BAB IV

IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Tahap implementasi merupakan tahap penciptaan perangkat lunak, tahap lanjutan dari 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 *website E-Commerce* pada Riri Collection adalah tercantum pada tabel 4.1 sebagai berikut.

Tabel 4.1 Perangkat keras yang digunakan

Perangkat	Spesifikasi
Processor	Dual Core @ 2.8 GHz
RAM	1 GHz
Harddisk	320 Gb
Monitor	Resolusi 1280 x 800
VGA	128 Mb

4.1.2 Perangkat lunak yang digunakan

Spesifikasi perangkat lunak yang digunakan dalam pembangunan *website E-Commerce* pada Riri Collection adalah sebagai berikut.

Tabel 4.2 Perangkat Lunak yang digunakan

Perangkat	Spesifikasi
Processor	Core 2 Duo @ 2.1 GHz.
RAM	2 GHz
Harddisk	320 Gb
Monitor	Resolusi 1280 x 800
VGA	128 Mb

4.1.3 Implementasi database

1. Tabel Admin

```
CREATE TABLE IF NOT EXISTS `admin` (
  `id_admin` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `last_visit` varchar(50) NOT NULL,
  PRIMARY KEY (`id_admin`),
  UNIQUE KEY `email` (`email`),
  KEY `username` (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Tabel Cart

```
CREATE TABLE IF NOT EXISTS `cart` (
        `id_cart` int(11) NOT NULL auto_increment,
        `id_produk` int(11) NOT NULL,
        `jumlah` int(11) NOT NULL,
        `tanggal` date NOT NULL,
        `jam` varchar(100) NOT NULL,
        `id_member` int(11) NOT NULL,
        PRIMARY KEY  (`id_cart`),
        KEY `id_produk` (`id_produk`),
        KEY `id_member` (`id_member`),
CONSTRAINT `cart_ibfk_1` FOREIGN KEY (`id_produk`)
```

```
REFERENCES `produk` (`id_produk`) ON DELETE CASCADE ON UPDATE CASCADE,
 CONSTRAINT `cart_ibfk_2` FOREIGN KEY (`id_member`) REFERENCES `member` (`id_member`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

3. Tabel Detail Pesanan

```
CREATE TABLE IF NOT EXISTS `detail_pesanan` (
        `id_detail_pesanan` int(11) NOT NULL auto_increment,
        `id_pesanan` int(11) NOT NULL,
        `id_produk` int(11) NOT NULL,
        `jumlah` int(11) NOT NULL,
        `diskon` int(3) NOT NULL,
        `harga` int(20) NOT NULL,
        `nama_produk` varchar(50) NOT NULL,
        `kategori` int(11) NOT NULL,
        `berat` float NOT NULL,
        `panjang` int(11) NOT NULL,
        `lebar` int(11) NOT NULL,
        `tinggi` int(11) NOT NULL,
        `diameter` int(11) NOT NULL,
        PRIMARY KEY  (`id_detail_pesanan`),
        KEY `id_pesanan` (`id_pesanan`),
        KEY `id_produk` (`id_produk`),
CONSTRAINT `detail_pesanan_ibfk_3` FOREIGN KEY (`id_pesanan`) REFERENCES `pesanan` (`id_pesanan`),
CONSTRAINT `detail_pesanan_ibfk_2` FOREIGN KEY (`id_produk`) REFERENCES `produk` (`id_produk`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

4. Tabel Gambar

```
CREATE TABLE IF NOT EXISTS `gambar` (
          `id_gambar` int(11) NOT NULL AUTO_INCREMENT,
          `id_produk` int(11) NOT NULL,
          `gambar` varchar(50) NOT NULL,
          PRIMARY KEY (`id_gambar`),
          KEY `id_produk` (`id_produk`),
        CONSTRAINT `gambar_ibfk_1` FOREIGN KEY (`id_produk`)
        REFERENCES `produk` (`id_produk`) ON DELETE CASCADE
        ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

5. Tabel Jasa

```
CREATE TABLE IF NOT EXISTS `jasa` (
          `id_jasa` int(5) NOT NULL AUTO_INCREMENT,
          `nama_jasa` varchar(100) NOT NULL,
          PRIMARY KEY (`id_jasa`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

6. Tabel Kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL AUTO_INCREMENT,
  `nama_kategori` varchar(50) NOT NULL,
  PRIMARY KEY (`id_kategori`),
  UNIQUE KEY `nama_kategori` (`nama_kategori`)
  ) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

7. Tabel Konfirmasi

```
CREATE TABLE IF NOT EXISTS `konfirmasi` (
          `id_pesanan` int(11) NOT NULL,
          `tanggal_konfirmasi` date NOT NULL,
          `jenis_konfirmasi` varchar(10) COLLATE latin1_general_ci
        NOT NULL,
          `jumlah_konfirmasi` float NOT NULL,
          `no_transaksi` varchar(50) COLLATE latin1_general_ci NOT
        NULL,
          `bank_pengirim` varchar(50) COLLATE latin1_general_ci NOT
        NULL,
          `rekening_pengirim` int(15) NOT NULL,
```

```
        `atas_nama_pengirim` varchar(50) COLLATE latin1_general_ci
      NOT NULL,
        `bank_tujuan` varchar(50) COLLATE latin1_general_ci NOT
      NULL,
        `cabang_tujuan` varchar(50) COLLATE latin1_general_ci NOT
      NULL,
        `rekening_tujuan` int(15) NOT NULL,
        `atas_nama_tujuan` varchar(50) COLLATE latin1_general_ci
      NOT NULL,
        KEY `id_pesanan` (`id_pesanan`),
      CONSTRAINT `konfirmasi_ibfk_1` FOREIGN KEY
      (`id_pesanan`) REFERENCES `pesanan` (`id_pesanan`) ON
      DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1
COLLATE=latin1_general_ci;
```

8. Tabel Kota

```
CREATE TABLE IF NOT EXISTS `kota` (
        `id_kab_kota` int(5) NOT NULL AUTO_INCREMENT,
        `id_provinsi` int(5) NOT NULL,
        `kab_kota` varchar(100) NOT NULL,
        PRIMARY KEY (`id_kab_kota`),
        UNIQUE KEY `kab_kota` (`kab_kota`),
        KEY `id_provinsi` (`id_provinsi`),
      CONSTRAINT `kota_ibfk_1` FOREIGN KEY (`id_provinsi`)
      REFERENCES `provinsi` (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

9. Tabel Member

```
CREATE TABLE IF NOT EXISTS `member` (
        `id_member` int(11) NOT NULL auto_increment,
        `nama` varchar(50) NOT NULL,
        `username` varchar(50) NOT NULL,
        `password` varchar(50) NOT NULL,
        `email` varchar(50) NOT NULL,
        `alamat` text NOT NULL,
        `kota` int(11) NOT NULL,
        `provinsi` int(11) NOT NULL,
        `kode_pos` int(5) NOT NULL,
        `telepon` varchar(15) NOT NULL,
        `status` int(11) NOT NULL,
        `last_visit` date NOT NULL,
```

```
        PRIMARY KEY  (`id_member`),
        UNIQUE KEY `email` (`email`),
        KEY `kota` (`kota`),
      CONSTRAINT `member_ibfk_1` FOREIGN KEY (`kota`)
      REFERENCES `kota` (`id_kab_kota`) ON DELETE CASCADE
      ON UPDATE CASCADE
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

10. Tabel Paket Jasa

```
CREATE TABLE IF NOT EXISTS `paket_jasa` (
        `id_paket_jasa` int(5) NOT NULL auto_increment,
        `id_jasa` int(5) NOT NULL,
        `paket_jasa` varchar(50) character set latin1 NOT NULL,
        PRIMARY KEY  (`id_paket_jasa`),
        KEY `id_jasa` (`id_jasa`),
CONSTRAINT `paket_jasa_ibfk_1` FOREIGN KEY (`id_jasa`)
REFERENCES `jasa` (`id_jasa`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

11. Tabel Petunjuk

```
CREATE TABLE IF NOT EXISTS `petunjuk` (
        `id_petunjuk` int(5) NOT NULL AUTO_INCREMENT,
        `petunjuk` varchar(50) NOT NULL,
        `isi_petunjuk` text NOT NULL,
        PRIMARY KEY (`id_petunjuk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

12. Tabel Provinsi

```
CREATE TABLE IF NOT EXISTS `provinsi` (
        `id_provinsi` int(5) NOT NULL AUTO_INCREMENT,
        `provinsi` varchar(100) NOT NULL,
        PRIMARY KEY (`id_provinsi`),
        UNIQUE KEY `provinsi` (`provinsi`),
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

13. Tabel Ongkos Kirim

```
CREATE TABLE IF NOT EXISTS `ongkos_kirim` (
        `id_ongkos` int(11) NOT NULL AUTO_INCREMENT,
        `jasa_kirim` int(11) NOT NULL,
        `provinsi_kirim` int(11) NOT NULL,
        `kab_kota_kirim` int(11) NOT NULL,
        `ongkos_kirim` int(15) NOT NULL,
        PRIMARY KEY (`id_ongkos`),
        KEY `jasa_kirim` (`jasa_kirim`),
        KEY `kab_kota_kirim` (`kab_kota_kirim`),
CONSTRAINT `ongkos_kirim_ibfk_2` FOREIGN KEY (`kab_kota_kirim`) REFERENCES `kota` (`id_kab_kota`),
CONSTRAINT `ongkos_kirim_ibfk_3` FOREIGN KEY (`jasa_kirim`) REFERENCES `paket_jasa` (`id_paket_jasa`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

14. Tabel Detail Retur

```
CREATE TABLE IF NOT EXISTS `detail_retur` (
  `id_detail_retur` int(11) NOT NULL,
  `id_detail_pesanan` int(11) NOT NULL,
  `id_retur` int(11) NOT NULL,
  `alasan` varchar(50) NOT NULL,
  `jumlah_retur` int(11) NOT NULL,
  `gambar_retur` varchar(50) NOT NULL,
  KEY `id_detail_pesanan` (`id_detail_pesanan`),
  KEY `id_retur` (`id_retur`),
CONSTRAINT `detail_retur_ibfk_2` FOREIGN KEY (`id_retur`) REFERENCES `retur` (`id_retur`),
CONSTRAINT `detail_retur_ibfk_1` FOREIGN KEY (`id_detail_pesanan`) REFERENCES `detail_pesanan` (`id_detail_pesanan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

15. Tabel Pemesanan

```
CREATE TABLE IF NOT EXISTS `pesanan` (
  `id_pesanan` int(11) NOT NULL AUTO_INCREMENT,
  `member` int(11) NOT NULL,
  `nama_pemesan` varchar(50) NOT NULL,
  `alamat_pemesan` text NOT NULL,
  `telepon_pemesan` varchar(15) NOT NULL,
  `email_pemesan` varchar(50) NOT NULL,
  `kota_pemesan` int(11) NOT NULL,
  `provinsi_pemesan` int(11) NOT NULL,
  `kode_pos_pemesan` varchar(50) NOT NULL,
  `status_pesanan` varchar(20) NOT NULL,
  `tanggal` date NOT NULL,
  `jam` time NOT NULL,
  `jasa` int(11) NOT NULL,
  `ongkos` int(15) NOT NULL,
  `total` int(15) NOT NULL,
  `no_track` varchar(20) DEFAULT NULL,
  `batas_konfirmasi` date NOT NULL,
  `batas_retur` date DEFAULT NULL,
  `nama_penerima` varchar(20) DEFAULT NULL,
  `tgl_terima` datetime DEFAULT NULL,
  `penyetuju` varchar(50) NOT NULL,
  PRIMARY KEY (`id_pesanan`),
  UNIQUE KEY `no_track` (`no_track`),
  KEY `kota_pemesan` (`kota_pemesan`),
  KEY `member` (`member`),
  KEY `ongkos` (`ongkos`),
  KEY `penyetuju` (`penyetuju`),
 CONSTRAINT `pesanan_ibfk_1` FOREIGN KEY (`kota_pemesan`) REFERENCES `kota` (`id_kab_kota`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `pesanan_ibfk_2` FOREIGN KEY (`member`) REFERENCES `member` (`id_member`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `pesanan_ibfk_3` FOREIGN KEY (`ongkos`) REFERENCES `ongkos_kirim` (`id_ongkos`) ON DELETE CASCADE ON UPDATE CASCADE,
CONSTRAINT `pesanan_ibfk_4` FOREIGN KEY (`penyetuju`) REFERENCES `admin` (`username`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

16. Tabel Produk

```
CREATE TABLE IF NOT EXISTS `produk` (
          `id_produk` int(11) NOT NULL AUTO_INCREMENT,
          `nama_produk` varchar(50) NOT NULL,
          `harga` int(15) NOT NULL,
          `deskripsi` text NOT NULL,
          `kategori_produk` int(11) NOT NULL,
          `gambar_utama` varchar(50) NOT NULL,
          `tanggal_masuk` date NOT NULL,
          `diskon` int(11) NOT NULL,
          `berat` float NOT NULL,
          `panjang` int(11) DEFAULT NULL,
          `lebar` int(11) DEFAULT NULL,
          `tinggi` int(11) DEFAULT NULL,
          `diameter` int(11) DEFAULT NULL,
          `terjual` int(11) NOT NULL,
          `viewed` int(11) NOT NULL,
          PRIMARY KEY (`id_produk`),
          UNIQUE KEY `nama_produk` (`nama_produk`),
          KEY `kategori_produk` (`kategori_produk`),
         CONSTRAINT `produk_ibfk_1` FOREIGN KEY
        (`kategori_produk`) REFERENCES `kategori` (`id_kategori`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

17. Tabel Rekening

```
CREATE TABLE IF NOT EXISTS `rekening` (
          `id_rekening` int(5) NOT NULL AUTO_INCREMENT,
          `bank` varchar(50) COLLATE latin1_general_ci NOT NULL,
          `rekening` varchar(50) COLLATE latin1_general_ci NOT NULL,
          `atas_nama` varchar(50) COLLATE latin1_general_ci NOT
        NULL,
          `cabang` varchar(50) COLLATE latin1_general_ci NOT NULL,
          PRIMARY KEY (`id_rekening`)
)       ENGINE=InnoDB          DEFAULT          CHARSET=latin1
COLLATE=latin1_general_ci;
```

18. Tabel Retur

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(11) NOT NULL AUTO_INCREMENT,
  `id_detail_pesanan` int(11) NOT NULL,
  `tanggal_retur` date NOT NULL,
  `alasan` text NOT NULL,
  `id_member` int(11) NOT NULL,
  `status_retur` varchar(30) NOT NULL,
  `jumlah_retur` int(11) NOT NULL,
  `no_resi` int(11) NOT NULL,
  `gambar_retur` varchar(100) NOT NULL,
  PRIMARY KEY (`id_retur`),
  KEY `id_detail_pesanan` (`id_detail_pesanan`),
  KEY `id_member` (`id_member`),
  CONSTRAINT `retur_ibfk_1` FOREIGN KEY (`id_detail_pesanan`) REFERENCES `detail_pesanan` (`id_pesanan`) ON DELETE CASCADE ON UPDATE CASCADE,
 CONSTRAINT `retur_ibfk_2` FOREIGN KEY (`id_member`) REFERENCES `member` (`id_member`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

19. Tabel Stok

```
CREATE TABLE IF NOT EXISTS `stok` (
        `id_produk` int(11) NOT NULL,
        `stok` int(11) NOT NULL,
        KEY `id_produk` (`id_produk`),
CONSTRAINT `stok_ibfk_2` FOREIGN KEY (`id_produk`) REFERENCES `produk` (`id_produk`) ON DELETE CASCADE ON UPDATE CASCADE

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.1.4 Implementasi Antarmuka

Implementasi antar muka 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 Admin

Menu	Deskripsi	Nama File
Login	Digunakan sebagai halaman login admin	index.php
Dashboard	Digunakan untuk melihat informasi di dashboard admin	dasboard.php
Data User		
Admin	Digunakan untuk melihat data admin	admin.php
Member	Digunakan untuk melihat data *member*	member.php
Data Produk		
Kategori	Digunakan untuk melihat, menambah, merubah dan menghapus data kategori	kategori.php
Produk	Digunakan untuk melihat, menambah, merubah dan menghapus data produk	produk.php
Stok	Digunakan untuk menambah, merubah stok produk	stok.php
Order & Retur		
Order	Digunakan untuk melihat, merubah data pemesanan produk	order.php
Retur	Digunakan untuk melihat, merubah data retur	retur.php
Wilayah		
Provinsi	Digunakan untuk melihat, menambah, merubah dan menghapus data	provinsi.php

	provinsi	
Kota	Digunakan untuk melihat, menambah, merubah dan menghapus data kota	kota.php
Ongkos & Pengiriman		
Ongkos Pengiriman	Digunakan untuk melihat, menambah, merubah dan menghapus data harga kirim	ongkos.php
Jasa Pengiriman	Digunakan untuk melihat, menambah, merubah dan menghapus data jasa pengiriman	Jasa.php
Paket Jasa Pengiriman	Digunakan untuk melihat, menambah, merubah dan menghapus data paket jasa pengiriman	paket_jasa.php
Laporan		
Laporan Penjualan	Digunakan untu melakukan pengolahan pembelian	laporan_penjualan.php
Laporan Produk	Digunakan untu melakukan pengolahan retur	Laporan_produk.php
Logout	Digunakan administrator untuk dapat keluar dari aplikasi	Logout.php

4.2 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan 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 Alpa

Pengujian alpha dilakukan dengan menggunakan metode *black box*. Metode *black box* adalah suatu pengujian yang memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi input yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian ini bertujuan untuk menunjukan fungsi tentang cara beroperasinya, apakah pemasukan data keluaran telah berjalan sebagaimana yang diharapkan. Untuk pengujian alpha ini dibagi menjadi dua, rencana pengujian member dan rencana pengujian admin.

Tabel 4.4 Pengujian Alpha *Member*

Kelas Uji	Butir Uji	Tabel Pengujian
Daftar *member*	Menampilkan *form* pendaftaran untuk calon *member*	Tabel 4.19
	Isi *form* pendaftaran	
	Simpan data calon *member*	
	Kirim aktivasi *member* ke email	

Login	Isi data login	Tabel 4.20
	Verifikasi *username*	
	Verifikasi *password*	
Logout	Menghapus *session user*	Tabel 4.20
Jumlah pembelian	*Input* jumlah pembelian	Tabel 4.21
Pengiriman	*Input* data pengiriman	Tabel 4.22
Data pembayaran	*Input* data pembayaran	Tabel 4.23

Tabel 4.5 Pengujian Alpha Admin

Kelas Uji	Butir Uji	Tabel Pengujian
Login	Isi data login	Tabel 4.16
	Verifikasi *username*	
	Verifikasi *password*	
Logout	Menghapus *session user*	Tabel 4.23
Data User	Tambah user admin	Tabel 4.15
	Edit user admin	
	Hapus user admin	
Kategori	Tambah data kategori	Tabel 4.6
	Edit data kategori	Tabel 4.7
	Hapus data kategori	Tabel 4.8
Produk	Cari data produk	Tabel 4.9
	Tambah data produk	Tabel 4.10
Stok	Edit data stok	Tabel 4.11
Order	Cari data pembelian	Tabel 4.18
Provinsi	Tambah data provinsi	Tabel 4.12
	Hapus data provinsi	Tabel 4.13
Data Kota	Tambah data Kota	Tabel 4.14

4.2.1.1 Kasus dan Hasil Pengujian

4.2.1.1.1 Pengujian Tambah Kategori

Tabel 4.6 Pengujian Tambah Kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan kategori yang belum ada Kategori : Boneka Karakter	Menampilkan kategori boneka karakter	Dapat Menampilkan kategori kaos	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan kategori yang sudah ada Kategori : Boneka Karakter	Menampilkan Kategori yang di inputkan “Boneka Karakter gagal disimpan”	Kategori tidak dapat disimpan “Boneka Karakter gagal disimpan"	[√] Diterima [ ] Ditolak

4.2.1.1.2 Pengujian Edit Kategori

Tabel 4.7 Pengujian Edit Kategori

Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Edit dengan kategori yang sudah ada Kategori : Boneka Karakater	Kategori yang diedit bisa berubah	Kategori tidak dapat diedit dan menampilkan pesan”Boneka Karakter gagal disimpan”	[√] Diterima [ ] Ditolak

4.2.1.1.3 Pengujian Hapus Kategori

Tabel 4.8 Pengujian Hapus Kategori

Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Hapus dengan kategori yang sudah ada Kategori : Boneka Karakater	Kategori yang dihapus bisa hilang	Kategori tidak dapat dihapus dan menampilkan pesan”Boneka Karakter gagal dihapus”	[√] Diterima [ ] Ditolak

4.2.1.1.4 Pengujian Cari Produk

Tabel 4.9 Pengujian Cari Produk

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan produk dengan benar Nama Produk : Bola Merah	Menampilkan data produk Bola Merah	Menampilkan data produk Bola Merah	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukkan data produk yang belum ada Nama Produk : Sendal Jepit	Produk tidak ditemukan menampilkan pesan kesalahan "*Ditemukan 0 data dengan query "**sandal**".* Hasil pencarian tidak ditemukan."	Produk tidak ditemukan menampilkan pesan kesalahan "*Ditemukan 0 data dengan query "**sandal**".* Hasil pencarian tidak ditemukan."	[√] Diterima [ ] Ditolak

4.2.1.1.5 Pengujian Tambah Data Produk

Tabel 4.10 Pengujian Tambah Data Produk

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan data produk dengan benar yang belum ada sebelumnya Kategori : Boneka Bola Nama Produk : Bola merah Gambar Utama : bola_pink.jpg Berat Produk : 1 Harga : 3000 Deskripsi : Stok : 2	Data tersimpan ke *database* Dan menampilkan pesan "Boneka Bola berhasil disimpan"	Data tersimpan ke *database* Dan menampilkan pesan "Boneka Bola berhasil disimpan"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kategori : Boneka Bola Nama Produk : Bola merah Gambar Utama : bola_pink.jpg Berat Produk : 4 Harga : 400 Deskripsi : Stok : 2	Data produk tidak tersimpan dan menampilkan pesan kesalahan "Boneka bola gagal disimpan"	Data produk tidak tersimpan dan menampilkan pesan kesalahan "Boneka bola gagal disimpan""	[√] Diterima [ ] Ditolak

4.2.1.1.6 Pengujian Edit Stok

Tabel 4.11 Pengujian Edit Stok

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klick edit Boneka	Data tersimpan ke *database* Dan menampilkan pesan "Produk Id berhasil disimpan"	Data tersimpan ke *database* Dan menampilkan pesan "Boneka Bola berhasil disimpan"	[√] Diterima [ ] Ditolak

4.2.1.1.7 Pengujian Tambah Provinsi

Tabel 4.12 Pengujian Tambah Provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan provinsi yang belum ada provinsi : Aceh	Menampilkan provinsi aceh	Dapat Menampilkan provinsi aceh	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan kategori yang sudah ada Kategori : Aceh	Menampilkan provinsi yang di inputkan "provinsi gagal disimpan"	Kategori tidak dapat disimpan "provinsi gagal disimpan"	[√] Diterima [ ] Ditolak

4.2.1.1.8 Pengujian Hapus Provinsi

Tabel 4.13 Pengujian Hapus Provinsi

Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Hapus dengan kategori yang sudah ada	Provinsi yang dihapus bisa hilang	Provinsi tidak dapat dihapus,menampilkan pesan"Provinsi gagal	[√] Diterima [ ] Ditolak

Provinsi : Aceh		dihapus”	

4.2.1.1.9 Pengujian Tambah Kota

Tabel 4.14 Pengujian Tambah Kota

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan kota yang belum ada provinsi : Jawa Barat kota/kab : Indramayu	Menampilkan kota indramayu	Dapat Menampilkan kota indramayu	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan provinsi yang sudah ada Kota : Indramayu	Menampilkan kota yang di inputkan “kota gagal disimpan”	Kategori tidak dapat disimpan “kota gagal disimpan"	[√] Diterima [ ] Ditolak

4.2.1.1.10 Pengujian Tambah Admin

Tabel 4.15 Pengujian Tambah Kota

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan admin yang belum ada username : Riri password : opan_1410 Emai : opan.1410@gmail.com	Menampilkan user admin baru	Dapat Menampilkan user admin baru	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan admin yang sudah ada username : Riri password : opan_1410 Emai : opan.1410@gmail.com	Menampilkan user gagal disimpan	Menampilkan pesan “Riri gagal disimpan”	[√] Diterima [ ] Ditolak

4.2.1.1.11 Pengujian Login Admin

Tabel 4.16 Pengujian Login Admin

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi dengan data login yang benar Username : admin Password : admin	Menampilkan menu utama *member*	Dapat menampilkan menu utama *member*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi dengan data Username : ari Password : jelek	Menampilkan pesan kesalahan" Username tidak ditemukan."	Dapat menampilkan pesan kesalahan" Username tidak ditemukan.	[√] Diterima [ ] Ditolak

4.2.1.1.12 Pengujian Logout Admin

Tabel 4.17 Pengujian Logout

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik menu logout	Dapat keluar dari menu Admin	Dapat keluar dari menu Admin	[√] Diterima [ ] Ditolak

4.2.1.1.13 Pengujian Pencarian Order

Tabel 4.18 Pengujian Pencarian Pemesanan

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan nama pemesan, status pemesanan dengan benar Nama pemesan : Sopan Sofyan	Menampilkan data pemesanan dengan nama sopan sofyan	Menampilkan data pemesanan dengan nama sopan sofyan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan nama	Menampilkan	Menampilkan	[√] Diterima

pemesan, status pemesanan dengan benar Nama pemesan : Ari	pesan Data tidak ditemukan untuk **ARI**. *query "**sandal**".* Hasil pencarian tidak ditemukan."	pesan data tidak ditemukan untuk **ARI** *data dengan query "**sandal**".* Hasil pencarian tidak ditemukan."	[ ] Ditolak

4.2.1.1.14 Pengujian Daftar *Member*

Tabel 4.19 Pengujian Daftar *Member*

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi dengan data yang benar Email : opan.jelek@gmail.com Username : jelek Password : jelek Konfirmasi Password : jelek Nama Lengkap : Sopan Alamat : Jln. Indramayu No.Tlp : Provinsi :Jawa Barat Kota : Indramayu Kode Pos : 45252	Registrasi dengan username **ipang** berhasil dilakukan. Untuk mengaktifkan account Anda, silahkan lihat email Anda.	Registrasi dengan username **ipang** berhasil dilakukan. Untuk mengaktifkan account Anda, silahkan lihat email Anda.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Kode pos diisi dengan selain angka Kodepos : klmk	Menampilkan pesan kesalahan"harus angka"	Menampilkan pesan kesalahan"harus angka"	[√] Diterima [ ] Ditolak

4.2.1.1.15 Pengujian Login *Member*

Tabel 4.20 Pengujian Login *Member*

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi dengan data login yang benar Username : jelek Password : jelek	Menampilkan menu utama *member*	Dapat menampilkan menu utama *member*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi dengan data Username : ari Password : jelek	Menampilkan pesan kesalahan" Username tidak ditemukan."	Dapat menampilkan pesan kesalahan" Username tidak ditemukan.	[√] Diterima [ ] Ditolak

4.2.1.1.16 Pengujian Jumlah Pembelian

Tabel 4.21 Pengujian Jumlah Pembelian

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi jumlah pembelian dengan benar Jumlah pembelian : 1	Data pembelian masuk ke keranjang belanja	Data pembelian masuk ke keranjang belanja	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi jumlah pembelian dengan selain angka Jumlah : xvt	Menampilkan pesan kesalahan" Jumlah pesanan harus berupa angka."	Menampilkan pesan kesalahan"jumlah pesanan tidak valid"login anda"	[√] Diterima [ ] Ditolak
Mengisi jumlah pembelian melebihi stok Jumlah Pembelian : 20	Menampilkan pesan kesalahan" Jumlah yang dipesan melebihi stok produk."	Menampilkan pesan kesalahan" Jumlah yang dipesan melebihi stok produk.""	[√] Diterima [ ] Ditolak

4.2.1.1.17 Pengujian Pengiriman

Tabel 4.22 Pengujian Pengiriman

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data pengiriman dengan benar Nama Penerima : Ahmad Jaelani Alamat : Jl Merdeka no 1423 Provinsi : Lampung Kota : Bangunrejo Paket JNE : Super Speed (SS) Kodepos : 35389	Menampilkan form alamat pengiriman dan detail pembelian	Menampilkan form alamat pengiriman dan detail pembelian	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
kodepos diisi selain angka kodepos : abc	Menampilkan pesan kesalahan"harus angka"	Menampilkan pesan kesalahan"harus angka"	[√] Diterima [ ] Ditolak

4.2.1.1.18 Pengujian Data Pembayaran

Tabel 4.23 Pengujian Data Pembayaran

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data pembayaran dengan benar Jumlah pembayaran : 225000 No rek : 811255996332	data pembayaran disimpan	data pembayaran disimpan ke *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji Salah (Data Salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Atas nama :	Menampilkan pesan kesalahan "nama harus diisi	Menampilkan pesan kesalahan "nama harus diisi	[√] Diterima [ ] Ditolak
Bank Tujuan : kosong	Menampilkan pesan	Menampilkan pesan	[√] Diterima [ ] Ditolak

	kesalahan”jumlah pembayaran tidak mencukupi”	kesalahan”jumlah pembayaran tidak mencukupi”	

4.2.1.1.19 Pengujian Logout Member

Tabel 4.24Pengujian Data Pembayaran

Kasus dan Hasil Uji Benar (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Klik menu logout	Dapat keluar dari menu member	Dapat keluar dari menu member	[√] Diterima [ ] Ditolak

4.2.1.1.20 Kesimpulan Pengujian Alpha

Berdasarkan hasi pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan aplikasi *e-commerce* berbasis web di Riri Collection sudah tidak ada kesalahan penulisan 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 dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang ada. Pengujian ini dilakukan di dua lingkungan.

1. RIRI COLLECTION

   Pengujian di lingkungan RIRI COLLECTION, adapun beberapa poin-poin yang diteliti adalah :

   a. Promosi produk di toko Riri Collection

b. Kemudahan dalam proses transaksi

c. Perluasan daerah pemasaran Lingkungan Masyarakat Umum

2. Pengujian 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.

Data interval hasil dari kuesioner tersebut, setiap pertanyaan dianalisis dengan menghitung rata – rata jawaban berdasarkan skoring setiap jawaban dari responden. Perhitungan dilakukan berdasarkan skor yang ditetapkan sebagai berikut :

$$Y = P/Q * 100\ \%$$

Keterangan :

P = Banyaknya jawaban responden tiap soal

Q = Jumlah responden

Y = Nilai persentase

Adapun beberapa poin yang diteliti adalah :

a. Kemudahan dalam mendapatkan informasi yang dibutuhkan pelanggan

b. Tampilan antar muka yang menarik.

c. Kemudahan proses pemesanan dan transaksi

Berikut ini adalah hasil persentase masing-masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuisioner ini diujikan kepada 15 orang (pelanggan dan pengunjung).

Tabel 4.25 Hasil Kuesioner

No	Pertanyaan	Jawaban presentase					jumlah
		SS	ST	CK	BS	TS	
1.	Apakah anda mudah mendapatkan informasi mengenai produk beserta detailnya di situs *E-commerce* Riri Collection?	2	12	1			15
	Presentase	13,3	80	6,7			100
2.	Menurut anda, apakah informasi produk yang diberikan oleh website *e-commerce* Riri Collection sudah sesuai dengan informasi yang anda butuhkan?	1	12	2			15
	Presentase	6,7	80	13			100
3.	Apakah anda setuju semua informasi pemesanan dan pembayaran yang dikirim melalui email oleh *website* penjualan *online* Toko Riri Collection sudah sesuai?	2	9	4			15
	Presentase	6,1	66,67	26,67			100
4.	Apakah anda setuju bahwa tampilan *website* ini cukup menarik dan mudah digunakan ?	1	10	4			1
	Presentase	6,7	66,67	13,3			100
5.	Apakah *website* Riri Collection penampilannya sudah menarik pelanggan untuk membeli?	1	8	4	2		15
	Presentase	6,7	53,33	26,67	13,3		100
6.	Apakah anda setuju bahwa dengan website ini menjadi salah satu media penghubung antara Riri Collection dan pelanggannya ?		10	5			15
	Presentase		66,67	33,33			100
7.	Apakah anda setuju dengan adanya *website* penjualan *online* Toko Riri	3	9	3			15

	Collection dapat memperluas pemasarannya?						
	Presentase	20	60	20			100
8.	Apakah anda setuju bahwa *website* ini lebih memudahkan dalam melakukan pemesanan barang?	2	9	3	1		15
	Presentase	13,3	60	20	6,7		100
9.	Apakah dengan adanya *website* penjualan *online* Toko Riri Collection anda tertarik untuk melakukan pemesanan secara *online?*	1	11	3			15
	Presentase	6,7	60	20			100
10.	Menurut anda, apakah anda setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan dalam proses transaksi?	2	12	1			15
	Presentase	13,3	80	6,7			100
11.	Apakah anda setuju bahwa *website* ini akan lebih mempermudah anda dalam melakukan transaksi pembelian dibandingkan transaksi secara manual?	2	10	3			15
	Presentase	13,3	66,67	20			100
12.	Apakah anda setuju bahwa penjualan *online* Toko Riri Collection telah layak untuk di-*online*-kan?	3	8	3	1		15
	Presentase	20	53,33	20	6,7		100
13.	Apakah fitur pada *website* Riri Collection sudah memenuhi?	3	8	3	1		15
	Presentase	20	53,33	20	6,7		100
14.	Apakah menurut ada media komunikasi sudah pada *website* Riri Collection sudah membantu?		12	2	1		15
	Presentase		80	13,3	6,7		100

15.	Apakah Website Riri Collection yang dibangun ini menampilkan dan menghasilkan olahan data yang sesuai?	2	12	1			15
	Presentase	13,3	80	6,7			100

Hasil wawancara pada seorang admin :

1. Apakah dengan adanya *website e-commerce* ini dapat membantu dalam sarana promosi kepada masyarakat luas ?
   Jawaban : *Website* membantu mempromosikan produk yang dijual ke masyarakat
2. Apakah *website* yang dibangun ini mempermudah dalam proses transaksi ?
   Jawaban : Proses transaksi yang dibangun ini mudah dengan adanya pembayaran *online* dan *offline*
3. Apakah *website* yang dibangun ini dapat memperluas daerah pemasaran toko anda?
   Jawaban : dengan adanya *website* ini daerah pemasaran jadi luas
4. Apakah *website* dapat memudahkan dalam pembuatan laporan rekapitulasi penjualan?
   Jawaban : laporan lebih mudah dibuat, sehingga mudah untuk di cek dan mengurangi kesalahan
5. Apakah *website* yang dibangun ini mudah digunakan?
   Jawaban : Informasi yang disampaikan sesuai dengan yang diharapkan sebelumnya

4.2.2.2 Kesimpulan Pengujian Beta

Berdasarkan hasil wawancara dan kuesioner diatas yang didapatkan dari pengujian beta, maka hasil dari wawancara menyatakan penerapan *e-commerce* untuk sistem penjualan ini sudah sesuai dengan tujuan yaitu dapat mempermudah dalam promosi produk, proses transaksi, dapat memperluas daerah pemasaran dan memudahkan dalam pembuatan laporan rekapitulasi penjualan. Sedangkan berdasarkan hasil kuesioner mengenai kemudahan mendapatkan informasi produk didapat rata-rata 73% responden menjawab setuju kemudian mengenai tampilan antar muka didapat rata-rata 66% responden menjawab setuju dan mengenai kemudahan dalam proses transaksi didapat rata-rata 73% responden menjawab setuju dan 10% responden menjawab sangat setuju. Jadi kesimpulannya mayoritas responden setuju dengan kemudahan informasi, proses transaksi dan tampilan menarik yang ada di *website* Riri Collection.