BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem yang baru. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan diimplementasikan pada bahasa pemrograman. Setelah implementasi maka dilakukan pengujian terhadap sistem yang baru dan akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk pengembangan sistem selanjutnya.

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap menerjemahkan perancangan berdasarkan hasil analisi dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Tujuan implementasi ini untuk mengkonfirmasikan modul program perancangan kepada para pelaku sistem sehingga *user* dapat memberi masukan kepada pengembang sistem.

4.1.1 Implementasi Kebutuhan Sistem

Implementasi kebutuhan sistem terdiri atas implementasi perangkat keras dan implementasi perangkat lunak.

4.1.1.1 Implementasi Perangkat Keras

Kebutuhan minimal perangkat keras (*Hardware*) yang digunakan untuk mengimplementasikan aplikasi *e-commerce* di PT. Firebolt Citra Indotama ini adalah perangkat keras komputer yang diusulkan dengan spesifikasi sebagai berikut:

1. *Hardisk* 120 GB
2. *Memory* 1024 MB
3. *Monitor* 15’
4. *Keyboard*
5. *Mouse*
6. *Processor* Intel(R) Core(TM) 2 Duo 1.7GHz (2 CPUs)
7. *Printer*

4.1.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak (*Software*) yang dipasang pada sistem komputer yang akan digunakan untuk mengembangkan aplikasi *e-commerce* di PT. Firebolt Citra Indotama ini ini adalah sebagai berikut :

1. *PHP* 5.3.4 sebagai bahasa pemrograman
2. Sistem Operasi *Windows 7*
3. *PhpMyAdmin* 3.3.9 sebagai *interfacedatabase*
4. XAMPP Version 2.5 sebagai *webserver*
5. *Adobe Dreamweaver CS3* sebagai kode dan perancangan tampilan
6. *Microsoft Visio 2007* sebagai pembuatan gambar dalam analisis sistem

7. *MySQL 5.0.7* yang digunakan sebagai *Database Management System* (DBMS).
8. *Mozilla Firefox dan Chrome* yang digunakan sebagai *Web browser* untuk membuka aplikasi *E-commerce* ini

4.1.2 Implementasi Basis Data

Implementasi basis data dari aplikasi *e-commerce* di PT. Firebolt Citra indotama ini adalah sebagai berikut :

1. Admin

```
CREATE TABLE `admin` (
                  `id_admin` int(2) NOT NULL AUTO_INCREMENT,
                  `username` varchar(50) NOT NULL,
                  `password` varchar(50) NOT NULL,
                  `nama_lengkap` varchar(100) NOT NULL,
                  `email` varchar(100) NOT NULL,
                  `no_telp` varchar(20) NOT NULL,
                  `level` varchar(20) NOT NULL DEFAULT 'user',
                  `blokir` char(1) NOT NULL,
                  PRIMARY KEY (`id_admin`),
                  ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Bank

```
CREATE TABLE `bank` (
                  `id_bank` int(2) NOT NULL AUTO_INCREMENT,
                  `nama_bank` varchar(20) NOT NULL,
                  `no_rek` varchar(20) NOT NULL,
                  `pemilik_rek` varchar(30) NOT NULL,
                  `gambar` varchar(100) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_bank`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT `bank_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

3. harga_paket

```
CREATE TABLE `harga_paket` (
                  `id_hargapaket` int(5) NOT NULL AUTO_INCREMENT,
                  `harga_paket` double NOT NULL,
                  `id_kabkota` int(5) NOT NULL,
                  `id_paket` int(6) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_hargapaket`),
                  KEY `id_kabkota` (`id_kabkota`),
                  KEY `id_paket` (`id_paket`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT   `harga_paket_ibfk_9`   FOREIGN   KEY
(`id_admin`) REFERENCES `admin` (`id_admin`),
                  CONSTRAINT   `harga_paket_ibfk_7`   FOREIGN   KEY
(`id_kabkota`) REFERENCES `kabkota` (`id_kabkota`),
                  CONSTRAINT   `harga_paket_ibfk_8`   FOREIGN   KEY
(`id_paket`) REFERENCES `paket` (`id_paket`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4. hubungi

```
CREATE TABLE `hubungi` (
                  `id_hubungi` int(5) NOT NULL AUTO_INCREMENT,
                  `nama_pengirim` varchar(100) NOT NULL,
                  `email` varchar(100) NOT NULL,
                  `subjek` varchar(100) NOT NULL,
                  `pesan` text NOT NULL,
                  `tgl_hubungi` date NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_hubungi`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT    `hubungi_ibfk_1`    FOREIGN    KEY
(`id_admin`) REFERENCES `admin` (`id_admin`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. kabkota

```
CREATE TABLE `kabkota` (
                  `id_kabkota` int(5) NOT NULL AUTO_INCREMENT,
                  `nama_kabkota` varchar(50) NOT NULL,
                  `jns_kabkota` varchar(20) NOT NULL,
                  `id_provinsi` int(5) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_kabkota`),
                  KEY `id_provinsi` (`id_provinsi`),
                  KEY `id_admin` (`id_admin`),
```

```
                  CONSTRAINT     `kabkota_ibfk_6`     FOREIGN     KEY
(`id_admin`) REFERENCES `admin` (`id_admin`),
                  CONSTRAINT     `kabkota_ibfk_5`     FOREIGN     KEY
(`id_provinsi`) REFERENCES `provinsi` (`id_provinsi`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

6. kategori

```
CREATE TABLE `kategori` (
                  `id_kategori` int(3) NOT NULL AUTO_INCREMENT,
                  `nama_kategori` varchar(100) NOT NULL,
                  `seo_kategori` varchar(100) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_kategori`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT     `kategori_ibfk_1`     FOREIGN     KEY
(`id_admin`) REFERENCES `admin` (`id_admin`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

7. kurs

```
CREATE TABLE `kurs` (
                  `id_kurs` int(2) NOT NULL AUTO_INCREMENT,
                  `harga_kurs` double NOT NULL,
                  `tgl_kurs` date NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_kurs`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT `kurs_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

8. orders

```
CREATE TABLE `orders` (
                  `id_order` int(10) NOT NULL AUTO_INCREMENT,
                  `nama_pemesan` varchar(100) NOT NULL,
                  `alamat` text NOT NULL,
                  `email` varchar(50) NOT NULL,
                  `status_order`   varchar(50)   NOT   NULL   DEFAULT
'baru',
                  `tgl_order` date NOT NULL,
                  `tgl_kirim` date NOT NULL,
                  `tgl_terima` date NOT NULL,
                  `no_resi` int(11) NOT NULL,
                  `jam_order` time NOT NULL,
```

```
                  `id_provinsi` int(5) NOT NULL,
                  `id_kabkota` int(5) NOT NULL,
                  `ongkos_kirim` double NOT NULL,
                  `grand_total` double NOT NULL,
                  `id_paket` int(2) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_order`),
                  KEY `email` (`email`),
                  KEY `id_provinsi` (`id_provinsi`),
                  KEY `id_kabkota` (`id_kabkota`),
                  KEY `id_paket` (`id_paket`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT    `orders_ibfk_10`    FOREIGN    KEY
(`id_admin`) REFERENCES `admin` (`id_admin`),
                  CONSTRAINT `orders_ibfk_6` FOREIGN KEY (`email`)
REFERENCES `pelanggan` (`email`),
                  CONSTRAINT    `orders_ibfk_7`    FOREIGN    KEY
(`id_provinsi`) REFERENCES `provinsi` (`id_provinsi`),
                  CONSTRAINT    `orders_ibfk_8`    FOREIGN    KEY
(`id_kabkota`) REFERENCES `kabkota` (`id_kabkota`),
                  CONSTRAINT    `orders_ibfk_9`    FOREIGN    KEY
(`id_paket`) REFERENCES `paket` (`id_paket`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

9. order_detail

```
CREATE TABLE `order_detail` (
                  `id_orderdetail` int(10) NOT NULL AUTO_INCREMENT,
                  `id_order` int(10) NOT NULL,
                  `id_produk` int(5) NOT NULL,
                  `jumlah` int(5) NOT NULL,
                  `harga` double NOT NULL,
                  `total` double NOT NULL,
                  PRIMARY KEY (`id_orderdetail`),
                  KEY `id_order` (`id_order`),
                  KEY `id_produk` (`id_produk`),
                  CONSTRAINT   `order_detail_ibfk_4`   FOREIGN   KEY
(`id_produk`) REFERENCES `produk` (`id_produk`),
                  CONSTRAINT   `order_detail_ibfk_3`   FOREIGN   KEY
(`id_order`) REFERENCES `orders` (`id_order`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

10. keranjang

```
CREATE TABLE `keranjang` (
                  `id_keranjang` int(5) NOT NULL AUTO_INCREMENT,
                  `id_produk` int(5) NOT NULL,
                  `id_session` varchar(100) NOT NULL,
```

```
                  `jumlah` int(5) NOT NULL,
                  `tgl_keranjang` date NOT NULL,
                  `jam_keranjang` time NOT NULL,
                  `stok_keranjang` int(5) NOT NULL,
                  PRIMARY KEY (`id_keranjang`),
                  KEY `id_produk` (`id_produk`),
                  CONSTRAINT    `keranjang_ibfk_1`    FOREIGN    KEY
(`id_produk`) REFERENCES `produk` (`id_produk`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

11. paket

```
CREATE TABLE `paket` (
                  `id_paket` int(2) NOT NULL AUTO_INCREMENT,
                  `nama_paket` varchar(10) NOT NULL,
                  `lama_pengiriman` int(2) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_paket`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT `paket_ibfk_1` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

12. pelanggan

```
CREATE TABLE `pelanggan` (
                  `email` varchar(50) NOT NULL,
                  `password` varchar(50) NOT NULL,
                  `nama_lengkap` varchar(100) NOT NULL,
                  `alamat` text NOT NULL,
                  `id_provinsi` int(5) NOT NULL,
                  `id_kabkota` int(5) NOT NULL,
                  `kode_pos` varchar(10) NOT NULL,
                  `aktif` enum('1','0') NOT NULL DEFAULT '0',
                  `tgl_daftar` date NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`email`),
                  KEY `id_provinsi` (`id_provinsi`),
                  KEY `id_kabkota` (`id_kabkota`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT    `pelanggan_ibfk_6`    FOREIGN    KEY
(`id_admin`) REFERENCES `admin` (`id_admin`),
                  CONSTRAINT    `pelanggan_ibfk_4`    FOREIGN    KEY
(`id_provinsi`) REFERENCES `provinsi` (`id_provinsi`),
                  CONSTRAINT    `pelanggan_ibfk_5`    FOREIGN    KEY
(`id_kabkota`) REFERENCES `kabkota` (`id_kabkota`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13. pembayaran

```
CREATE TABLE `pembayaran` (
                  `id_pembayaran` int(10) NOT NULL AUTO_INCREMENT,
                  `id_order` int(10) NOT NULL,
                  `email` varchar(100) NOT NULL,
                  `id_bank` int(2) DEFAULT NULL,
                  `id_kurs` int(3) DEFAULT NULL,
                  `total_bayar` double NOT NULL,
                  `tgl_bayar` date NOT NULL,
                  `no_transaksi` varchar(10) NOT NULL,
                  `status` varchar(10) NOT NULL,
                  PRIMARY KEY (`id_pembayaran`),
                  KEY `id_order` (`id_order`),
                  KEY `email` (`email`),
                  KEY `id_bank` (`id_bank`),
                  KEY `id_kurs` (`id_kurs`),
                  CONSTRAINT    `pembayaran_ibfk_7`    FOREIGN    KEY
(`id_kurs`) REFERENCES `kurs` (`id_kurs`),
                  CONSTRAINT    `pembayaran_ibfk_4`    FOREIGN    KEY
(`id_order`) REFERENCES `orders` (`id_order`),
                  CONSTRAINT    `pembayaran_ibfk_5`    FOREIGN    KEY
(`email`) REFERENCES `pelanggan` (`email`),
                  CONSTRAINT    `pembayaran_ibfk_6`    FOREIGN    KEY
(`id_bank`) REFERENCES `bank` (`id_bank`)
               ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

14. produk

```
CREATE TABLE `produk` (
                  `id_produk` int(5) NOT NULL AUTO_INCREMENT,
                  `id_kategori` int(5) NOT NULL,
                  `nama_produk` varchar(100) NOT NULL,
                  `seo_produk` varchar(100) NOT NULL,
                  `deskripsi` text NOT NULL,
                  `stok` int(5) NOT NULL,
                  `berat` decimal(5,2) NOT NULL,
                  `ukuran` varchar(3) NOT NULL,
                  `harga_produk` double NOT NULL,
                  `tgl_masuk` date NOT NULL,
                  `gambar` varchar(100) NOT NULL,
                  `dibeli` int(5) NOT NULL,
                  `dilihat` int(5) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_produk`),
                  KEY `id_kategori` (`id_kategori`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT     `produk_ibfk_4`     FOREIGN     KEY
(`id_admin`) REFERENCES `admin` (`id_admin`),
```

```
                  CONSTRAINT     `produk_ibfk_3`     FOREIGN     KEY
(`id_kategori`) REFERENCES `kategori` (`id_kategori`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

15. provinsi

```
CREATE TABLE `provinsi` (
                  `id_provinsi` int(5) NOT NULL AUTO_INCREMENT,
                  `nama_provinsi` varchar(100) NOT NULL,
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_provinsi`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT `provinsi_ibfk_1` FOREIGN KEY
(`id_admin`) REFERENCES `admin` (`id_admin`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

16. retur

```
CREATE TABLE `retur` (
                  `id_retur` int(5) NOT NULL AUTO_INCREMENT,
                  `id_order` int(10) NOT NULL,
                  `email` varchar(100) NOT NULL,
                  `tgl_retur` date NOT NULL,
                  `tgl_kirim_retur` date NOT NULL,
                  `tgl_terima_retur` date NOT NULL,
                  `no_resi_retur` int(10) NOT NULL,
                  `status_retur` varchar(10) NOT NULL DEFAULT
'baru',
                  `id_admin` int(2) NOT NULL DEFAULT '1',
                  PRIMARY KEY (`id_retur`),
                  KEY `id_order` (`id_order`),
                  KEY `email` (`email`),
                  KEY `id_admin` (`id_admin`),
                  CONSTRAINT `retur_ibfk_4` FOREIGN KEY (`id_order`)
REFERENCES `orders` (`id_order`),
                  CONSTRAINT `retur_ibfk_5` FOREIGN KEY (`email`)
REFERENCES `pelanggan` (`email`),
                  CONSTRAINT `retur_ibfk_6` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

17. retur detail

```
CREATE TABLE `retur_detail` (
                  `id_returdetail` int(5) NOT NULL AUTO_INCREMENT,
                  `id_retur` int(10) NOT NULL,
```

```
                  `id_orderdetail` int(10) NOT NULL,
                  `jumlah` int(5) NOT NULL,
                  `keterangan` text NOT NULL,
                  PRIMARY KEY (`id_returdetail`),
                  KEY `id_retur` (`id_retur`),
                  KEY `id_orderdetail` (`id_orderdetail`),
                  CONSTRAINT   `retur_detail_ibfk_3`   FOREIGN   KEY
(`id_retur`) REFERENCES `retur` (`id_retur`),
                  CONSTRAINT   `retur_detail_ibfk_4`   FOREIGN   KEY
(`id_orderdetail`) REFERENCES `order_detail` (`id_orderdetail`)
                ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.1.3 Implementasi Antarmuka

Implementasi sistem menghasilkan tampilan program dan sistem sesuai dengan hasil dari perancangan yang telah dilakukan sebelumnya. Untuk memperjelas bentuk dari implementasi diatas, berikut ini adalah implementasi antarmuka untuk pengguna. Adapun untuk tampilan implementasi akan dilampirkan pada halaman lampiran.

Tabel 4.1 implementasi antarmuka pengunjung dan pelanggan

Menu	Deskripsi	Nama file
Registrasi pelanggan	Antarmuka yang digunakan untuk melakukan pendaftaran untuk menjadi seorang pelanggan.	registrasi.php
Masuk pelanggan	Antarmuka yang digunakan sebagai halaman untuk *login* bagi pelanggan	*login*.php
Lupa kata sandi	Antarmuka yang digunakan apabila seorang pelanggan lupa dengan kata sandi	media.php?module=lupa*p assword*
Ubah Akun: - Ubah biodata - Ganti kata sandi		
Profil perusahaan	Antarmuka yang digunakan untuk melihat profil perusahaan	media.php?module=profil kami

Kategori	Antarmuka yang digunakan untuk melihat produk perkategori	media.php?module=detail kategori
Produk : - Produk terbaru	Antarmuka yang digunakan untuk melihat seluruh produk terbaru	media.php?module=home
- Produk terlaris	Antarmuka yang digunakan untuk melihat produk yang paling banyak dibeli	media.php?module=home
- Produk terpopuler	Antarmuka yang digunakan untuk melihat produk yang paling banyak dilihat	media.php?module=home
- Pencarian produk	Antarmuka untuk pencarian produk	media.php?module=hasilcari
Cara pembelian	Antarmuka yang digunakan untuk melihat tata cara pembelian pada aplikasi *e-commerce* ini.	media.php?module=carabeli
Keranjang belanja	Antarmuka yang digunakan untuk menyimpan pesanan belanja yang telah dipilih	media.php?module=keranjangbelanja
Konfirmasi pesanan	Antarmuka yang digunakan untuk melakukan konfirmasi pemesanan pada produk yang telah dimasukkan ke keranjang belanja	media.php?module=selesaibelanja
Konfirmasi pembayaran	Antarmuka yang digunakan untuk melakukan konfirmasi pembayaran pada produk yang telah dipesan	media.php?module=bayar
Data kontak	Antarmuka yang digunakan untuk melihat kontak perusahaan	media.php?module=hubungikami
Logout	Digunakan untuk *logout* pelanggan yang telah *login*	aksi.php?module=logout

Tabel 4.2 Implementasi antarmuka bagian gudang

Menu	Deskripsi	Nama file
Login bagian gudang	Antarmuka yang digunakan sebagai halaman untuk *login* bagi general manager	index.php
Lupa Kata Sandi	Antarmuka yang digunakan apabila seorang admin lupa dengan kata sandi	lupa_*password*.php
Ganti Kata Sandi	Antarmuka yang digunakan untuk merubah *password* admin	passuser.php
Pengolahan data bank	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data bank.	bank.php
Pengolahan data ongkos kirim	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data ongkos kirim.	harga_paket.php
Pengolahan data kabkota	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data kabkota.	kabkota.php
Pengolahan data kategori	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data kategori.	kategori.php
Pengolahan data kurs	Antarmuka yang digunakan untuk melihat dan merubah data kurs.	kurs.php
Pengolahan data orders	Antarmuka yang digunakan untuk melihat dan merubah data pemesanan.	order.php
Pengolahan data paket	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data paket.	paket.php

Pengolahan data pelanggan	Antarmuka yang digunakan untuk melihat, merubah dan menghapus data pelanggan.	pelanggan.php
Pengolahan data pembayaran	Antarmuka yang digunakan untuk melihat data pembayaran.	pembayaran.php
Pengolahan data produk	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data produk.	produk.php
Pengolahan data provinsi	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data provinsi.	provinsi.php
Pengolahan data retur	Antarmuka yang digunakan untuk melihat dan merubah retur.	retur.php
Pengolahan laporan	Antarmuka yang digunakan untuk membuat laporan produk, laporan pemesanan, dan laporan retur.	laporan.php

Tabel 4.3 Implementasi antarmuka general manager

Menu	Deskripsi	Nama file
Login general manager	Antarmuka yang digunakan sebagai halaman untuk *login* bagi general manager	index.php
Lupa Kata Sandi	Antarmuka yang digunakan apabila seorang admin lupa dengan kata sandi	lupa_*password*.php
Ganti Kata Sandi	Antarmuka yang digunakan untuk merubah *password* admin	passdir.php
Pengolahan data bagian gudang	Antarmuka yang digunakan untuk melihat, menambah, merubah dan menghapus data bagian operator.	admin.php

Pengolahan laporan	Antarmuka yang digunakan untuk membuat laporan produk, laporan pemesanan, dan laporan retur.	laporan.php

4.2 Pengujian Sistem

Pengujian sistem merupakan metode yang dilakukan untuk menjelaskan mengenai pengoperasian perangkat lunak yang terdiri dari perangkat pengujian, metode pengujian dan pelaksanaan pengujian.

4.2.1 Metode pengujian

Pengujian yang dilakukan pada program ini menggunakan metode *Black box* . Pengujian *Black Box* merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *Black Box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode *blackbox* dilakukan dengan cara memberikan sejumlah input pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah program aplikasi menghasilkan output yang diinginkan dan sesuai dengan fungsi dari program tersebut. Apabila dari input yang diberikan proses menghasilkan output yang sesuai dengan kebutuhan fungsionalnya, maka program aplikasi yang bersangkutan telah benar, tetapi jika output yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi.

Teknik pengujian yang digunakan adalah *equivalence partitioning* dimana pengujian aplikasi dilakukan dengan cara membagi data masukan dari unit perangkat lunak menjadi beberapa partisi data dari mana test *case* dapat diturunkan Rencana Pengujian. Pada prinsipnya, pengujian dirancang untuk menutupi setiap partisi minimal sekali. Teknik ini mencoba mendefinisikan pengujian yang mengungkap kelas kesalahan, sehingga mengurangi jumlah kasus uji yang harus di kembangkan.

4.2.2 Rencana Pengujian

Rencana pengujian aplikasi *e-commerce* di PT. Firebolt Citra Indotama ini menggunakan data uji berdasarkan data dari masing-masing data. Rencana pengujian dilakukan secara alpa dan beta. Rencana selengkapnya dapat dilihat pada tabel berikut.

Tabel 4.4 Rencana pengujian *e-commerce* pada pelanggan dan pengunjung (frontend)

Kelas Pengujian	Butir Pengujian	Jenis Pengujian
Registrasi	Input data registrasi	*Black box*
Login	*Verifikasi username dan password*	*Black box*
Ubah biodata	Merubah data pelanggan	*Black box*
Ubah *password*	Merubah *password* pelanggan	*Black box*
Profil perusahaan	Melihat profil perusahaan	*Black box*
Produk	Menampilkan data produk	*Black box*
	Menampilkan detail produk	*Black box*
Kategori	Menampilkan data produk sesuai kategori	*Black box*
Keranjang belanja	Menampilkan data pesanan	*Black box*

	Menghapus data pesanan	*Black box*
Cara pembelian	Menampilkan cara pembelian	*Black box*
Konfirmasi pemesanan	Input konfirmasi pemesanan	*Black box*
Konfirmasi pembayaran	Input konfirmasi pembayaran	*Black box* *Black box*
Retur produk	Input retur produk	*Black box*
Hubungi kami	Menampilkan data hubungi kami	*Black box*
Kritik dan saran	Pengisian data kritik dan saran	*Black box*
Logout pelanggan	Menghapus *session*	*Black box*

Tabel 4.5 Rencana pengujian *e-commerce* pada bagian gudang (backend)

Item Pengujian	Detail Pengujian	Jenis Pengujian
Login	*Login* admin	*Black box*
Ganti *password*	Ubah *password* admin	*Black box*
Pengolahan data bank	Tampil data bank	*Black box*
	Tambah data bank	*Black box*
	Ubah data bank	*Black box*
	Hapus data bank	*Black box*
Pengolahan data paket	Tampil data paket	*Black box*
	Tambah data paket	*Black box*
	Ubah data paket	*Black box*
	Hapus data paket	*Black box*
Pengolahan data kabkota	Tampil data kabkota	*Black box*
	Tambah data kabkota	*Black box*
	Ubah data kabkota	*Black box*
	Cari data kabkota	*Black box*
	Hapus data kabkota	*Black box*
Pengolahan data kategori	Tampil data kategori	*Black box*
	Tambah data kategori	*Black box*
	Ubah data kategori	*Black box*

	Hapus data kategori	*Black box*
Pengolahan data kurs	Tampil data kurs	*Black box*
	Ubah data kurs	*Black box*
Pengolahan data orders	Tampil data pemesanan	*Black box*
	Ubah data pemesanan	*Black box*
Pengolahan data ongkos kirim	Tampil data ongkos kirim	*Black box*
	Tambah data ongkos kirim	*Black box*
	Ubah data ongkos kirim	*Black box*
	Cari data ongkos kirim	*Black box*
	Hapus data ongkos kirim	*Black box*
Pengolahan data pelanggan	Tampil data pelanggan	*Black box*
	Ubah Aktivasi data pelanggan	*Black box*
	Cari data pelanggan	*Black box*
	Hapus data pelanggan	*Black box*
Pengolahan data pesan	Tampil data pesan	*Black box*
	Hapus data pesan	*Black box*
Pengolahan data produk	Tampil data produk	*Black box*
	Tambah data produk	*Black box*
	Ubah data produk	*Black box*
	Cari data produk	*Black box*
	Hapus data produk	*Black box*
Pengolahan data provinsi	Tampil data provinsi	*Black box*
	Tambah data provinsi	*Black box*
	Ubah data provinsi	*Black box*
	Cari data provinsi	*Black box*
	Hapus data provinsi	*Black box*
Pengolahan data retur	Tampil data retur	*Black box*
	Ubah data retur	*Black box*
Pengolahan laporan	Mencetak laporan produk	*Black box*
	Mencetak laporan pemesanan	*Black box*
	Mencetak aporan retur	*Black box*
Logout bagian gudang	Menghapus *session*	*Black box*

Tabel 4.6 Rencana pengujian pada general manager (backend)

Item Pengujian	Detail Pengujian	Jenis Pengujian
Login	*Login* admin	*Black box*
Ganti *password*	Ubah *password* admin	*Black box*
Pengolahan data bagian gudang	Tampil data bagian gudang	*Black box*
	Tambah data bagian gudang	*Black box*
	Ubah data bagian gudang	*Black box*
	Hapus data bagian gudang	*Black box*
Pengolahan laporan	Mencetak laporan produk	*Black box*
	Mencetak laporan pemesanan	*Black box*
	Mencetak laporan retur	*Black box*
Logout general manager	Menghapus *session*	*Black box*

4.2.3 Skenario Pengujian

Hasil implementasi perangkat lunak sistem informasi penjualan ini diuji untuk melihat masih ada kesalah atau tidak pada aplikasi. Pengujian perangkat lunak sistem informasi penjualan ini menggunakan data uji berdasarkan field pada form yang terdapat didalam aplikasi.

4.2.3.1 Pengujian Pelanggan dan Pengunjung

Pada bagian ini akan dilakukan pengujian terhadap aplikasi *e-commerce* yang dilakukan untuk pelanggan dan pengunjung.

1. **Pengujian registrasi pengunjung**

Berikut ini merupakan pengujian yang dilakukan untuk melakukan pendaftar pelanggan.

Tabel 4.7 Pengujian registrasi pengunjung

Kasus dan Hasil Uji (Data *Valid*)			
Data Masukan	***Valid Class***	**Pengamatan**	**Kesimpulan**
Email Contoh masukan: *Email* : rion_bdg@yahoo.com	[‘A’-‘Z’|’a’-‘z’|0-9|@|.]	Data *email* dapat diterima jika input data: 1. *Email* 5 samapi 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus mengandung simbol “@” dan ”.” Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika salah satu field tidak diisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Password Contoh masukan: *Password* : test123	[‘a’-’z’|’A’-’Z’|0-9]	Data *password* dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi	[√] Diterima [ ] Ditolak

		antara angka dengan huruf	
Ulang Password Contoh masukan: Ulang *password*: test123	[‘a’-’z’|’A’-’Z’|0-9]	Data ulang *password* dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus sama dengan field *password*	[√] Diterima [ ] Ditolak
Nama Lengkap Contoh masukan: Nama Lengkap: Uchi sanusi	[‘a’-’z’|’A’-’Z’]	Data nama lengkap dapat diterima jika input data: 1. Nama 3 samapai 30 char 2. Tidak boleh menggunakan angka	[√] Diterima [ ] Ditolak
Alamat Contoh masukan: Alamat: kp. Panguseupan rt/rw 04/04	[‘a’-’z’|’A’-’Z’|0-9]	Data alamat dapat diterima jika input data: 1. Alamat diisi dengan char 2. Boleh menggunakan angka 3. Boleh kombinasi angka dah huruf	[√] Diterima [ ] Ditolak
Provinsi	Pilih Provinsi	Pengguna	[√] Diterima

Contoh masukan: Provnsi: Banten	[1|2|3...]	memilih provinsi dengan option yang tersedia	[ ] Ditolak
Kota/Kabupaten Contoh masukan: Kota/Kabupaten: Pandeglang	Pilih Kota/Kab [1|2|3...]	Pengguna memilih kota/kabupaten menggunakan option yang tersedia dan sebelumnya sudah memilih provinsi	[√] Diterima [ ] Ditolak
Kode pos Contoh masukan: Kode pos: 42264	[0-9]	Data nama lengkap dapat diterima jika input data: 1. Kode pos harus 5 number 2. Kode pos diisi dengan number	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji(Data *Invalid*)			
Data Masukan	***Invalid Class***	**Pengamatan**	**Kesimpulan**
Email Contoh masukan: *Email* : rion_bdg.com *Email*:	[‘a’-’z’|’A’-’Z’|0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Password Contoh masukan: *Password* : 123 *Password*:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Ulang Password Contoh masukan:	Data kosong	Dapat memunculkan	[√] Diterima [ ] Ditolak

Ulang *password*: 321 Ulang *password*:		pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter, data tidak sama dengan field *password*	
Nama Lengkap Contoh masukan: Nama Lengkap: Uchi sanusi Nama lengkap:Aa Nama lengkap : qwertyuiopas Dfghjklzxcvb Nmqwejhj Nama lengkap:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 3 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Alamat Contoh masukan: Alamat:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak
Provinsi Contoh masukan: Provnsi:	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih provinsi dengan option yang tersedia	[√] Diterima [ ] Ditolak
Kota/Kabupaten Contoh masukan: Kota/Kabupaten:	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih kota/kabupaten	[√] Diterima [ ] Ditolak

		dengan option yang tersedia dan sebelumnya harus memilih provinsi	
Kode pos Contoh masukan: Kode pos: 422 Kode pos:ABCD Kode pos 123456	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika mengisi dengan huruf dan kurang kurang dari 5 atau lebih dari 5 number	[√] Diterima [ ] Ditolak

2. Pengujian *login* pelanggan

Berikut ini merupakan pengujian yang dilakukan untuk melakukan *login* pelanggan.

Tabel 4.8 Pengujian login pelanggan

Kasus dan Hasil Uji (Data *Valid*)			
Data Masukan	***Valid Class***	**Pengamatan**	**Kesimpulan**
Email Contoh masukan: *Email:* rion_bdg@yahoo.com	[‘a’-’z’|’A’-’Z’|0-9|@|.] Data kosong	Data *email* dapat diterima jika input data: 1. *Email* 5 samapi 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf	[√] Diterima [ ] Ditolak

		4. Harus mengandung simbol "@" dan "." Dapat memunculkan pesan kesalahan jika terdapat kesalahan dalam pengisian.	
Password Contoh masukan: *Password* : testskripsi	['a'-'z'|'A'-'Z'|0-9] Data kosong	Data *Password* dapat diterima jika *input* data: 1. Nama 5 sampai 32 char 2. Boleh mengunakan angka 3. Boleh kombinasi antara angka dengan huruf.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid Class***	**Pengamatan**	**Kesimpulan**
Email Contoh masukan: *Email:* test.com *Email*:	['a'-'z'|'A'-'Z'|0-9] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Password Contoh masukan: *Password* : 123 *Password*:	['a'-'z'|'A'-'Z'|0-9] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam	[√] Diterima [ ] Ditolak

		pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter	

3. Pengujian ubah data pelanggan

Berikut ini merupakan pengujian yang dilakukan untuk merubah data pelanggan.

Tabel 4.9 Pengujian ubah data pelanggan

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Lengkap Contoh masukan: Nama Lengkap: Uchi sanusi	[‘a’-’z’|’A’-’Z’]	Data nama lengkap dapat diterima jika input data: 1. Nama 3 samapai 30 char 2. Tidak boleh menggunakan angka	[√] Diterima [ ] Ditolak
Alamat Contoh masukan: Alamat: kp. Panguseupan rt/rw 04/04	[‘a’-’z’|’A’-’Z’|0-9]	Data alamat dapat diterima jika input data: 1. Alamat diisi dengan char 2. Boleh menggunakan angka 3. Boleh kombinasi angka dah huruf	[√] Diterima [ ] Ditolak
Provinsi Contoh masukan:	Pilih Provinsi [1|2|3...]	Pengguna memilih provinsi	[√] Diterima [ ] Ditolak

Provnsi: Banten		dengan option yang tersedia	
Kota/Kabupaten Contoh masukan: Kota/Kabupaten: Pandeglang	Pilih Kota/Kab [1|2|3...]	Pengguna memilih kota/kabupaten menggunakan option yang tersedia dan sebelumnya sudah memilih provinsi	[√] Diterima [ ] Ditolak
Kode pos Contoh masukan: Kode pos: 42264	[0-9]	Data nama lengkap dapat diterima jika input data: 1. Kode pos harus 5 number 2. Kode pos diisi dengan number	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Lengkap Contoh masukan: Nama Lengkap: Nama Lengkap : 12345	[0-9] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 3 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Alamat Contoh masukan: Alamat:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika data alamat kosong	[√] Diterima [ ] Ditolak
Provinsi Contoh masukan:	Data kosong	Dapat memunculkan	[√] Diterima

Provnsi:		pesan kesalahan jika pengguna tidak memilih provinsi dengan option yang tersedia	[ ] Ditolak
Kota/Kabupaten Contoh masukan: Kota/Kabupaten:	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih kota/kabupaten dengan option yang tersedia dan sebelumnya harus memilih provinsi	[√] Diterima [ ] Ditolak
Kode pos Contoh masukan: Kode pos: Kode pos: ABCD	[‘A’-‘Z’|’a’-‘z’] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika mengisi dengan huruf dan kurang kurang dari 5 atau lebih dari 5 number	[√] Diterima [ ] Ditolak

4. Pengujian ubah *password* pelanggan

Berikut ini merupakan pengujian yang dilakukan untuk merubah *password* pelanggan.

Tabel 4.10 Pengujian ubah password pelanggan

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
***Password* lama**	[‘a’-’z’|’A’-’Z’|0-	Data *password*	[√] Diterima

Contoh masukan: *Password* lama:	9]	baru dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus sama dengan *password* yang lama yang pernah diinputkan	[ ] Ditolak
***Password* baru** Contoh masukan: *Password* baru:	[‘a’-’z’|’A’-’Z’|0-9]	Data *password* baru dapat diterima jika input data: 5. *Password* 5 sampai 30 char 6. Boleh menggunakan angka 7. Boleh kombinasi antara angka dengan huruf	[√] Diterima [ ] Ditolak
Ulang password Contoh masukan: Ulangi *password*:	[‘a’-’z’|’A’-’Z’|0-9]	Data ulang *password* dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi	[√] Diterima [ ] Ditolak

		antara angka dengan huruf 4. Harus sama dengan field *password*	
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
***Password* lama** Contoh masukan: *Password* lama:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter dan *password* lama tidak sama dengan *password* yang dulu pernah diinput	[√] Diterima [ ] Ditolak
***Password* baru** Contoh masukan: *Password* baru:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Ulang password Contoh masukan: Ulangi *password*:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika	[√] Diterima [ ] Ditolak

		kurang dari 5 karakter dan lebih dari 30 karakter, data tidak sama dengan field *password*	

5. Pengujian keranjang belanja

Berikut ini merupakan pengujian yang dilakukan untuk membuka halaman keranjang belanja.

Tabel 4.11 Pengujian keranjang belanja

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Jumlah Produk Contoh: Jumlah produk: 10 Sementara stok: 50	[0-9] <=stok	Data jumlah produk dapat diterima jika input data: 1. Berupa angka 2. Jumlah kurang dari stok yang ada 3. Klik tombol update keranjang dapat menampilkan perubahan data jumlah produk, subtotal dan total pembayaran yang belum termasuk dengan ongkos kirim	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			

Data Masukan	*Invalid class*	Pengamatan	Kesimpulan
Jumlah Produk Contoh: Jumlah produk: 60 Sementara stok: 50	[‘A’-‘Z’|’a’-‘z’] >stok	Dapat memunculkan pesan kesalahan bahwa stok tidak mencukup dan data yang dimasukan invalid apabila diisi dengan huruf	[√] Diterima [ ] Ditolak

6. Pengujian konfirmasi pemesanan

Berikut ini merupakan pengujian yang dilakukan untuk melakukan konfirmasi pemesanan.

Tabel 4.12 Pengujian konfirmasi pemesanan

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama pemesan Contoh masukan: Nama pemesan:	[‘a’-’z’|’A’-’Z’]	Data nama lengkap dapat diterima jika input data: 1. Nama 3 samapai 30 char 2. Tidak boleh menggunakan angka	[√] Diterima [ ] Ditolak
Alamat Contoh masukan: Alamat:	[‘a’-’z’|’A’-’Z’|0-9]	Data alamat dapat diterima jika input data: 1. Alamat diisi dengan char 2. Boleh menggunakan angka 3. Boleh	[√] Diterima [ ] Ditolak

		kombinasi angka dah huruf	
Provinsi tujuan Contoh masukan: Povinsi tujuan	Pilih Provinsi [1|2|3...]	Pengguna memilih provinsi dengan option yang tersedia	[√] Diterima [ ] Ditolak
Kota/Kab tujuan Contoh masukan: Kota/Kab tujuan:	Pilih Kota/Kab [1|2|3...]	Pengguna memilih kota/kabupaten menggunakan option yang tersedia dan sebelumnya sudah memilih provinsi	[√] Diterima [ ] Ditolak
Paket Contoh masukan: Paket:	Pilih Paket [1|2]	Pengguna memilih paket menggunakan option yang tersedia dan sebelumnya sudah memilih kota/kabupaten	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama pemesan Contoh masukan: Nama pemesan: Nama pemesan: 12345	[0-9] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 3 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Alamat Contoh masukan: Alamat:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus	[√] Diterima [ ] Ditolak

		diisi, jika data alamat kosong	
Provinsi tujuan Contoh masukan: Povinsi tujuan:	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih provinsi dengan option yang tersedia	[√] Diterima [ ] Ditolak
Kota/Kab tujuan Contoh masukan: Kota/Kab tujuan:	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih kota/kabupaten dengan option yang tersedia dan sebelumnya harus memilih provinsi	[√] Diterima [ ] Ditolak
Paket Contoh masukan: Paket:	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih paket dengan option yang tersedia dan sebelumnya harus memilih kota/kabupaten	[√] Diterima [ ] Ditolak

7. Pengujian konfirmasi pembayaran

Berikut ini merupakan pengujian yang dilakukan untuk melakukan konfirmasi pembayaran.

Tabel 4.13 Pengujian konfirmasi pembayaran

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
klik konfirmasi pembayaran via paypal	Akun paypal yang sudah terverivikasi, terisi saldo yang cukup	Dapat Terintegrasi dengan sistem pembayaran paypal	[√] Diterima [ ] Ditolak
Tanggal bayar Contoh masukan: Tanggal bayar:	Pilih tanggal	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Bank Contoh masukan: Bank:	Pilih Bank [1|2..]	Pengguna memilih bank dengan option yang tersedia	[√] Diterima [ ] Ditolak
No. Transaksi Contoh masukan: No. Transaksi:	[0-9]	Data nama lengkap dapat diterima jika input data: 1. Nomor transaksi harus lebih dari 10 number 2. Nomor transaksi diisi dengan number	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
klik konfirmasi pembayaran via paypal	Akun paypal yang belum terverivikasi, saldo yang ada	Dapat Terintegrasi dengan sistem pembayaran	[√] Diterima [ ] Ditolak

	tidak mencukupi	paypal , namun tidak bisa melakukan transaksi	
Tanggal bayar Contoh masukan: Tanggal bayar:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi	[√] Diterima [ ] Ditolak
Bank Contoh masukan: Bank:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi	[√] Diterima [ ] Ditolak
No. Transaksi Contoh masukan: No. Transaksi: No. Transaksi: abcdef	[‘A’-‘Z’|’a’|’z’] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, dan datatidak boleh huruf	[√] Diterima [ ] Ditolak

8. Pengujian kritik dan saran

Berikut ini merupakan pengujian yang dilakukan untuk mengisi data kritik dan saran.

Tabel 4.14 Pengujian kritik dan saran

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Contoh masukan: Nama: deden	[‘a’-’z’|’A’-’Z’]	Data nama dapat diterima jika input data: 1. Nama 3 samapai 30 char	[√] Diterima [ ] Ditolak

		2. Tidak boleh menggunakan angka	
Email Contoh masukan: *Email*: deden@yahoo.com	[‘A’-‘Z’|’a’-‘z’|0-9|@|.]	Data *email* dapat diterima jika input data: 1. *Email* 5 samapi 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus mengandung simbol “@” dan ”.” Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika salah satu field tidak diisi atau salah dalam pengisian.	
Subjek Contoh masukan: Subjek: tanya tentang produk	[‘a’-’z’|’A’-’Z’|0-9]	Data subjek dapat diterima jika input data: 1. Subjek 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf	

Pesan Contoh masukan: Pesan: firebolt untuk pembutan produknya berapa kali ya?	[‘a’-’z’|’A’-’Z’|0-9]	Data pesan dapat diterima jika input data: 1. Pesan diisi dengan char 2. Boleh menggunakan angka 3. Boleh kombinasi angka dah huruf	
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Contoh masukan: Nama: Nama: 12345	[0-9] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 3 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Email Contoh masukan: *Email*: *Email*: Sanusi.com	[‘a’-’z’|’A’-’Z’|0-9] Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Subjek Contoh masukan: Subjek:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika	[√] Diterima [ ] Ditolak

		kurang dari 5 karakter dan lebih dari 30 karakter	
Pesan Contoh masukan: Pesan:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika data alamat kosong	[√] Diterima [ ] Ditolak

4.2.3.2 Pengujian Bagian gudang

Pada bagian ini akan dilakukan pengujian terhadap aplikasi *e-commerce* yang dilakukan untuk bagian gudang.

1. Pengujian *login* bagian gudang

Berikut ini merupakan pengujian yang dilakukan untuk melakukan *login* bagian gudang.

Tabel 4.15 Pengujian login bagian gudang

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Username Contoh masukan: *Username:* sanusi	[‘a’-’z’|’A’-’Z’|0-9]	Data *Username* dapat diterima jika *input* data: 1. Nama 5 sampai 32 char 2. Boleh mengunakan angka 3. Boleh kombinasi antara angka dengan huruf.	[√] Diterima [ ] Ditolak

Password Contoh masukan: *Password* : admimistrator	[‘a’-’z’|’A’-’Z’|0-9]	Data *Password* dapat diterima jika *input* data: 1. Nama 5 sampai 32 char 2. Boleh mengunakan angka 3. Boleh kombinasi antara angka dengan huruf.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Username Contoh masukan: *Username:* sanusi	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Password Contoh masukan: *Password* : admimistrator	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak

2. Pengujian ubah *password* bagian gudang

Berikut ini merupakan pengujian yang dilakukan untuk merubah *password* bagian gudang.

Tabel 4.16 Pengujian ubah password bagian gudang

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
***Password* lama** Contoh masukan: *Password* lama:	[‘a’-’z’|’A’-’Z’|0-9]	Data *password* baru dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus sama dengan *password* yang lama yang pernah diinputkan	[√] Diterima [ ] Ditolak
***Password* baru** Contoh masukan: *Password* baru:	[‘a’-’z’|’A’-’Z’|0-9]	Data *password* baru dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf	[√] Diterima [ ] Ditolak

Ulang password Contoh masukan: Ulangi *password*:	[‘a’-’z’|’A’-’Z’|0-9]	Data ulang *password* dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus sama dengan field *password*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
***Password* lama** Contoh masukan: *Password* lama:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter dan *password* lama tidak sama dengan *password* yang dulu pernah diinput	[√] Diterima [ ] Ditolak
***Password* baru** Contoh masukan: *Password* baru:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika	[√] Diterima [ ] Ditolak

		kurang dari 5 karakter dan lebih dari 30 karakter	
Ulang password Contoh masukan: Ulangi *password*:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter, data tidak sama dengan field *password*	[√] Diterima [ ] Ditolak

3. Pengujian pengolahan data bank

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data bank.

Tabel 4.17 Pengujian pengolahan data bank (tambah data bank)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Bank Contoh masukan: Nama Bank: BCA	[‘a’-’z’|’A’-’Z’]	Data nama bank dapat diterima jika input data: 1. Tidak boleh menggunakan angka	[√] Diterima [ ] Ditolak
No. Rekening Contoh masukan: No. Rekening: 7771002271	[0-9]	Data no. Rekening dapat diterima jika input data: 1. Tidak boleh menggunakan huruf	[√] Diterima [ ] Ditolak
Nama Pemilik Contoh masukan: Nama pemilik:	[‘a’-’z’|’A’-’Z’]	Data nama pemilik dapat diterima jika	[√] Diterima [ ] Ditolak

Ahmad Maizir		input data: 1. Tidak boleh menggunakan angka	
Gambar Contoh masukan: Gambar: bca.jpg	[‘a’-’z’|’A’-’Z’|0-9]	Pengguna memilih gambar di storage	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Bank Contoh masukan: Nama Bank: Nama Bank: 1234	[0-9] Data ksosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
No. Rekening Contoh masukan: No. Rekening: No. Rekening: abcdef	[‘a’-’z’|’A’-’Z’] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Nama Pemilik Contoh masukan: Nama pemilik: Nama pemilik: 12345	[0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Gambar Contoh masukan: Gambar:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak

4. Pengujian pengolahan data ongkos kirim

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data ongkos kirim

Tabel 4.18 Pengujian pengolahan data ongkos kirim (tambah data ongkos kirim)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Kabkota Contoh masukan: Nama kabkota: Bandung	Pilih kabkota [1|2|3...]	Pengguna memilih kabkota pada option yang tersedia	[√] Diterima [ ] Ditolak
Nama Paket Contoh masukan: Nama Paket: REG	Pilih paket [1|2]	Pengguna memilih paket pada option yang tersedia	[√] Diterima [ ] Ditolak
Harga Contoh masukan: Harga: 4000	[0-9]	Data harga dapat diterima jika input data: 1. Harga diisi hanya dengan angka	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Kabkota Contoh masukan: Nama kabkota:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak
Nama Paket Contoh masukan: Nama Paket:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak
Harga Contoh masukan: Harga: Harga: abcde	[‘a’-’z’|’A’-’Z’] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak

5. Pengujian pengolahan data kabkota

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data kabupaten/kota.

Tabel 4.19 Pengujian pengolahan data kabkota (tambah data kabkota)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Provinsi Contoh masukan: Nama Provinsi: Banten	Pilih provinsi [1|2|3..]	Pengguna memilih provinsi pada option yang tersedia	[√] Diterima [ ] Ditolak
Nama Kabkota Contoh masukan: Nama Kabkota: Pandeglang	[‘a’-’z’|’A’-’Z’]	Data nama kabkota dapat diterima jika input data: 1. Nama kabkota diisi hanya dengan huruf	[√] Diterima [ ] Ditolak
Jenis Contoh masukan: Jenis: Kabupaten	[‘a’-’z’|’A’-’Z’]	Data jenis dapat diterima jika input data: 1. Jenis diisi hanya dengan huruf	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Provinsi Contoh masukan: Nama Provinsi:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak
Nama Kabkota Contoh masukan: Nama Kabkota: Nama kabkota: 123435	[0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Jenis Contoh masukan:	[0-9] Data kosong	Dapat memunculkan	[√] Diterima [ ] Ditolak

Jenis: Jenis: 12345		pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	

6. Pengujian pengolahan data kategori

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data kategori

Tabel 4.20 Pengujian pengolahan data kategori (tambah data kategori)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Kategori Contoh masukan: Nama Kategori: Dompet	[‘A’-‘Z’|’a’-‘z’]	Data nama kategori dapat diterima jika input data: 1. Nama kategori diisi hanya dengan huruf	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Kategori Contoh masukan: Nama Kategori: Nama kategori: 123456	[0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak

7. Pengujian pengolahan data kurs

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data kurs dollar.

Tabel 4.21 Pengujian pengolahan data kurs (edit data kurs)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Harga Kurs Contoh masukan:	[0-9]	Data harga kurs dapat diterima	[√] Diterima [ ] Ditolak

Harga Kurs: 9500		jika input data: 1. Harga kurs diisi hanya dengan angka	
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Harga Kurs Contoh masukan: Harga Kurs: Harga kurs: abcde	[‘A’-‘Z’|’a’-‘z’] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak

8. Pengujian pengolahan data pemesanan

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data pemesanan

Tabel 4.22 Pengujian pengolahan data pemesanan (ubah status pemesanan)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Status pemesanan data masukan: Status pemesanan: dikirim	Pilih status retur [1|2|3...]	Pengguna memilih status pemesanan pada option yang tersedia	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data in*valid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Status pemesanan data masukan: Status pemesanan:	Data kosong	Dapat memunculkan pesan kesalahan, jika pengguna tidak memilih status pemesanan pada option yang tersedia	[√] Diterima [ ] Ditolak

9. Pengujian pengolahan data paket

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data paket

Tabel 4.23 Pengujian pengolahan data paket (tambah data paket)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Paket Contoh masukan: Nama Paket: REG	[‘A’-‘Z’|’a’-‘z’]	Data nama paket dapat diterima jika input data: 1. Nama paket diisi hanya dengan huruf	[√] Diterima [ ] Ditolak
Lama Pengiriman Contoh masukan: Lama Pengiriman: 3	[0-9]	Data lama pengiriman dapat diterima jika input data: 1. Lama pengiriman diisi hanya dengan angka	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Paket Contoh masukan: Nama Paket: Nama paket: 123	[0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Lama Pengiriman Contoh masukan: Lama Pengiriman: Lama pengiriman: abd	[‘A’-‘Z’|’a’-‘z’] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak

10. Pengujian pengolahan data pelanggan

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data pelanggan

Tabel 4.24 Pengujian pengolahan data pelanggan (ubah aktivasi)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Aktivasi Contoh masukan: Aktivasi: 0	Pilih aktivasi[0|1]	Pengguna memilih status aktivasi pada option yang tersedia	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data in*valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Aktivasi Contoh masukan: Aktivasi:	Data kosong	Muncul pesan kesalahan jika pengguna tidak memilih status aktivasi pada option yang tersedia	[√] Diterima [ ] Ditolak

11. Pengujian pengolahan data produk

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data produk

Tabel 4.25 Pengujian pengolahan data produk (tambah data produk)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Produk Contoh masukan: Nama Produk: fbdp 031 black	[‘A’-‘Z’|’a’-‘z’|0-9]	Data nama produk dapat diterima jika input data: 1. Boleh menggunakan angka 2. Boleh kombinasi antara angka	[√] Diterima [ ] Ditolak

		dengan huruf	
Nama Kategori Contoh masukan: Nama kategori: dompet	Pilih kategori [1|2|3...]	Pengguna memilih kategori pada option yang tersedia	[√] Diterima [ ] Ditolak
Berat Contoh masukan: Berat: 0.3	[0-9]	Data berat dapat diterima jika input data: 1. Berat diisi hanya dengan angka	[√] Diterima [ ] Ditolak
Harga Contoh masukan: Harga: 70000	[0-9]	Data harga dapat diterima jika input data: 1. Harga diisi hanya dengan angka	[√] Diterima [ ] Ditolak
Ukuran Contoh masukan: Ukuran: -	[‘A’-‘Z’|’a’-‘z’|0-9]	Data ukuran dapat diterima jika input data: 1. Ukuran diisi dengan angka 2. Ukuran diisi dengan huruf 3. Ukuran diisi dengan tanda “-”	[√] Diterima [ ] Ditolak
Stok Contoh masukan: Stok: 50	[0-9]	Data stok dapat diterima jika input data: 1. Stok diisi hanya dengan huruf	[√] Diterima [ ] Ditolak
Deskripsi Contoh masukan: Deskripsi: dompet terbaru masa kini	[‘A’-‘Z’|’a’-‘z’|0-9]	Data deskripsi dapat diterima jika input data: 1. Deskripsi diisi dengan char 2. Deskripsi Boleh menggunakan angka	[√] Diterima [ ] Ditolak

		3. Boleh kombinasi angka dan huruf	
Gambar Contoh masukan: Gambar: fbdp-031-black.jpg	[‘A’-‘Z’|’a’-‘z’|0-9]	Pengguna memilih produk dari storage	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Produk Contoh masukan: Nama Produk:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Nama Kategori Contoh masukan: Nama kategori:	Data kosong	Dapat memunculkan pesan kesalahan pengguna tidak memilih option kategori yang tersedia	[√] Diterima [ ] Ditolak
Berat Contoh masukan: Berat: Berat: ac	[‘A’-‘Z’|’a’-‘z’] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau mengisi dengan huruf	[√] Diterima [ ] Ditolak
Harga Contoh masukan: Harga: Harga:abcds	[‘A’-‘Z’|’a’-‘z’] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau mengisi dengan huruf	[√] Diterima [ ] Ditolak
Ukuran Contoh masukan: Ukuran:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak
Stok Contoh masukan:	[‘A’-‘Z’|’a’-‘z’] Data kosong	Dapat memunculkan	[√] Diterima [ ] Ditolak

Stok: Stok: ad		pesan kesalahan, jika field tidak diisi atau diisi dengan huruf	
Deskripsi Contoh masukan: Deskripsi:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak
Gambar Contoh masukan: Gambar:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak

12. Pengujian pengolahan data provinsi

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data provinsi

Tabel 4.26 Pengujian pengolahan data provinsi (tambah data provinsi)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Nama Provinsi Contoh masukan: Nama Provinsi : Banten	[‘A’-‘Z’|’a’-‘z’]	Data nama provinsi dapat diterima jika input data: 1. Nama provinsi hanya diisi dengan huruf	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Nama Provinsi Contoh masukan: Nama Provinsi : Nama provinsi: 12345	[0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau diis dengan angka	[√] Diterima [ ] Ditolak

13. Pengujian pengolahan data retur

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data retur

Tabel 4.27 Pengujian pengolahan data retur (ubah status)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Status retur data masukan: Status retur: ditolak	Pilih status retur [1|2|3...]	Pengguna memilih status retur pada option yang tersedia	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data in*valid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Status retur data masukan: Status retur:	Data kosong	Dapat memunculkan pesan kesalahan, jika pengguna tidak memilih status retur pada option yang tersedia	[√] Diterima [ ] Ditolak

14. Pengujian pengolahan laporan pemesanan

Berikut ini merupakan pengujian yang dilakukan untuk mengolah laporan pemesanan

Tabel 4.28 Pengujian pengolahan laporan pemesanan

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Status pemesanan Contoh masukan: Status retur : diterima	Pilih status retur [1|2|3..]	Pengguna memilih status pemesanan pada option yang tersedia	[√] Diterima [ ] Ditolak
Dari tanggal Contoh masukan: Dari tanggal : 2012-01-1	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak

Sampai tanggal Contoh masukan: Sampai tanggal : 2012-06-17	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Status pemesanan Contoh masukan: Status pemesanan : 	Data kosong	Dapat memunculkan pesan kesalahan, jika pengguna tidak memilih status pemesanan pada option yang tersedia	[√] Diterima [ ] Ditolak
Dari Tanggal Contoh masukan: Dari tanggal :	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Sampai Tanggal Contoh masukan: Sampai tanggal :	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak

15. Pengujian pengolahan laporan retur

Berikut ini merupakan pengujian yang dilakukan untuk mengolah laporan retur

Tabel 4.29 Pengujian pengolahan retur

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Data status retur Contoh masukan:	Pilih status retur [1|2|3..]	Pengguna memilih status	[√] Diterima [ ] Ditolak

Status retur : diterima		retur pada option yang tersedia	
Dari tanggal Contoh masukan: Dari tanggal : 2012-01-1	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Sampai tanggal Contoh masukan: Sampai tanggal : 2012-06-17	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Data status retur Contoh masukan: Status retur :	Data kosong	Dapat memunculkan pesan kesalahan, jika pengguna tidak memilih status retur pada option yang tersedia	[√] Diterima [ ] Ditolak
Dari Tanggal Contoh masukan: Dari tanggal :	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Sampai Tanggal Contoh masukan: Sampai tanggal :	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak

4.2.3.3 Pengujian General manager

Pada bagian ini akan dilakukan pengujian terhadap aplikasi *e-commerce* yang dilakukan untuk general manager

1. Pengujian login general manager

Berikut ini merupakan pengujian yang dilakukan untuk melakukan *login* general manager.

Tabel 4.30 Pengujian login general manager

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Username Contoh masukan: *Username:* maizir	[‘a’-’z’|’A’-’Z’|0-9]	Data *Username* dapat diterima jika *input* data: 1. Nama 5 sampai 32 char 2. Boleh mengunakan angka 3. Boleh kombinasi antara angka dengan huruf.	[√] Diterima [ ] Ditolak
Password Contoh masukan: *Password* : generalmanager	[‘a’-’z’|’A’-’Z’|0-9]	Data *Password* dapat diterima jika *input* data: 1. Nama 5 sampai 32 char 2. Boleh mengunakan angka 3. Boleh kombinasi antara angka dengan huruf.	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Username Contoh masukan: *Username:* sanusi	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Password Contoh masukan: *Password* : admimistrator	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak

2. Pengujian ubah password general manager

Berikut ini merupakan pengujian yang dilakukan untuk merubah *password* general manager.

Tabel 4.31 Pengujian ubah password general manager

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
***Password* lama** Contoh masukan: *Password* lama:	[‘a’-’z’|’A’-’Z’|0-9]	Data *password* baru dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus sama dengan	[√] Diterima [ ] Ditolak

		password yang lama yang pernah diinputkan	
***Password* baru** Contoh masukan: *Password* baru:	[‘a’-’z’|’A’-’Z’|0-9]	Data *password* baru dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf	[√] Diterima [ ] Ditolak
Ulang password Contoh masukan: Ulangi *password*:	[‘a’-’z’|’A’-’Z’|0-9]	Data ulang *password* dapat diterima jika input data: 1. *Password* 5 sampai 30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus sama dengan field *password*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
***Password* lama** Contoh masukan: *Password* lama:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus	[√] Diterima [ ] Ditolak

		diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter dan *password* lama tidak sama dengan *password* yang dulu pernah diinput	
***Password* baru** Contoh masukan: *Password* baru:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter	[√] Diterima [ ] Ditolak
Ulang password Contoh masukan: Ulangi *password*:	Data kosong	Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika kosong atau salah dalam pengisian, jika kurang dari 5 karakter dan lebih dari 30 karakter, data tidak sama dengan field *password*	[√] Diterima [ ] Ditolak

3. Pengujian pengolahan data bagian gudang

Berikut ini merupakan pengujian yang dilakukan untuk mengolah data bagian gudang.

Tabel 4.32 Pengujian pengolahan data bagian gudang (tambah data bagian gudang)

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Username Contoh masukan: Username: sanusi	[‘A’-‘Z’|’a’-‘z’|0-9]	Data Username dapat diterima jika input data: 1. Boleh menggunakan angka 2. Boleh kombinasi antara angka dengan huruf	[√] Diterima [ ] Ditolak
Password Contoh masukan: *Password*: administrator	[‘A’-‘Z’|’a’-‘z’|0-9]	Data *password* dapat diterima jika input data: 1. Boleh menggunakan angka 2. Boleh kombinasi antara angka dengan huruf	[√] Diterima [ ] Ditolak
Nama Lengkap Contoh masukan: Nama Lengkap: sanusi	[‘A’-‘Z’|’a’-‘z’]	Data nama lengkap dapat diterima jika input data: 1. Nama 3 samapai 30 char 2. Tidak boleh menggunakan angka	[√] Diterima [ ] Ditolak
Email Contoh masukan: *Email*: rion_bdg@yahoo.co	[‘A’-‘Z’|’a’-‘z’|0-9|@|.]	Data *email* dapat diterima jika input data: 1. *Email* 5 samapi	[√] Diterima [ ] Ditolak

m		30 char 2. Boleh menggunakan angka 3. Boleh kombinasi antara angka dengan huruf 4. Harus mengandung simbol "@" dan "." Dapat memunculkan pesan kesalahan bahwa field tersebut harus diisi, jika salah satu field tidak diisi atau salah dalam pengisian.	
No. Telp Contoh masukan: No. Telp: 082260060331	[0-9]	Data No. Telp dapat diterima jika input data: 1. No. Telp Diisi dengan nomor 2. No. Telp harus lebih dari 10 number	[√] Diterima [ ] Ditolak
Blokir Contoh masukan: Blokir: N	Pilih blokir [1|2]	Pengguna memilih status blokir untuk bagian gudang	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Username Contoh masukan: Username:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Password	Data kosong	Dapat	[√] Diterima

Contoh masukan: *Password*:		memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[ ] Ditolak
Nama Lengkap Contoh masukan: Nama Lengkap: Nama lengkap: 12345	[0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
Email Contoh masukan: *Email*: sanusi.com *Email*:	[‘A’-‘Z’|’a’-‘z’|0-9] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau salah dalam pengisian	[√] Diterima [ ] Ditolak
No. Telp Contoh masukan: No. Telp: No. Telp: dghdgdgdg	[‘A’-‘Z’|’a’-‘z’] Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi atau diisi dengan huruf	[√] Diterima [ ] Ditolak
Blokir Contoh masukan: Blokir:	Data kosong	Dapat memunculkan pesan kesalahan, jika field tidak diisi	[√] Diterima [ ] Ditolak

4. Pengujian pengolahan laporan pemesanan

Berikut ini merupakan pengujian yang dilakukan untuk mengolah laporan pemesanan

Tabel 4.33 Pengujian pengolahan laporan pemesanan

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Data status pemesanan Contoh masukan: Status retur :	Pilih status retur [1|2|3..]	Pengguna memilih status pemesanan pada option yang	[√] Diterima [ ] Ditolak

diterima		tersedia	
Dari tanggal Contoh masukan: Dari tanggal : 2012-01-1	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Sampai tanggal Contoh masukan: Sampai tanggal : 2012-06-17	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Data status pemesanan Contoh masukan: Status pemesanan : 	Data kosong	Dapat memunculkan pesan kesalahan, jika pengguna tidak memilih status pemesanan pada option yang tersedia	[√] Diterima [ ] Ditolak
Dari Tanggal Contoh masukan: Dari tanggal : 	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Sampai Tanggal Contoh masukan: Sampai tanggal : 	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak

5. Pengujian pengolahan laporan retur

Berikut ini merupakan pengujian yang dilakukan untuk mengolah laporan retur

Tabel 4.34 Pengujian pengolahan retur

Kasus dan Hasil Uji (Data *valid*)			
Data Masukan	***Valid class***	**Pengamatan**	**Kesimpulan**
Data status retur Contoh masukan: Status retur : diterima	Pilih status retur [1|2|3..]	Pengguna memilih status retur pada option yang tersedia	[√] Diterima [ ] Ditolak
Dari tanggal Contoh masukan: Dari tanggal : 2012-01-1	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Sampai tanggal Contoh masukan: Sampai tanggal : 2012-06-17	[0-9]	Pengguna memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data *invalid*)			
Data Masukan	***Invalid class***	**Pengamatan**	**Kesimpulan**
Data status retur Contoh masukan: Status retur :	Data kosong	Dapat memunculkan pesan kesalahan, jika pengguna tidak memilih status retur pada option yang tersedia	[√] Diterima [ ] Ditolak
Dari Tanggal Contoh masukan: Dari tanggal :	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna tidak memilih tanggal dengan fungsi yang disediakan	[√] Diterima [ ] Ditolak
Sampai Tanggal Contoh masukan: Sampai tanggal :	Data kosong	Dapat memunculkan pesan kesalahan jika pengguna	[√] Diterima [ ] Ditolak

		tidak memilih tanggal dengan fungsi yang disediakan	

4.2.4 *Cause effect testing*

Cause effect testing adalah teknik yang merupakan suplemen untuk equivalence testing yang menyediakan cara untuk memilih kombinasi data input yang melibatkan kondisi input (cause) dan kondisi output (effect) untuk mencegah pendefinisian kasus uji yang terlalu banyak. Adapun langkah-langkah untuk membagi spesifikasi kebutuhan menjadi bagian yang memiliki kemungkinan kerja kemudian mendefinisikan cause effect berdasarkan spesifikasi kebutuhan dan membuat tabel keputusan. Kolom untuk test case dan baris untuk cause/effect, simbol 1 untuk mewakili ya dan simbol 0 untuk tidak.

4.2.4.1 *Cause effect testing* untuk pengunjung dan pelanggan

Teknik ini merupakan suplemen dari untuk *equivalence testing* menyediakan cara untuk memilih kombinasi data *input*, melibatkan kodisi *input (Cause)* dan kondisi *output (effect)* untuk mencegah pendefinisian kasus uji yang terlalu banyak. Berikut ini *cause effect testing* untuk pelanggan dan pengunjung:

1. *Cause effect testing* registrasi pengunjung

Input states:

1. Email memenuhi kondisi yang telah ditentukan

2. Password memenuhi kondisi yang telah ditentukan
3. Ulang password memenuhi kondisi yang telah ditentukan
4. Nama lengkap memenuhi kondisi yang telah ditentukan
5. Alamat memenuhi kondisi yang telah ditentukan
6. Provinsi memenuhi kondisi yang telah ditentukan
7. Kota/kabupaten memenuhi kondisi yang telah ditentukan
8. Kode pos memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database
B. Menampilkan pesan kesalahan dari email yang diinputkan
C. Menampilkan pesan kesalahan dari password yang diinputkan
D. Menampilkan pesan kesalahan dari ulang password yang diinputkan
E. Menampilkan pesan kesalahan dari nama lengkap yang diinputkan
F. Menampilkan pesan kesalahan dari alamat yang diinputkan
G. Menampilkan pesan kesalahan dari provinsi yang diinputkan
H. Menampilkan pesan kesalahan dari kota/kabupaten yang diinputkan
I. Menampilkan pesan kesalahan dari kode pos yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari registrasi pengujung. Tabel *cause effect testing* untuk registrasi pengunjung dapat dilihat pada tabel 4.35

Tabel 4.35 *Cause effect testing* registrasi pengunjung

test case / *cause effect*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	1	0	1	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1	1
3	1	1	1	0	1	1	1	1	1
4	1	1	1	1	0	1	1	1	1
5	1	1	1	1	1	0	1	1	1
6	1	1	1	1	1	1	0	1	1
7	1	1	1	1	1	1	1	0	1
8	1	1	1	1	1	1	1	1	0
A	1	0	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0

E	0	0	0	0	1	0	0	0	0
F	0	0	0	0	0	1	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	0	0	1	0
I	0	0	0	0	0	0	0	0	1

1. *Cause effect testing* login pelanggan

Input states:

1. Email memenuhi kondisi yang telah ditentukan
2. Password memenuhi kondisi yang telah ditentukan

Output states:

A. Login berhasil

B. Menampilkan pesan kesalahan dari email yang diinputkan

C. Menampilkan pesan kesalahan dari password yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari login pelanggan. Tabel *cause effect testing* untuk login pelanggan dapat dilihat pada tabel 4.36

Tabel 4.36 *Cause effect testing* login pelanggan

test case / *cause effect*	**1**	**2**	**3**
1	1	0	1
2	1	1	0
A	1	0	0
B	0	1	0
C	0	0	1

2. *Cause effect testing* ubah data pelanggan

Input states:

1. Nama lengkap memenuhi kondisi yang telah ditentukan
2. Alamat memenuhi kondisi yang telah ditentukan
3. Provinsi memenuhi kondisi yang telah ditentukan
4. Kota/kabupaten memenuhi kondisi yang telah ditentukan
5. Kode pos memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama lengkap yang diinputkan

C. Menampilkan pesan kesalahan dari alamat yang diinputkan

D. Menampilkan pesan kesalahan dari provinsi yang diinputkan

E. Menampilkan pesan kesalahan dari kota/kabupaten yang diinputkan

F. Menampilkan pesan kesalahan dari kode pos yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari ubah data pelanggan. Tabel *cause effect testing* untuk registrasi ubah data pelanggan dapat dilihat pada tabel 4.37

Tabel 4.37 *Cause effect testing* ubah data pelanggan

test case / *cause effect*	**1**	**2**	**3**	**4**	**5**	**6**
1	1	0	1	1	1	1
2	1	1	0	1	1	1
3	1	1	1	0	1	1
4	1	1	1	1	0	1
5	1	1	1	1	1	0
A	1	0	0	0	0	0
B	0	1	0	0	0	0
C	0	0	1	0	0	0

D	0	0	0	1	0	0
E	0	0	0	0	1	0
F	0	0	0	0	0	1

3. *Cause effect testing* ubah password pelanggan

Input states:

1. Password lama memenuhi kondisi yang telah ditentukan
2. Password baru memenuhi kondisi yang telah ditentukan
3. Ulang password memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database
B. Menampilkan pesan kesalahan dari password lama ang diinputkan
C. Menampilkan pesan kesalahan dari password yang diinputkan
D. Menampilkan pesan kesalahan password baru dengan ulang password yang berbeda

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari ubah password pelanggan. Tabel *cause effect testing* untuk ubah password pelanggan dapat dilihat pada tabel 4.38

Tabel 4.38 *Cause effect testing* ubah password pelanggan

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

4. *Cause effect testing* keranjang belanja

Input states:

1. Jumlah memenuhi kondisi yang telah ditentukan

Output states:

A. Disimpan kedalam databse

B. Menampilkan pesan kesalahan dari jumlah yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah

membaca hasil dari uji atau testing dari keranjang belanja. Tabel *cause effect testing* untuk keranjang belanja dapat dilihat pada tabel 4.39

Tabel 4.39 *Cause effect testing* keranjang belanja

test case / *cause effect*	**1**	**2**
1	1	0
A	1	0
B	0	1

5. *Cause effect testing* konfirmasi pemesanan

Input states:

1. Nama pemesan memenuhi kondisi yang telah ditentukan
2. Alamat memenuhi kondisi yang telah ditentukan
3. Provinsi tujuan memenuhi kondisi yang telah ditentukan
4. Kota/kabupaten tujuan memenuhi kondisi yang telah ditentukan
5. Paket pengiriman memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database
B. Menampilkan pesan kesalahan dari nama pemesan yang diinputkan
C. Menampilkan pesan kesalahan dari alamat yang diinputkan
D. Menampilkan pesan kesalahan dari provinsi tujuan yang diinputkan

E. Menampilkan pesan kesalahan dari kota/kabupaten tujuan yang diinputkan

F. Menampilkan pesan kesalahan dari paket pengiriman yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari konfirmasi pemesanan. Tabel *cause effect testing* untuk konfirmasi pemesanan dapat dilihat pada tabel 4.40

Tabel 4.40 *Cause effect testing* konfirmasi pemesanan

test case / ***cause effect***	**1**	**2**	**3**	**4**	**5**	**6**
1	1	0	1	1	1	1
2	1	1	0	1	1	1
3	1	1	1	0	1	1
4	1	1	1	1	0	1
5	1	1	1	1	1	0
A	1	0	0	0	0	0
B	0	1	0	0	0	0
C	0	0	1	0	0	0

D	0	0	0	1	0	0
E	0	0	0	0	1	0
F	0	0	0	0	0	1

6. *Cause effect testing* konfirmasi pembayaran

Input states:

1. Tanggal bayar memenuhi kondisi yang telah ditentukan
2. Bank memenuhi kondisi yang telah ditentukan
3. No. Transaksi memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari tanggal bayar yang diinputkan

C. Menampilkan pesan kesalahan dari bank yang diinputkan

D. Menampilkan pesan kesalahan dari no. Transaksi yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari konfirmasi pembayaran. Tabel *cause effect testing* untuk konfirmasi pembayaran dapat dilihat pada tabel 4.41

Tabel 4.41 *Cause effect testing* konfirmasi pembayaran

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

7. *Cause effect testing* kritik dan saran

Input states:

1. Nama memenuhi kondisi yang telah ditentukan
2. Email memenuhi kondisi yang telah ditentukan
3. Subjek memenuhi kondisi yang telah ditentukan
4. Pesan memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama yang diinputkan

C. Menampilkan pesan kesalahan dari email yang diinputkan

D. Menampilkan pesan kesalahan dari subjek yang diinputkan

E. Menampilkan pesan kesalahan dari pesan yang diinpukan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari kritik dan saran. Tabel *cause effect testing* untuk kritik dan saran dapat dilihat pada tabel 4.42

Tabel 4.42 *Cause effect testing* kritik dan saran

test case / *cause effect*	**1**	**2**	**3**	**4**	**5**
1	1	0	1	1	1
2	1	1	0	1	1
3	1	1	1	0	1
4	1	1	1	1	0
A	1	0	0	0	0
B	0	1	0	0	0
C	0	0	1	0	0

D	0	0	0	1	0
E	0	0	0	0	1

4.2.4.2 *Cause effect testing* untuk bagian gudang

Teknik ini merupakan suplemen dari untuk *equivalence testing* menyediakan cara untuk memilih kombinasi data *input*, melibatkan kodisi *input (Cause)* dan kondisi *output (effect)* untuk mencegah pendefinisian kasus uji yang terlalu banyak. Berikut ini *cause effect testing* untuk bagian gudang:

1. *Cause effect testing* login bagian gudang

Input states:

1. Username memenuhi kondisi yang telah ditentukan
2. Password memenuhi kondisi yang telah ditentukan

Output states:

A. Login berhasil

B. Menampilkan pesan kesalahan dari username yang diinputkan

C. Menampilkan pesan kesalahan dari password yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari login bagian gudang. Tabel *cause effect testing* untuk login bagian gudang dapat dilihat pada tabel 4.43

Tabel 4.43 *Cause effect testing* login bagian gudang

test case / *cause effect*	**1**	**2**	**3**
1	1	0	1
2	1	1	0
A	1	0	0
B	0	1	0
C	0	0	1

2. *Cause effect testing* ubah passwor bagin gudang

Input states:

1. Password lama memenuhi kondisi yang telah ditentukan
2. Password baru memenuhi kondisi yang telah ditentukan
3. Ulang password memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari password lama ang diinputkan

C. Menampilkan pesan kesalahan dari password yang diinputkan

D. Menampilkan pesan kesalahan password baru dengan ulang password yang berbeda

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari ubah password bagian gudang. Tabel *cause effect testing* untuk ubah password bagian gudang dapat dilihat pada tabel 4.44

Tabel 4.44 *Cause effect testing* ubah password bagian gudang

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

3. *Cause effect testing* pengolahan data bank

Input states:

1. Nama bank memenuhi kondisi yang telah ditentukan
2. No. Rekening memenuhi kondisi yang telah ditentukan

3. Nama pemilik memenuhi kondisi yang telah ditentukan
4. Gambar memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database
B. Menampilkan pesan kesalahan dari nama bank yang diinputkan
C. Menampilkan pesan kesalahan dari no. rekening yang diinputkan
D. Menampilkan pesan kesalahan dari nama pemilik yang diinputkan
E. Menampilkan pesan kesalahan dari gambar yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data bank. Tabel *cause effect testing* untuk pengolahan data bank dapat dilihat pada tabel 4.45

Tabel 4.45 *Cause effect testing* pengolahan data bank

test case / ***cause effect***	**1**	**2**	**3**	**4**	**5**
1	1	0	1	1	1
2	1	1	0	1	1
3	1	1	1	0	1
4	1	1	1	1	0
A	1	0	0	0	0

B	0	1		0	0
C	0	0	1	0	
D	0	0	0	1	0
E	0	0	0	0	1

4. *Cause effect testing* pengolahan data ongkos kirim

Input states:

1. Nama kabkota memenuhi kondisi yang telah ditentukan
2. Nama paket memenuhi kondisi yang telah ditentukan
3. Harga memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama kabkota yang diinputkan

C. Menampilkan pesan kesalahan dari nama paket yang diinputkan

D. Menampilkan pesan kesalahan dari harga yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data ongkos kirim. Tabel *cause effect testing* untuk pengolahan data ongkos kirim dapat dilihat pada tabel 4.46

Tabel 4.46 *Cause effect testing* pengolahan data ongkos kirim

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

5. *Cause effect testing* pengolahan data kabkota

Input states:

1. Nama provinsi memenuhi kondisi yang telah ditentukan
2. Nama kabkota memenuhi kondisi yang telah ditentukan
3. Jenis memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama provinsi yang diinputkan

C. Menampilkan pesan kesalahan nama kabkota yang diinputkan

D. Menampilkan pesan kesalahan dari jenis yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data kabkota. Tabel *cause effect testing* untuk pengolahan data kabkota dapat dilihat pada tabel 4.47

Tabel 4.47 *Cause effect testing* pwngolahan data kabkota

test case / ***cause effect***	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

6. *Cause effect testing* pengolahan data kategori

Input states:

1. Nama kategori memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama kategori yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data kategori. Tabel *cause effect testing* untuk pengolahan data kategori dapat dilihat pada tabel 4.48

Tabel 4.48 *Cause effect testing* pengolahan data kategori

test case / *cause effect*	**1**	**2**
1	1	0
A	1	0
B	0	1

7. *Cause effect testing* pengolahan data kurs

Input states:

1. Harga kurs memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari harga kurs yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data kurs. Tabel *cause effect testing* untuk pengolahan data kurs dapat dilihat pada tabel 4.49

Tabel 4.49 *Cause effect testing* pengolahan data kurs

test case / *cause effect*	**1**	**2**
1	1	0
A	1	0
B	0	1

8. *Cause effect testing* pengolahan data pemesanan

Input states:

1. Status pemesanan memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari status pemesanan yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data pemesanan. Tabel *cause effect testing* untuk pengolahan data pemesanan dapat dilihat pada tabel 4.50

Tabel 4.50 *Cause effect testing* pengolahan data pemesanan

test case / *cause effect*	**1**	**2**
1	1	0
A	1	0
B	0	1

9. *Cause effect testing* pengolahan data paket

Input states:

1. Nama paket memenuhi kondisi yang telah ditentukan
2. Lama pengiriman memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama paket yang diinputkan

C. Menampilkan pesan kesalahan dari lama pengiriman yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah

membaca hasil dari uji atau testing dari pengolahan data paket. Tabel *cause effect testing* untuk pengolahan data paket dapat dilihat pada tabel 4.51

Tabel 4.51 *Cause effect testing* pengolahan data paket

test case / *cause effect*	**1**	**2**	**3**
1	1	0	1
2	1	1	0
A	1	0	0
B	0	1	0
C	0	0	1

10. *Cause effect testing* pengolahan data pelanggan

Input states:

1. Aktivasi memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari no. rekening yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah

membaca hasil dari uji atau testing dari pengolahan data pelanggan. Tabel *cause effect testing* untuk pengolahan data pelanggan dapat dilihat pada tabel 4.52

Tabel 4.52 *Cause effect testing* pengolahan data pelanggan

test case / *cause effect*	**1**	**2**
1	1	0
A	1	0
B	0	1

11. *Cause effect testing* pengolahan data produk

Input states:

1. Nama produk memenuhi kondisi yang telah ditentukan
2. Nama kategori memenuhi kondisi yang telah ditentukan
3. Berat memenuhi kondisi yang telah ditentukan
4. Harga memenuhi kondisi yang telah ditentukan
5. Ukuran memenuhi kondisi yang telah ditentukan
6. Stok memenuhi kondisi yang telah ditentukan
7. Deskripsi memenuhi kondisi yang telah ditentukan
8. Gambar memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama produk yang diinputkan

C. Menampilkan pesan kesalahan dari nama kategori yang diinputkan

D. Menampilkan pesan kesalahan dari berat yang diinputkan

E. Menampilkan pesan kesalahan dari harga yang diinputkan

F. Menampilkan pesan kesalahan dari ukuran yang diinputkan

G. Menampilkan pesan kesalahan dari stok yang diinputkan

H. Menampilkan pesan kesalahan dari deskripsi yang diinputkan

I. Menampilkan pesan kesalahan dari gambar yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data produk. Tabel *cause effect testing* untuk pengolahan data produk dapat dilihat pada tabel 4.53

Tabel 4.53 *Cause effect testing* pengolahan data produk

test case / *cause effect*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
1	1	0	1	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1	1
3	1	1	1	0	1	1	1	1	1
4	1	1	1	1	0	1	1	1	1
5	1	1	1	1	1	0	1	1	1

6	1	1	1	1	1	1	0	1	1
7	1	1	1	1	1	1	1	0	1
8	1	1	1	1	1	1	1	1	0
A	1	0	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0	0
D	0	0	0	1	0	0	0	0	0
E	0	0	0	0	1	0	0	0	0
F	0	0	0	0	0	1	0	0	0
G	0	0	0	0	0	0	1	0	0
H	0	0	0	0	0	0	0	1	0
I	0	0	0	0	0	0	0	0	1

12. *Cause effect testing* pengolahan data provinsi

Input states:

1. Nama provinsi memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari nama provinsi yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data provinsi. Tabel *cause effect testing* untuk pengolahan data provinsi dapat dilihat pada tabel 4.54

Tabel 4.54 *Cause effect testing* pengolahan data provinsi

test case / *cause effect*	**1**	**2**
1	1	0
A	1	0
B	0	1

13. *Cause effect testing* pengolahan data retur

Input states:

1. Status retur memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari status retur yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data retur. Tabel *cause effect testing* untuk pengolahan data retur dapat dilihat pada tabel 4.55

Tabel 4.55 *Cause effect testing* pengolahan data retur

test case / *cause effect*	**1**	**2**
1	1	0
A	1	0
B	0	1

14. *Cause effect testing* pengolahan laporan pemesanan

Input states:

1. Status pemesanan memenuhi kondisi yang telah ditentukan
2. Dari tanggal memenuhi kondisi yang telah ditentukan
3. Sampai tanggal memenuhi kondisi yang telah ditentukan

Output states:

A. Menampilkan laporan pemesanan dalam bentuk pdf

B. Menampilkan pesan kesalahan dari status pemesanan yang diinputkan

C. Menampilkan pesan kesalahan dari dari tanggal yang diinputkan

D. Menampilkan pesan kesalahan dari sampai tanggal yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan laporan pemesanan. Tabel *cause effect testing* untuk pengolahan laporan pemesanan dapat dilihat pada tabel 4.56

Tabel 4.56 *Cause effect testing* pengolahan laporan pemesanan

test case / ***cause effect***	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

15. *Cause effect testing* pengolahan laporan retur

Input states:

1. Status retur memenuhi kondisi yang telah ditentukan
2. Dari tanggal memenuhi kondisi yang telah ditentukan
3. Sampai tanggal memenuhi kondisi yang telah ditentukan

Output states:

A. Menampilkan laporan retur dalam bentuk pdf
B. Menampilkan pesan kesalahan dari status retur yang diinputkan
C. Menampilkan pesan kesalahan dari dari tanggal yang diinputkan
D. Menampilkan pesan kesalahan dari sampai tanggal yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan laporan retur. Tabel *cause effect testing* untuk pengolahan laporan retur dapat dilihat pada tabel 4.57

Tabel 4.57 *Cause effect testing* pengolahan laporan retur

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0

A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

4.2.4.3 *Cause effect testing* untuk general manager

Teknik ini merupakan suplemen dari untuk *equivalence testing* menyediakan cara untuk memilih kombinasi data *input*, melibatkan kodisi *input (Cause)* dan kondisi *output (effect)* untuk mencegah pendefinisian kasus uji yang terlalu banyak. Berikut ini *cause effect testing* untuk general manager:

1. *Cause effect testing* login general manager

Input states:

1. Username memenuhi kondisi yang telah ditentukan
2. Password memenuhi kondisi yang telah ditentukan

Output states:

A. Login berhasil

B. Menampilkan pesan kesalahan dari username yang diinputkan

C. Menampilkan pesan kesalahan dari password yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari login general manager. Tabel *cause effect testing* untuk login general manager dapat dilihat pada tabel 4.58

Tabel 4.58 *Cause effect testing* login general manager

test case / *cause effect*	**1**	**2**	**3**
1	1	0	1
2	1	1	0
A	1	0	0
B	0	1	0
C	0	0	1

2. *Cause effect testing* ubah password general manager

Input states:

1. Password lama memenuhi kondisi yang telah ditentukan
2. Password baru memenuhi kondisi yang telah ditentukan
3. Ulang password memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database

B. Menampilkan pesan kesalahan dari password lama ang diinputkan

C. Menampilkan pesan kesalahan dari password yang diinputkan

D. Menampilkan pesan kesalahan password baru dengan ulang password yang berbeda

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari ubah password general manager. Tabel *cause effect testing* untuk ubah password general manager dapat dilihat pada tabel 4.59

Tabel 4.59 *Cause effect testing* ubah password general manager

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

3. *Cause effect testing* pengolahan data bagian gudang

Input states:

1. Username memenuhi kondisi yang telah ditentukan
2. Password memenuhi kondisi yang telah ditentukan
3. Nama lengkap memenuhi kondisi yang telah ditentukan
4. Alamat memenuhi kondisi yang telah ditentukan
5. Email memenuhi kondisi yang telah ditentukan
6. No. Telp memenuhi kondisi yang telah ditentukan
7. Blokir memenuhi kondisi yang telah ditentukan

Output states:

A. Data disimpan kedalam database
B. Menampilkan pesan kesalahan dari username yang diinputkan
C. Menampilkan pesan kesalahan dari password yang diinputkan
D. Menampilkan pesan kesalahan dari nama lengkap yang diinputkan
E. Menampilkan pesan kesalahan dari alamat yang diinputkan
F. Menampilkan pesan kesalahan dari email yang diinputkan
G. Menampilkan pesan kesalahan dari no. telp yang diinputkan
H. Menampilkan pesan kesalahan dari blokir yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan data bagian gudang. Tabel *cause effect testing* untuk pengolahan data bagian gudang dapat dilihat pada tabel 4.60

Tabel 4.60 *Cause effect testing* pengolahan data bagian gudang

test case / *cause effect*	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	1	0	1	1	1	1	1	1
2	1	1	0	1	1	1	1	1
3	1	1	1	0	1	1	1	1
4	1	1	1	1	0	1	1	1
5	1	1	1	1	1	0	1	1
6	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	0
A	1	0	0	0	0	0	0	0
B	0	1	0	0	0	0	0	0
C	0	0	1	0	0	0	0	0
D	0	0	0	1	0	0	0	0
E	0	0	0	0	1	0	0	0

F	0	0	0	0	0	1	0	0
G	0	0	0	0	0	0	1	0
H	0	0	0	0	0	0		1

4. *Cause effect testing* pengolahan laporan pemesanan

Input states:

1. Status pemesanan memenuhi kondisi yang telah ditentukan
2. Dari tanggal memenuhi kondisi yang telah ditentukan
3. Sampai tanggal memenuhi kondisi yang telah ditentukan

Output states:

A. Menampilkan laporan pemesanan dalam bentuk pdf
B. Menampilkan pesan kesalahan dari status pemesanan yang diinputkan
C. Menampilkan pesan kesalahan dari dari tanggal yang diinputkan
D. Menampilkan pesan kesalahan dari sampai tanggal yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan laporan pemesanan. Tabel *cause effect testing* untuk pengolahan laporan pemesanan dapat dilihat pada tabel 4.61

Tabel 4.61 *Cause effect testing* pengolahan laporan pemesanan

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

5. *Cause effect testing* pengolahan laporan retur

Input states:

1. Status retur memenuhi kondisi yang telah ditentukan
2. Dari tanggal memenuhi kondisi yang telah ditentukan
3. Sampai tanggal memenuhi kondisi yang telah ditentukan

Output states:

A. Menampilkan laporan retur dalam bentuk pdf

B. Menampilkan pesan kesalahan dari status retur yang diinputkan

C. Menampilkan pesan kesalahan dari dari tanggal yang diinputkan

D. Menampilkan pesan kesalahan dari sampai tanggal yang diinputkan

Dari *input states* dan *output states* yang didapatkan dan didefinisikan sebelumnya maka dapat ditampilkan tabel *cause effect testing* untuk mempermudah membaca hasil dari uji atau testing dari pengolahan laporan retur. Tabel *cause effect testing* untuk pengolahan lapran retur dapat dilihat pada tabel 4.62

Tabel 4.62 *Cause effect testing* pengolahan laporan retur

test case / *cause effect*	**1**	**2**	**3**	**4**
1	1	0	1	1
2	1	1	0	1
3	1	1	1	0
A	1	0	0	0
B	0	1	0	0
C	0	0	1	0
D	0	0	0	1

4.2.5 Pengujian Beta

Pengujian beta dilakukan pada satu atau lebih pengguna oleh pemakai akhir perangkat lunak, pengujian beta merupakan sebuah aplikasi “live” dari perangkat lunak didalam suatu lingkungan yang tidak dapat di kontrol oleh pengembang.

Pengujian beta dilakukan secara objektif dimana perangkat lunak diuji secara langsung ke lapangan yaitu dengan membuat kuesioner yang ditujukan kepada pengguna *website e-commerce* pada PT. Firebolt Citra Indotama. Pengujian melalui kuesioner berupa pertanyaan mengenai kepuasan user dan kandungan point syarat user friendly untuk selanjutnya dibagikan kepada sebagian user. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk mengambil kesimpulan terhadap penilaian penerapan sistem yang baru.

4.2.5.1 Skenario Pengujian Beta

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

1. Lingkungan PT. Firebolt Citra Indotama

   Pengujian di lingkungan oleh pegawai Firebolt dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

2. Lingkungan Masyarakat Umum

   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.

4.2.5.2 Kuesioner Pengujian Beta Untuk Pengguna

Kuesioner dibagikan langsung kepada para pengguna untuk mengetahui hasil dari pertanyaan-pertanyaan yang ada pada form kuesioner, adapun pertanyaan-pertanyaan yang ada dalam form kuesioner adalah sebagai berikut :

A. Daftar pertanyaan yang diberikan untuk lingkungan masyarakat umum adalah sebagai berikut:

1. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama menampilkan informasi produk yang anda butuhkan?
   a. Sangat setuju
   b. Setuju
   c. Cukup setuju
   d. Kurang setuju
   e. Tidak setuju
2. Apakah anda setuju *website e-commerce* pada PT. Firebolt Citra Indotama mudah digunakan?
   a. Sangat setuju
   b. Setuju
   c. Cukup setuju
   d. Kurang setuju
   e. Tidak setuju
3. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama memudahkan pelanggan untuk melakukan pembelian produk?
   a. Sangat setuju
   d. Kurang setuju

b. Setuju
e. Tidak setuju
c. Cukup setuju

4. Apakah anda setuju bahwa sistem pembayaran yang ada pada *website e-commerce* pada PT. Firebolt Citra Indotama memudahkan anda dalam melakukan pembayaran?
   a. Sangat setuju
   d. Kurang setuju
   b. Setuju
   e. Tidak setuju
   c. Cukup setuju
5. Apakah anda tertarik untuk mengunjungi kembali *website e-commerce* PT. Firebolt Citra Indotama?
   a. Sangat tertarik
   d. Kurang tertarik
   b. Tertarik
   e. Tidak tertarik
   c. Cukup tertarik

Berikut ini adalah hasil prosentase masing-masing jawaban yang diambil dari kuesioner yang telah diisi oleh responden, adapun hasil prosentase dapat diketahui menggunakan rumus : Z = N/Q * 100%.

Keterangan : Z = nilai prosentase
N = banyaknya jawaban tiap pertanyaan
Q = jumlah responden

Kuesioner ini dibagikan kepada 10 orang pengguna di lingkungan masyarakat umum dengan 5 pertanyaan dan menggunakan 5 pilihan jawaban. Berikut ini adalah

hasil prosentase masing-masing jawaban yang ditujukan kepada pengunjung yang sudah dihitung nilainya menggunakan rumus Z=N/Q* 100%.

1. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama menampilkan informasi produk yang anda butuhkan?

Tabel 4.63 Hasil kuesioner untuk pengguna pertanyaan nomor 1

Pilihan	Keterangan	Responden	Prosentase
a	Sangat setuju	1	10%
b	Setuju	6	60%
c	Cukup setuju	3	30%
d	Kurang setuju		
e	Tidak setuju		

2. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama mudah digunakan?

Tabel 4.64 Hasil kuesioner untuk pengguna pertanyaan nomor 2

Pilihan	Keterangan	Responden	Prosentase
a	Sangat setuju		
b	Setuju	7	70%
c	Cukup setuju	3	30%
d	Kurang setuju		
e	Tidak setuju		

3. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama memudahkan pelanggan untuk melakukan pembelian produk?

Tabel 4.65 Hasil kuesioner untuk pengguna pertanyaan nomor 3

Pilihan	Keterangan	Responden	Prosentase
a	Sangat setuju		
b	Setuju	6	60%
c	Cukup setuju	3	30%
d	Kurang setuju	1	10%

e	Tidak setuju		

4. Apakah anda setuju bahwa sistem pembayaran yang ada pada *website e-commerce* pada PT. Firebolt Citra Indotama memudahkan anda dalam melakukan pembayaran?

Tabel 4.66 Hasil kuesioner untuk pengguna pertanyaan nomor 4

Pilihan	Keterangan	Responden	Prosentase
a	Sangat setuju	1	10%
b	Setuju	6	60%
c	Cukup setuju	2	20%
d	Kurang setuju	1	10%
e	Tidak setuju		

5. Apakah anda tertarik untuk mengunjungi kembali *website e-commerce* PT. Firebolt Citra Indotama?

Tabel 4.67 Hasil kuesioner untuk pengguna pertanyaan nomor 5

Pilihan	Keterangan	Responden	Prosentase
a	Sangat tertarik		
b	Tertarik	6	60%
c	Cukup tertarik	3	30%
d	Kurang tertarik	1	10%
e	Tidak tertarik		

B. Daftar pertanyaan yang ditanyakan kepada general manager adalah sebagai berikut:

1. Apakah *website e-commerce* pada PT. Firebolt Citra Indotama dapat memperluas penyebaran informasi produk pada perusahaan?

Jawab: ya, karena aplikasi e-commerce ini dapat diakses dimana saja melalui media internet, memungkinkan masyarakt umum yang tadinya tidak tahu mengenai produk pada perusahaan menjadi tahu.

2. Apakah anda setuju dengan adanya *website e-commerce* pada PT. Firebolt Citra Indotama dapat menambah konsumen pada perusahaan?

   Jawab: saya setuju, dengan adanya aplikasi e-commerce ini memudahkan konsumen dalam melakukan pembelian, dengan cara memesan produk secara langsung tanpa harus datang ke toko yang menyediakan produk firebolt, sehinggan dapat menambah minat para konsumen untuk melakukan pembelian tanpa harus datang ke toko.

3. Apakah *website e-commerce* pada PT. Firebolt Citra Indotama mudah dalam mengelola data bagian gudang?

   Jawab: Ya, pengelolaan data bagian gudang tidak begitu sulit dan mudah dimengerti.

4. Apakah *website e-commerce* pada PT. Firebolt Citra Indotama dapat membantu dalam membuat laporan penjualan?

   Jawab: Sangat membantu, karena sebelumnya dalam membuat laporan masih dilakukan secara manual yang dapat membutuhkan waktu lebih lama.

C. Daftar pertanyaan yang ditanyakan kepada bagian gudang adalah sebagai berikut:

1. Apakah anda setuju bahwa dengan adanya *website e-commerce* pada PT. Firebolt Citra Indotama membantu anda dalam mengelola data produk?

   Jawab: Ya setuju, pengelolaan produk pada aplikasi ini sangat membantu dalam pengelolaan produk yang sebelum di olah secara manual

2. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama mudah dalam mengelola data pelanggan?

   Jawab: Ya setuju, dengan adanya aplikasi ini jadi mempermudah bagi saya untuk mengelola data pelanggan

3. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama mudah dalam mengelola data pemesanan?

   Jawab: Cukup setuju, tidak begitu sulit untuk mengelola data pemesanan

4. Apakah anda setuju bahwa *website e-commerce* pada PT. Firebolt Citra Indotama mudah dalam mengelola data retur?

   Jawab: Cukup setuju, pengelolaan data retur tidak begitu berbeda dengan pengelolaan data pemesanan.

5. Apakah anda setuju bahwa dengan adanya *website e-commerce* pada PT. Firebolt Citra Indotama memudahkan anda dalam membuat laporan?

   Jawab: Ya setuju, pembuatan laporan pada aplikasi begitu mudah dan sangat membantu dalam pembuatan laporan, yang sebelumnya dilakukan

secara manual dengan mengetikan data menggunakan microsoft office excel.

4.2.5.3 Kesimpulan Hasil Pengujian Beta

Berdasarkan hasil yang didapat dari prosentse dengan menggunkan kuesioner kepada pengguna di masyarakat umum dan dilakukannya wawancara pada pengguna di perusahaan pada tahap pengujian beta, maka dapat diambil kesimpulan bahwa pembangunan *website e-commerce* sesuai dengan tujuan yaitu *website e-commerce* PT. Firebolt Citra Indotama dapat memberikan informasi produk perusahaan dan mempermudah proses transaksi pembelian, memudahkan dalam pengolahan data produk, pembuatan laporan dan proses-proses yang terdapat pada aplikasi dapat berjalan dengan baik.