 12 orang atau 60% mengatakan setuju, 4 orang atau 20% mengatakan cukup setuju. Sehingga secara keseluruhan mengatakan setuju bahwa *website* ini membantu pelanggan yang berada di kota Bandung atau yang berada di luar kota Bandung dalam melakukan pemesanan.

4. Apakah Anda setuju dengan adanya aplikasi *e-commerce* ini, dapat dijadikan suatu media untuk mempermudah penjualan di Toko Gayaku?

Tabel 4.50 Pertanyaan mengenai pendapat adanya *website* ini

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
4	1	Sangat Setuju	7	35%
	2	Setuju	10	50%
	3	Cukup Setuju	3	15%
	4	Kurang Setuju	0	0%
	5	Tidak Setuju	0	0%
			20	100%

Berdasarkan hasil prosentase pada tabel 4.50maka dapat disimpulkan bahwa sebanyak 7 orang atau 35% menyatakan sangat setuju, 10 orang atau 50% menyatakan setuju, 3 orang atau 15% menyatakan cukup setuju. Sehingga secara keseluruhan aplikasi tersebut dapat membantu penjualan di Toko Gayaku.

5. Apakah Anda setuju dengan aplikasi *e-commerce* ini dapat mempermudah dalam proses pengolahan data dan pembuatan laporan di Toko Gayaku?

Tabel 4.51 Pertanyaan mengenai kemudahan pengolahan data

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
5	1	Sangat Setuju	7	35%
	2	Setuju	11	55%
	3	Cukup Setuju	2	10%
	4	Kurang Setuju	0	0%
	5	Tidak Setuju	0	0%
			20	100%

Berdasarkan hasil prosentase pada tabel 4.51 maka dapat disimpulkan bahwa sebanyak 7 orang atau 35% mengatakan sangat setuju, 11 orang atau 55% mengatakan setuju, 2 orang atau 10% mengatakan cukup setuju. Sehingga secara keseluruhan aplikasi tersebut mudah untuk digunakan dalam melakukan proses pengolahan data dan pembuatan laporan.

6. Apakah Anda setuju proses pemesanan dan pembayaran pada *website* Toko Gayaku sudah memudahkan Anda?

Tabel 4.52 Pertanyaan mengenai proses transaksi

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
6	1	Sangat Setuju	2	10%
	2	Setuju	9	45%
	3	Cukup Setuju	9	45%
	4	Kurang Setuju	0	0%
	5	Tidak Setuju	0	0%
			20	100%

Berdasarkan hasil prosentase pada tabel 4.52 maka dapat disimpulkan bahwa sebanyak 2 orang atau 10% menyatakan sangat setuju, 9 orang atau 45% menyatakan setuju, 9 orang atau 45% menyatakan cukup setuju bahwa proses pemesanan dan pembayaran di *website* Toko Gayaku sudah cukup memudahkan pelanggan.

4.2.2.2 Kesimpulan Hasil Pengujian *Beta*

Berdasarkan pengujian *beta* yang dilakukan di lingkungan Toko Gayaku dan di lingkungan masyarakat umum, maka dapat diambil kesimpulan bahwa :

1. Aplikasi ini dapat memberi kemudahan pada pengolahan data serta dalam pembuatan laporan.
2. Tampilan *website* Toko Gayaku cukup menarik dan mudah untuk dipahami baik dari situs *backend* maupun *frontend*.
3. *Website* Toko Gayaku cukup membantu pihak Toko dalam melakukan penjualan produknya.

4. Proses pemesanan dan pembayaran pada *website* Toko Gayaku cukup memudahkan pelanggan.
5. *Website* Toko Gayaku cukup membantu para pelanggan yang berada di kota Bandung maupun yang berada di luar kota bandung dalam melakukan pemesanan.