# BAB 4

# IMPLEMENTASI DAN PENGUJIAN

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

## 4. 1. Implementasi Sistem

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

### 4.1.1. Perangkat Keras yang digunakan

Perangkat keras yang digunakan untuk mengimplementasikan perangkat lunak dari aplikasi *e-commerce* di showroom Cardinal. Adapun spesifikasi perangkat keras minimal yang disarankan untuk membangun sistem ini adalah sebagai berikut:

**Tabel 4.1 Perangkat keras yang digunakan**

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 1 | Processor | Intel Pentium 1.6 GHz |
| 2 | Monitor | Montor 15 inch |
| 3 | VGA | VGA On-Board 64 MB |

| No | Perangkat Keras | Spesifikasi |
|---|---|---|
| 4 | Memori | 256 MB |
| 5 | Keyboard | Standar |
| 6 | Mouse | Standar |
| 7 | Printer | Printer hitam-putih |
| 8 | Modem/Koneksi Internet | Bandwidth 64 Kbps |

### 4.1.2. Perangkat Lunak yang digunakan

Perangkat lunak yang digunakan untuk mengimplementasikan aplikasi *e-commerce* di showroom Cardinal adalah sebagai berikut:

**Tabel 4.2 Perangkat lunak yang digunakan**

| No | Perangkat Lunak | Keterangan |
|---|---|---|
| 1 | Sistem operasi | Window XP SP 2 |
| 2 | Bahasa Pemrograman | PHP |
| 3 | Web Server | WAMP |
| 4 | Database server | MySQL |
| 5 | Web Browser | Google Chorme |
| 6 | Code Editor | Adobe Dreamweaver CS 5 |
| 7 | DFD Modeler | Microsoft Visio 2007 |

## 4. 2. Implementasi Basis Data

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

### 4.2.1. Implementasi Tabel Member

Implementasi dari tabel member dapat dilihat pada tabel 4.3

**Tabel 4.3 implementasi tabel member**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| ☐ | id_member | int(11) | | | Tidak | None | | |
| ☐ | id_kota | int(11) | | | Tidak | None | | |
| ☐ | nama_member | varchar(31) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | email_member | varchar(31) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | password_member | varchar(41) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | tanggal_daftar | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | alamat | varchar(51) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | kode_pos | int(11) | | | Tidak | None | | |
| ☐ | telepon | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | telepon_genggam | varchar(21) | latin1_swedish_ci | | Ya | NULL | | |
| ☐ | fax | varchar(21) | latin1_swedish_ci | | Ya | NULL | | |
| ☐ | diaktifkan | varchar(6) | latin1_swedish_ci | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `nama_member` varchar(31) NOT NULL,
  `email_member` varchar(31) NOT NULL,
  `password_member` varchar(41) NOT NULL,
  `tanggal_daftar` varchar(21) NOT NULL,
  `alamat` varchar(51) NOT NULL,
  `kode_pos` int(11) NOT NULL,
  `telepon` varchar(21) NOT NULL,
  `telepon_genggam` varchar(21) DEFAULT NULL,
  `fax` varchar(21) DEFAULT NULL,
  `diaktifkan` varchar(6) NOT NULL,
  PRIMARY KEY (`id_member`),
  UNIQUE KEY `email_member` (`email_member`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB;

ALTER TABLE `member`
  ADD CONSTRAINT `member_ibfk_1` FOREIGN KEY (`id_kota`)
REFERENCES `kota` (`id_kota`);
```

### 4.2.2. Implementasi Tabel Petugas

Implementasi dari tabel petugas dapat dilihat pada tabel 4.4

**Tabel 4.4 implementasi tabel petugas**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| | id_petugas | int(11) | | | No | None | |
| | nama_petugas | varchar(61) | latin1_swedish_ci | | No | None | |
| | email_petugas | varchar(31) | latin1_swedish_ci | | No | None | |
| | password_petugas | varchar(41) | latin1_swedish_ci | | No | None | |
| | level | enum('ADMIN','OPERATOR','PRODUKSI') | latin1_swedish_ci | | No | None | |
| | tanggal_daftar | varchar(21) | latin1_swedish_ci | | No | None | |
| | diaktifkan | varchar(6) | latin1_swedish_ci | | No | None | |

```
CREATE TABLE IF NOT EXISTS `petugas` (
  `id_petugas` int(11) NOT NULL,
  `nama_petugas` varchar(61) NOT NULL,
  `email_petugas` varchar(31) NOT NULL,
  `password_petugas` varchar(41) NOT NULL,
  `level` enum('ADMIN','OPERATOR', PRODUKSI') NOT NULL,
  `tanggal_daftar` varchar(21) NOT NULL,
  `diaktifkan` varchar(6) NOT NULL,
  PRIMARY KEY (`id_petugas`),
  UNIQUE KEY `email_petugas` (`email_petugas`)
) ENGINE=InnoDB;
```

### 4.2.3. Implementasi Tabel Kota

Implementasi dari tabel kota dapat dilihat pada tabel 4.5

**Tabel 4.5 implementasi tabel kota**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| | id_kota | int(11) | | | No | None | |
| | id_provinsi | int(11) | | | No | None | |
| | nama_kota | varchar(21) | latin1_swedish_ci | | No | None | |

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(11) NOT NULL,
  `id_provinsi` int(11) NOT NULL,
  `nama_kota` varchar(21) NOT NULL,
  PRIMARY KEY (`id_kota`),
  KEY `id_provinsi` (`id_provinsi`)
) ENGINE=InnoDB;

ALTER TABLE `kota`
  ADD CONSTRAINT `kota_ibfk_1` FOREIGN KEY
(`id_provinsi`) REFERENCES `provinsi` (`id_provinsi`);
```

### 4.2.4. Implementasi Tabel Provinsi

Implementasi dari tabel provinsi dapat dilihat pada tabel 4.6

**Tabel 4.6 implementasi tabel provinsi**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| | id_provinsi | int(11) | | | No | None | |
| | nama_provinsi | varchar(21) | latin1_swedish_ci | | No | None | |

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(11) NOT NULL,
  `nama_provinsi` varchar(21) NOT NULL,
  PRIMARY KEY (`id_provinsi`)
) ENGINE=InnoDB;
```

### 4.2.5. Implementasi Tabel Barang

Implementasi dari tabel barang dapat dilihat pada tabel 4.7

**Tabel 4.7 implementasi tabel barang**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| | id_barang | varchar(9) | latin1_swedish_ci | | No | None | |
| | id_kategori | int(11) | | | No | None | |
| | nama_barang | varchar(31) | latin1_swedish_ci | | No | None | |
| | harga | double | | | No | None | |
| | diskon | float | | | No | None | |
| | tanggal | varchar(21) | latin1_swedish_ci | | No | None | |
| | deskripsi | text | latin1_swedish_ci | | No | None | |
| | ditampilkan | int(11) | | | No | None | |
| | jumlah_produksi | int(11) | | | No | None | |
| | diproduksi | int(11) | | | No | None | |

```
CREATE TABLE IF NOT EXISTS `barang` (
  `id_barang` varchar(9) NOT NULL,
  `id_kategori` int(11) NOT NULL,
  `nama_barang` varchar(31) NOT NULL,
  `harga` double NOT NULL,
  `diskon` float NOT NULL,
  `tanggal` varchar(21) NOT NULL,
  `deskripsi` text NOT NULL,
  `ditampilkan` int(11) NOT NULL,
  `jumlah_produksi` int(11) NOT NULL,
  `diproduksi` int(11) NOT NULL,
  PRIMARY KEY (`id_barang`),
  KEY `fk_id_kategori` (`id_kategori`)
) ENGINE=InnoDB;

ALTER TABLE `barang`
  ADD CONSTRAINT `fk_id_kategori` FOREIGN KEY
(`id_kategori`) REFERENCES `kategori` (`id_kategori`);
```

### 4.2.6. Implementasi Tabel Detail Barang

Implementasi dari tabel detail barang dapat dilihat pada tabel 4.8

**Tabel 4.8 implementasi tabel detail barang**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| ☐ | id_detail_barang | int(11) | | | Tidak | None | | |
| ☐ | id_barang | varchar(9) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | ukuran | varchar(4) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | berat | float | | | Tidak | None | | |
| ☐ | stok | int(11) | | | Tidak | None | | |
| ☐ | ditampilkan | int(11) | | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `detail_barang` (
  `id_detail_barang` int(11) NOT NULL,
  `id_barang` varchar(9) NOT NULL,
  `ukuran` varchar(4) NOT NULL,
  `berat` float NOT NULL,
  `stok` int(11) NOT NULL,
  `ditampilkan` int(11) NOT NULL,
  PRIMARY KEY (`id_detail_barang`),
  KEY `id_barang` (`id_barang`)
) ENGINE=InnoDB;

ALTER TABLE `detail_barang`
  ADD CONSTRAINT `detail_barang_ibfk_1` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`);
```

### 4.2.7. Implementasi Tabel Gambar

Implementasi dari tabel gambar dapat dilihat pada tabel 4.9

**Tabel 4.9 implementasi tabel gambar**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| | id_gambar | int(11) | | | Tidak | None | | |
| | id_barang | varchar(9) | latin1_swedish_ci | | Tidak | None | | |
| | url_gambar | varchar(61) | latin1_swedish_ci | | Tidak | None | | |
| | nama_gambar | varchar(31) | latin1_swedish_ci | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `gambar` (
  `id_gambar` int(11) NOT NULL,
  `id_barang` varchar(9) NOT NULL,
  `url_gambar` varchar(61) NOT NULL,
  `nama_gambar` varchar(31) NOT NULL,
  PRIMARY KEY (`id_gambar`),
  KEY `id_barang` (`id_barang`)
) ENGINE=InnoDB;

ALTER TABLE `gambar`
  ADD CONSTRAINT `gambar_ibfk_1` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`) ON DELETE
CASCADE ON UPDATE CASCADE;
```

### 4.2.8. Implementasi Tabel Kategori

Implementasi dari tabel kategori dapat dilihat pada tabel 4.10

**Tabel 4.10 implementasi tabel kategori**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| | id_kategori | int(11) | | | Tidak | None | | |
| | nama_kategori | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | deskripsi | text | latin1_swedish_ci | | Tidak | None | | |
| | gambar | varchar(61) | latin1_swedish_ci | | Tidak | None | | |
| | ditampilkan | int(11) | | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL,
  `nama_kategori` varchar(21) NOT NULL,
  `deskripsi` text NOT NULL,
  `gambar` varchar(61) NOT NULL,
  `ditampilkan` int(11) NOT NULL,
  PRIMARY KEY (`id_kategori`)
) ENGINE=InnoDB;
```

### 4.2.9. Implementasi Tabel Komentar

Implementasi dari tabel komentar dapat dilihat pada tabel 4.11

**Tabel 4.11 implementasi tabel komentar**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| ☐ | id_komentar | int(11) | | | Tidak | None | | |
| ☐ | id_barang | varchar(9) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | id_member | int(11) | | | Tidak | None | | |
| ☐ | isi_komentar | text | latin1_swedish_ci | | Tidak | None | | |
| ☐ | tanggal | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | ditampilkan | int(11) | | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `komentar` (
  `id_komentar` int(11) NOT NULL,
  `id_barang` varchar(9) NOT NULL,
  `id_member` int(11) NOT NULL,
  `isi_komentar` text NOT NULL,
  `tanggal` varchar(21) NOT NULL,
  `ditampilkan` int(11) NOT NULL,
  PRIMARY KEY (`id_komentar`),
  KEY `id_barang` (`id_barang`,`id_member`),
  KEY `id_member` (`id_member`)
) ENGINE=InnoDB;

ALTER TABLE `komentar`
  ADD CONSTRAINT `komentar_ibfk_2` FOREIGN KEY
(`id_member`) REFERENCES `member` (`id_member`),
  ADD CONSTRAINT `komentar_ibfk_4` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`);
```

### 4.2.10. Implementasi Tabel Rating

Implementasi dari tabel rating dapat dilihat pada tabel 4.12

**Tabel 4.12 implementasi tabel rating**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| ☐ | id_rating | int(11) | | | Tidak | None | | |
| ☐ | id_barang | varchar(9) | latin1_swedish_ci | | Tidak | None | | |
| ☐ | id_member | int(11) | | | Tidak | None | | |
| ☐ | nilai | int(11) | | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `rating` (
  `id_rating` int(11) NOT NULL,
  `id_barang` varchar(9) NOT NULL,
  `id_member` int(11) NOT NULL,
  `nilai` int(11) NOT NULL,
  PRIMARY KEY (`id_rating`),
  KEY `id_barang` (`id_barang`,`id_member`),
  KEY `id_member` (`id_member`)
) ENGINE=InnoDB;

ALTER TABLE `rating`
  ADD CONSTRAINT `rating_ibfk_2` FOREIGN KEY
(`id_member`) REFERENCES `member` (`id_member`),
  ADD CONSTRAINT `rating_ibfk_3` FOREIGN KEY
(`id_barang`) REFERENCES `barang` (`id_barang`);
```

### 4.2.11. Implementasi Tabel Pesanan

Implementasi dari tabel pesanan dapat dilihat pada tabel 4.13

**Tabel 4.13 implementasi tabel pesanan**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| ☐ | id_pesanan | varchar(12) | latin1_swedish_ci | | No | None | |
| ☐ | id_member | int(11) | | | No | None | |
| ☐ | id_kurs | int(11) | | | No | None | |
| ☐ | id_ongkos_kirim | int(11) | | | No | None | |
| ☐ | tanggal_pesanan | varchar(21) | latin1_swedish_ci | | No | None | |
| ☐ | tanggal_kadaluarsa | varchar(21) | latin1_swedish_ci | | No | None | |
| ☐ | nama_penerima | varchar(51) | latin1_swedish_ci | | No | None | |
| ☐ | alamat | varchar(61) | latin1_swedish_ci | | No | None | |
| ☐ | kode_pos | int(11) | | | No | None | |
| ☐ | sub_total_bayar | double | | | No | None | |
| ☐ | ongkos | double | | | No | None | |
| ☐ | total_ongkos | double | | | No | None | |
| ☐ | total_bayar | double | | | No | None | |
| ☐ | komentar | text | latin1_swedish_ci | | Yes | NULL | |
| ☐ | status | varchar(6) | latin1_swedish_ci | | No | None | |

```
CREATE TABLE IF NOT EXISTS `pesanan` (
  `id_pesanan` varchar(12) NOT NULL,
  `id_member` int(11) NOT NULL,
  `id_kurs` int(11) NOT NULL,
  `id_ongkos_kirim` int(11) NOT NULL,
  `tanggal_pesanan` varchar(21) NOT NULL,
  `tanggal_kadaluarsa` varchar(21) NOT NULL,
  `nama_penerima` varchar(51) NOT NULL,
  `alamat` varchar(61) NOT NULL,
  `kode_pos` int(11) NOT NULL,
  `sub_total_bayar` double NOT NULL,
  `ongkos` double NOT NULL,
  `total_ongkos` double NOT NULL,
  `total_bayar` double NOT NULL,
  `komentar` text,
  `status` varchar(6) NOT NULL,
  PRIMARY KEY (`id_pesanan`),
  KEY `id_member` (`id_member`),
  KEY `id_kurs` (`id_kurs`),
  KEY `id_ongkos_kirim` (`id_ongkos_kirim`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

ALTER TABLE `pesanan`
  ADD CONSTRAINT `pesanan_ibfk_1` FOREIGN KEY
(`id_member`) REFERENCES `member` (`id_member`),
  ADD CONSTRAINT `pesanan_ibfk_2` FOREIGN KEY (`id_kurs`)
REFERENCES `kurs` (`id_kurs`),
  ADD CONSTRAINT `pesanan_ibfk_3` FOREIGN KEY
(`id_ongkos_kirim`) REFERENCES `ongkos_kirim`
(`id_ongkos_kirim`);
```

### 4.2.12. Implementasi Tabel Detail Pesanan

Implementasi dari tabel detail pesanan dapat dilihat pada tabel 4.14

**Tabel 4.14 implementasi tabel detail pesanan**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| ☐ | id_detail_pesanan | int(11) | | | No | None | |
| ☐ | id_pesanan | varchar(12) | latin1_swedish_ci | | No | None | |
| ☐ | id_detail_barang | int(11) | | | No | None | |
| ☐ | harga | double | | | No | None | |
| ☐ | diskon | float | | | No | None | |
| ☐ | jumlah | int(11) | | | No | None | |
| ☐ | jumlah_terpenuhi | int(11) | | | No | None | |

```
CREATE TABLE IF NOT EXISTS `detail_pesanan` (
  `id_detail_pesanan` int(11) NOT NULL,
  `id_pesanan` varchar(12) NOT NULL,
  `id_detail_barang` int(11) NOT NULL,
  `harga` double NOT NULL,
  `diskon` float NOT NULL,
  `jumlah` int(11) NOT NULL,
  `jumlah_terpenuhi` int(11) NOT NULL,
  PRIMARY KEY (`id_detail_pesanan`),
  KEY `id_pesanan` (`id_pesanan`,`id_detail_barang`),
  KEY `id_detail_barang` (`id_detail_barang`)
) ENGINE=InnoDB;

ALTER TABLE `detail_pesanan`
  ADD CONSTRAINT `detail_pesanan_ibfk_2` FOREIGN KEY
(`id_detail_barang`) REFERENCES `detail_barang`
(`id_detail_barang`),
  ADD CONSTRAINT `detail_pesanan_ibfk_3` FOREIGN KEY
(`id_pesanan`) REFERENCES `pesanan` (`id_pesanan`);
```

### 4.2.13. Implementasi Tabel Transaksi

Implementasi dari tabel transaksi dapat dilihat pada tabel 4.15

**Tabel 4.15 implementasi tabel transaksi**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| | id_transaksi | int(11) | | | Tidak | None | | |
| | id_pesanan | varchar(12) | latin1_swedish_ci | | Tidak | None | | |
| | jenis_bayar | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | tanggal_bayar | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | komentar | text | latin1_swedish_ci | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `transaksi` (
  `id_transaksi` int(11) NOT NULL,
  `id_pesanan` varchar(12) NOT NULL,
  `jenis_bayar` varchar(21) NOT NULL,
  `tanggal_bayar` varchar(21) NOT NULL,
  `komentar` text NOT NULL,
  PRIMARY KEY (`id_transaksi`),
  KEY `id_pesanan` (`id_pesanan`)
) ENGINE=InnoDB;

ALTER TABLE `transaksi`
  ADD CONSTRAINT `transaksi_ibfk_1` FOREIGN KEY
(`id_pesanan`) REFERENCES `pesanan` (`id_pesanan`);
```

### 4.2.14. Implementasi Tabel Pengiriman

Implementasi dari tabel pengiriman dapat dilihat pada tabel 4.16

**Tabel 4.16 implementasi tabel pengiriman**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| | id_pengiriman | int(11) | | | Tidak | None | | |
| | id_pesanan | varchar(12) | latin1_swedish_ci | | Tidak | None | | |
| | id_petugas | int(11) | | | Tidak | None | | |
| | no_resi | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | tanggal_kirim | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | tanggal_terima | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | status | varchar(12) | latin1_swedish_ci | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `pengiriman` (
  `id_pengiriman` int(11) NOT NULL,
  `id_pesanan` varchar(12) NOT NULL,
  `id_petugas` int(11) NOT NULL,
  `no_resi` varchar(21) NOT NULL,
  `tanggal_kirim` varchar(21) NOT NULL,
  `tanggal_terima` varchar(21) NOT NULL,
  `status` varchar(12) NOT NULL,
  PRIMARY KEY (`id_pengiriman`),
  KEY `id_pesanan` (`id_pesanan`,`id_petugas`),
  KEY `id_petugas` (`id_petugas`)
) ENGINE=InnoDB;

ALTER TABLE `pengiriman`
  ADD CONSTRAINT `pengiriman_ibfk_2` FOREIGN KEY
(`id_petugas`) REFERENCES `petugas` (`id_petugas`),
  ADD CONSTRAINT `pengiriman_ibfk_3` FOREIGN KEY
(`id_pesanan`) REFERENCES `pesanan` (`id_pesanan`);
```

### 4.2.15. Implementasi Tabel Retur

Implementasi dari tabel retur dapat dilihat pada tabel 4.17

**Tabel 4.17 implementasi tabel retur**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| | id_retur | varchar(12) | latin1_swedish_ci | | Tidak | None | | |
| | id_pesanan | varchar(12) | latin1_swedish_ci | | Tidak | None | | |
| | tanggal_retur | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | resi_retur | varchar(21) | latin1_swedish_ci | | Tidak | None | | |
| | resi_kirim | varchar(21) | latin1_swedish_ci | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` varchar(12) NOT NULL,
  `id_pesanan` varchar(12) NOT NULL,
  `tanggal_retur` varchar(21) NOT NULL,
  `resi_retur` varchar(21) NOT NULL,
  `resi_kirim` varchar(21) NOT NULL,
  PRIMARY KEY (`id_retur`),
  KEY `id_pesanan` (`id_pesanan`)
) ENGINE=InnoDB;

ALTER TABLE `retur`
  ADD CONSTRAINT `retur_ibfk_1` FOREIGN KEY
(`id_pesanan`) REFERENCES `pesanan` (`id_pesanan`);
```

### 4.2.16. Implementasi Tabel Detail Retur

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.18

**Tabel 4.18 implementasi tabel detail retur**

| | Field | Jenis | Penyortiran | Atribut | Kosong | Default | Ekstra | Aksi |
|---|---|---|---|---|---|---|---|---|
| | id_detail_retur | int(11) | | | Tidak | None | | |
| | id_retur | varchar(12) | latin1_swedish_ci | | Tidak | None | | |
| | id_detail_pesanan | int(11) | | | Tidak | None | | |
| | jumlah | int(11) | | | Tidak | None | | |
| | alasan | text | latin1_swedish_ci | | Tidak | None | | |

```
CREATE TABLE IF NOT EXISTS `detail_retur` (
  `id_detail_retur` int(11) NOT NULL,
  `id_retur` varchar(12) NOT NULL,
  `id_detail_pesanan` int(11) NOT NULL,
  `jumlah` int(11) NOT NULL,
  `alasan` text NOT NULL,
  PRIMARY KEY (`id_detail_retur`),
  KEY `id_retur` (`id_retur`),
  KEY `id_detail_pesanan` (`id_detail_pesanan`)
) ENGINE=InnoDB;

ALTER TABLE `detail_retur`
  ADD CONSTRAINT `detail_retur_ibfk_1` FOREIGN KEY
(`id_retur`) REFERENCES `retur` (`id_retur`),
  ADD CONSTRAINT `detail_retur_ibfk_2` FOREIGN KEY
(`id_detail_pesanan`) REFERENCES `detail_pesanan`
(`id_detail_barang`);
```

### 4.2.17. Implementasi Tabel kurs

Implementasi dari tabel kurs dapat dilihat pada tabel 4.19

**Tabel 4.19 implementasi tabel kurs**

| | Field | Type | Collation | Attributes | Null | Default | Extra | Action |
|---|---|---|---|---|---|---|---|---|
| | id_kurs | int(11) | | | No | None | | |
| | id_petugas | int(11) | | | No | None | | |
| | nama_kurs | varchar(21) | latin1_swedish_ci | | No | None | | |
| | nilai_kurs | double | | | No | None | | |

```
CREATE TABLE IF NOT EXISTS `kurs` (
  `id_kurs` int(11) NOT NULL,
  `id_petugas` int(11) NOT NULL,
  `nama_kurs` varchar(21) NOT NULL,
  `nilai_kurs` double NOT NULL,
  PRIMARY KEY (`id_kurs`),
  KEY `id_petugas` (`id_petugas`)
) ENGINE=InnoDB;

ALTER TABLE `kurs`
  ADD CONSTRAINT `kurs_ibfk_1` FOREIGN KEY (`id_petugas`)
REFERENCES `petugas` (`id_petugas`);
```

## 4.2.18. Implementasi Tabel jasa pengiriman

Implementasi dari tabel jasa pengiriman dapat dilihat pada tabel 4.20

**Tabel 4.20 implementasi tabel jasa pengiriman**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| | id_jasa_pengiriman | int(11) | | | No | None | |
| | nama_jasa_pengiriman | varchar(21) | latin1_swedish_ci | | No | None | |
| | url_pengiriman | varchar(250) | latin1_swedish_ci | | No | None | |

```
CREATE TABLE IF NOT EXISTS `jasa_pengiriman` (
  `id_jasa_pengiriman` int(11) NOT NULL,
  `nama_jasa_pengiriman` varchar(21) NOT NULL,
  `url_pengiriman` varchar(250) NOT NULL,
  PRIMARY KEY (`id_jasa_pengiriman`)
) ENGINE=InnoDB;
```

## 4.2.19. Implementasi Tabel paket pengiriman

Implementasi dari tabel jasa pengiriman dapat dilihat pada tabel 4.21

**Tabel 4.21 implementasi tabel paket pengiriman**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| | id_paket_pengiriman | int(11) | | | No | None | |
| | id_jasa_pengiriman | int(11) | | | No | None | |
| | nama_paket_pengiriman | varchar(21) | latin1_swedish_ci | | No | None | |

```
CREATE TABLE IF NOT EXISTS `paket_pengiriman` (
  `id_paket_pengiriman` int(11) NOT NULL,
  `id_jasa_pengiriman` int(11) NOT NULL,
  `nama_paket_pengiriman` varchar(21) NOT NULL,
  PRIMARY KEY (`id_paket_pengiriman`),
  KEY `id_jasa_pengiriman` (`id_jasa_pengiriman`)
) ENGINE=InnoDB;

ALTER TABLE `paket_pengiriman`
  ADD CONSTRAINT `paket_pengiriman_ibfk_1` FOREIGN KEY
(`id_jasa_pengiriman`) REFERENCES `jasa_pengiriman`
(`id_jasa_pengiriman`);
```

## 4.2.20. Implementasi Tabel ongkos kirim

Implementasi dari tabel jasa pengiriman dapat dilihat pada tabel 4.22

**Tabel 4.22 implementasi tabel ongkos kirim**

| | Field | Type | Collation | Attributes | Null | Default | Extra |
|---|---|---|---|---|---|---|---|
| | id_ongkos_kirim | int(11) | | | No | None | |
| | id_paket_pengiriman | int(11) | | | No | None | |
| | id_kota | int(11) | | | No | None | |
| | ongkos | double | | | No | None | |

```
CREATE TABLE IF NOT EXISTS `ongkos_kirim` (
  `id_ongkos_kirim` int(11) NOT NULL,
  `id_paket_pengiriman` int(11) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `ongkos` double NOT NULL,
  PRIMARY KEY (`id_ongkos_kirim`),
  KEY `id_kota` (`id_kota`),
  KEY `id_paket_pengiriman` (`id_paket_pengiriman`)
) ENGINE=InnoDB;

ALTER TABLE `ongkos_kirim`
  ADD CONSTRAINT `ongkos_kirim_ibfk_2` FOREIGN KEY
(`id_kota`) REFERENCES `kota` (`id_kota`),
  ADD CONSTRAINT `ongkos_kirim_ibfk_3` FOREIGN KEY
(`id_paket_pengiriman`) REFERENCES `paket_pengiriman`
(`id_paket_pengiriman`);
```

## 4. 3. Implementasi Antar Muka

Implementasi antar muka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengunjung dan member pada tabel 4.23, operator pada tabel 4.24, dan administrator pada tabel 4.25. Adapun untuk tampilan implementasi pembangunan aplikasi *e-commerce* di showroom cardinal akan dilampirkan pada halaman lampiran.

**Tabel 4.23 Implementasi Antar Muka Pengunjung Dan Member**

| No | Menu | Deskripsi | Nama *File* |
|---|---|---|---|
| 1 | beranda | Digunakan oleh pengunjung atau member untuk masuk kehalaman utama | showproduct.php |
| 2 | kategori | Digunakan oleh pengunjung atau member untuk masuk kehalaman kategori | showkategori.php |
| 3 | informasi | Digunakan oleh pengunjung atau member untuk masuk kehalaman informasi | informasi.php |
| 4 | pencarian | Digunakan oleh pengunjung atau member untuk masuk kehalaman pencarian | cari.php |
| 5 | registrasi | Digunakan oleh pengunjung untuk melakukan pendaftaran menjadi | registermember.php |

| | | | |
|---|---|---|---|
| | | member baru | |
| 6 | login | Digunakan oleh member untuk melakukan login sebagai member | loginmember.php |
| 7 | Lupa *password* | Digunakan jika member lupa dengan *password* | lupapass.php |
| 8 | Lihat profil | Digunakan untuk melihat informasi profil member | lihatprofil.php |
| 9 | Ubah profil | Digunakan untuk mengganti informasi profil member | ubahmember.php |
| 10 | Ubah *password* | Digunakan untuk mengganti *password* member | ubahpass.php |
| 11 | Detail produk | Digunakan untuk melihat detail barang | showdetail.php |
| 12 | Pesanan | Digunakan untuk melihat history pesanan yang pernah dilakukan | historylist.php |
| 13 | Detail Pesanan | Digunakan untuk melihat history detail pesanan yang pernah dilakukan | historyorder.php |
| 14 | Keranjang belanja | Digunakan untuk melihat, merubah, menghapus daftar pesanan yang akan dilakukan | showcart.php |
| 14 | Logout | Digunakan member untuk keluar dari akunya | logout.php |

**Tabel 4.24 Implementasi Antar Muka Operator**

| No | Menu | Deskripsi | Nama *File* |
|---|---|---|---|
| 1 | *Login* operator | Digunakan sebagai halaman *login* operator | loginform.php |
| 2 | Data barang | Digunakan untuk melihat, menambah, dan merubah data barang | databarang.php |
| 3 | Data kategori | Digunakan untuk melihat, menambah, dan merubah data kategori | datakategori.php |
| 4 | Data Kota | Digunakan untuk melihat, menambah dan merubah data kota dan ongkos kirim | datakota.php |
| 5 | Data provinsi | Digunakan untuk melihat, menambah dan merubah data provinsi | dataprovinsi.php |
| 6 | Komentar | Digunakan untuk melihat dan mengelola data komentar pada barang | datakomentar.php |
| 7 | pemesanan | Digunakan untuk melihat dan mengolah data pesanan | datapesanan.php |
| 8 | pengiriman | Digunakan untuk melihat dan mengolah data pengiriman | pengiriman.php |

| No | Menu | Deskripsi | Nama *File* |
|---|---|---|---|
| 9 | Ubah *password* | Digunakan untuk mengubah data *password* | ubahpassword.php |
| 10 | Logout | Digunakan operator untuk dapat keluar dari halaman operator | logout.php |
| 11 | beranda | Digunakan operator untuk masuk kehalaman utama | index.php |

**Tabel 4.25 Implementasi Antar Muka Administrator**

| No | Menu | Deskripsi | Nama *File* |
|---|---|---|---|
| 1 | Login Administrator | Digunakan sebagai halaman *login* administrator | loginform.php |
| 2 | Data petugas | Digunakan untuk melihat, menambah, dan merubah data | datapetugas.php |
| 3 | Data member | Digunakan untuk meliha dan merubaht data member | datamember.php |
| 4 | Lupa *password* | Digunakan jika administrator lupa dengan *password* | ubahpassword.php |
| 5 | Ubah *password* | Digunakan untuk mengubah data *password* | ubahpassword.php |
| 6 | *Logout* | Digunakan administrator untuk keluar dari halaman administrator | logout.php |
| 7 | beranda | Digunakan operator untuk masuk kehalaman utama | index.php |

## 4. 4. Pengujian Perangkat Lunak

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

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

## 4. 5. Pengujian Alpa

Pengujian alpa dilakukan menggunakan metode *black box*. Untuk menentukan pengujian alpa ini dibagi menjadi tiga, yaitu skenario pengujian

halaman member, skenario pengujian halaman administrator dan skenario pengujian aplikasi operator.

### 4.5.1. Skenario Pengujian Alpa

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi member maka dapat dilihat pada Tabel 4.26 sebagai berikut:

**Tabel 4.26 Skenario Pengujian Aplikasi Member**

| Kelas Uji | Butir Uji | Jenis Pengujian |
|---|---|---|
| Lupa *password* | Menampilkan *form* lupa *password* | *Black Box* |
| Registrasi | Menampilkan *form* registrasi member | *Black Box* |
| | Isi *form* daftar member | *Black Box* |
| | Simpan data member baru | *Black Box* |
| *Login* | Isi data *login* | *Black Box* |
| | Verifikasi *e-mail* dan *password* | *Black Box* |
| Ubah profil | Menampilkan *form* ubah profil | *Black Box* |
| | Ubah data profil | *Black Box* |
| | Simpan perubahan | *Black Box* |
| Ubah *password* | Menampilkan form ubah *password* | *Black Box* |
| | Isi *form* ubah *password* | *Black Box* |
| | Simpan data baru | *Black Box* |
| Komentar | Menampilkan komentar barang | *Black Box* |
| Keranjang belanja | Menampilkan data pembelian | *Black Box* |
| | Mengubah data pembelian | *Black Box* |
| | Menghapus data pembelian | *Black Box* |
| | Menampilkan konfirmasi pembelian | *Black Box* |
| Isi *form* alamat tujuan baru | Menampilkan *form* alamat tujuan baru | *Black Box* |
| Isi *form* transaksi | Menampilkan *form* transaksi | *Black Box* |

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi operator maka dapat dilihat pada Tabel 4.27 sebagai berikut:

**Tabel 4.27 Skenario Pengujian Aplikasi Operator**

| Kelas Uji | Butir Uji | Jenis Pengujian |
|---|---|---|
| *Login* | Isi data *login* | *Black Box* |
| | Verifikasi *e-mail* dan *password* | *Black Box* |
| Data barang | Menampilkan data barang | *Black Box* |
| | Tambah data barang | *Black Box* |
| | Ubah data barang | *Black Box* |
| Data detail barang | Menampilkan data detail barang | *Black Box* |
| | Tambah data detail barang | *Black Box* |
| | Ubah data detail barang | *Black Box* |
| Data kategori | Menampilkan data kategori | *Black Box* |
| | Tambah data kategori | *Black Box* |
| | Ubah data kategori | *Black Box* |
| Data Kota | Menampilkan data ongkos kirim | *Black Box* |
| | Menambah data ongkos kirim | *Black Box* |
| | Mengubah data ongkos kirim | *Black Box* |
| Data Provinsi | Menampilkan data provinsi | *Black Box* |
| | Menambah data provinsi | *Black Box* |
| | Mengubah data provinsi | *Black Box* |
| Komentar | Menampilkan data komentar | *Black Box* |
| | Ubah status komentar | *Black Box* |
| pemesanan | Menampilkan data pesanan | *Black Box* |
| | Cari data pesanan | *Black Box* |
| | Ubah status pesanan | *Black Box* |
| pengiriman | Menampilkan data pengiriman | *Black Box* |
| | Cari data pengiriman | *Black Box* |
| | Ubah data pengiriman | *Black Box* |
| Ubah *password* | Menampilkan *form* ubah *password* | *Black Box* |
| | Isi *form* ubah *password* | *Black Box* |
| | Simpan data baru | *Black Box* |

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi administrator maka dapat dilihat pada Tabel 4.28 sebagai berikut:

**Tabel 4.28 Skenario Pengujian Aplikasi Admin**

| Kelas Uji | Butir Uji | Jenis Pengujian |
|---|---|---|
| *Login* | Isi data *login* | *Black Box* |
| | Verifikasi *e-mail* dan *password* | *Black Box* |
| Data petugas | Menampilkan data petugas | *Black Box* |
| | Tambah data petugas | *Black Box* |
| | Ubah data petugas | *Black Box* |
| Data Member | Menampilkan data member | *Black Box* |
| | Ubah data member | *Black Box* |
| Ganti *password* | Menampilkan *form* ganti *password* | *Black Box* |
| | Isi *form* ganti *password* | *Black Box* |
| | Simpan data baru | *Black Box* |

## 4.5.2. Kasus Dan Hasil Pengujian

Berdasarkan rencana pengujian yang disusun, maka dilakukan pengujian sebagai berikut.

### 4.5.2.1. Pengujian Lupa Password Member

Untuk penjelasan lebih lanjut tentang pengujian data benar lupa password member maka dapat dilihat pada Tabel 4.29 sebagai berikut:

**Tabel 4.29 Pengujian Data Benar Lupa Password Member**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | Pengamatan | Kesimpulan |
| Mengisi e-mail | Tercantum pada text box pada *field e-mail* | Dapat tercantum pada text box pada *field e-mail* | [√ ] Diterima<br>[ ] Ditolak |
| Email : slow_abiez@yahoo.com | Menampilkan data masukkan pada *field e-mail* | Dapat menampilkan data masukkan pada *field e-mail* | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah lupa password member maka dapat dilihat pada Tabel 4.30 sebagai berikut:

**Tabel 4.30 Pengujian Data Salah Lupa Password Member**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | Pengamatan | Kesimpulan |
| Format email tidak valid, contoh : Email : slow_abiez.com | Menampilkan pesan kesalahan "e-mail tidak valid" | Dapat menampilkan pesan kesalahan "e-mail tidak valid" | [√ ] Diterima [ ] Ditolak |
| Email tidak ditemukan dalam database | Menampilkan pesan kesalahan "ERROR : E-mail belum terdaftar!" | Dapat menampilkan pesan kesalahan "ERROR : E-mail belum terdaftar!" | [√ ] Diterima [ ] Ditolak |

### 4.5.2.2. Pengujian Registrasi Member

Untuk penjelasan lebih lanjut tentang pengujian data benar registrasi member maka dapat dilihat pada Tabel 4.31 sebagai berikut:

**Tabel 4.31 Pengujian Data Benar Registrasi Member**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | Pengamatan | Kesimpulan |
| Mengisi data registrasi member, contoh : Email: slow_abiez@yahoo.com Password: 123456 Nama Lengkap : Andreas Saputra Alamat : Jl. Santosa Asih II no 19 Provinsi : Jawa Barat Kota : Bandung Kode Pos : 40292 Kode gambar: h1l8u | Jika semua *field* terisi dan benar dalam pengisian, maka akan dapat melanjutkan proses registrasi | Form registrasi terisi dan benar dalam pengisian, maka akan dapat melanjutkan proses registrasi | [√ ] Diterima [ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah lupa password member maka dapat dilihat pada Tabel 4.32 sebagai berikut:

**Tabel 4.32 Pengujian Data Salah Registrasi Member**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | Pengamatan | Kesimpulan |
| Mengisi email yang sudah terdaftar | Tampilkan pesan kesalahan "Registrasi Gagal. Email yang anda masukan sudah terdaftar" | Dapat menampilkan pesan kesalahan "Registrasi Gagal. Email yang anda masukan sudah terdaftar" | [√ ] Diterima<br>[ ] Ditolak |
| Mengisi email yang dengan format tidak valid, contoh :<br>Email baru : slow_abiez.com | Tampilkan pesan kesalahan "Email tidak valid" | Dapat menampilkan pesan kesalahan "Email tidak valid" | [√ ] Diterima<br>[ ] Ditolak |
| Mengisi password kurang dari 6 karakter, contoh :<br>Password: 1234 | Tampilkan pesan kesalahan "Password minimal 6 karakter" | Dapat menampilkan pesan kesalahan "Password minimal 6 karakter" | [√ ] Diterima<br>[ ] Ditolak |
| Mengisi konfirmasi password yang berbeda dengan password,<br>contoh :<br>Password:123456<br>Konfirmasi Pasword : 654321 | Tampilkan pesan kesalahan "password tidak sesuai" | Dapat menampilkan pesan kesalahan "password tidak sesuai" | [√ ] Diterima<br>[ ] Ditolak |
| Format kode pos tidak sesuai, contoh :<br>Kode Pos : abcde | Tampilkan pesan kesalahan "kode pos salah" | Dapat menampilkan pesan kesalahan "kode pos salah" | [√ ] Diterima<br>[ ] Ditolak |

| Data Masukan | Yang diharapkan | Pengamatan | Kesimpulan |
|---|---|---|---|
| Memasukkan data kosong | Menampilkan pesan kesalahan "nama harus diisi", "email harus diisi", "password harus diisi", "alamat harus diisi", "provinsi harus diisi", "kota harus diisi", "kode pos harus diisi", "Perhitungan dibutuhkan" | Dapat menampilkan pesan kesalahan "nama harus diisi", "email harus diisi", "password harus diisi", "alamat harus diisi", "provinsi harus diisi", "kota harus diisi", "kode pos harus diisi", "Perhitungan dibutuhkan" | [√ ] Diterima<br>[ ] Ditolak |

### 4.5.2.3. Pengujian Login Member

Untuk penjelasan lebih lanjut tentang pengujian data benar login member maka dapat dilihat pada Tabel 4.33 sebagai berikut:

**Tabel 4.33 Pengujian Data Benar Login member**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | Pengamatan | Kesimpulan |
| Memasukan alamat *email* : slow_abiez@yahoo.com | Mengisikan data alamat *email* pada *field* alamat email | Dapat mengisi *field* alamat *email* pada *field* alamat *email* | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan Password : 123456 | Mengisikan data password pada field password | Mengisikan data password pada field password | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah login member maka dapat dilihat pada Tabel 4.34 sebagai berikut:

**Tabel 4.34 Pengujian Data Salah Login member**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | Pengamatan | Kesimpulan |
| Memasukan data *email* dan *password* yang tidak sesuai | Tampilkan pesan kesalahan "E-mail atau password salah" | Dapat menampilkan pesan kesalahan "E-mail atau password salah" | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan data *email* atau *password* yang tidak terdaftar | Tampilkan pesan kesalahan "E-mail atau password salah" | Dapat menampilkan pesan kesalahan "E-mail atau password salah" | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan data *email* yang tidak valid, contoh : Email : slow_abiez.com | Tampilkan pesan kesalahan "e-mail tidak valid" | Dapat menampilkan pesan kesalahan "e-mail tidak valid" | [√ ] Diterima<br>[ ] Ditolak |
| Mengosongkan data *email* atau *password* | Tampilkan pesan kesalahan "Email Atau Password tidak Lengkap" | Dapat menampilkan pesan kesalahan "Email Atau Password tidak Lengkap" | [√ ] Diterima<br>[ ] Ditolak |

### 4.5.2.4. Pengujian Ubah Profil

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah profil maka dapat dilihat pada Tabel 4.35 sebagai berikut:

**Tabel 4.35 Pengujian Data Benar Ubah Profil**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| mengubah data profil member, contoh :<br>Nama Lengkap : Andreas Saputra<br>Alamat : Jl. Santosa Asih II | Mengisi *field* data yang akan diubah dengan benar | Dapat mengisi *field* data yang akan diubah dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah profil maka dapat dilihat pada Tabel 4.36 sebagai berikut:

**Tabel 4.36 Pengujian Data Salah Ubah Profil**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Nama kurang dari 3 karakter, contoh : nama : a | Tampilkan pesan kesalahan "nama tidak boleh kurang dari 3 karakter" | Dapat menampilkan pesan kesalahan "nama tidak boleh kurang dari 3 karakter" | [√ ] Diterima [ ] Ditolak |
| Format kode pos tidak sesuai, contoh : Kode Pos : abcde | Tampilkan pesan kesalahan "kode pos salah" | Dapat menampilkan pesan kesalahan "kode pos salah" | [√ ] Diterima [ ] Ditolak |

## 4.5.2.5. Pengujian Ubah Password

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah password maka dapat dilihat pada Tabel 4.37 sebagai berikut:

**Tabel 4.37 Pengujian Data Benar Ubah password**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengisi data ubah password dengan benar, contoh : Password lama : 123456 Password baru: 654321 Konfirmasi password baru: 654321 | Mengisi *field* data yang akan diubah dengan benar | Dapat mengisi *field* data yang akan diubah dengan benar | [√ ] Diterima [ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah password maka dapat dilihat pada Tabel 4.38 sebagai berikut:

**Tabel 4.38 Pengujian Data Salah Ubah password**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| password kurang dari 6 karakter, contoh : nama : 12345 | Tampilkan pesan kesalahan "password tidak boleh kurang dari 6 karakter" | Dapat menampilkan pesan kesalahan "password tidak boleh kurang dari 6 karakter" | [√ ] Diterima<br>[ ] Ditolak |
| Konfirmasi password baru tidak sama dengan password baru<br>Contoh:<br>Password baru: 654321<br>Konfirmasi password baru: 754321 | Tampilkan pesan kesalahan "password tidak sesuai" | Dapat menampilkan pesan kesalahan "password tidak sesuai" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.6. Pengujian Keranjang Belanja

Untuk penjelasan lebih lanjut tentang pengujian data benar Keranjang Belanja maka dapat dilihat pada Tabel 4.39 sebagai berikut:

**Tabel 4.39 Pengujian Data Benar Keranjang Belanja**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan jumlah pembelian diantara 1 sampai 999 | Mengisi *field* data yang akan diubah dengan benar | Dapat mengisi *field* data yang akan diubah dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah Keranjang Belanja maka dapat dilihat pada Tabel 4.40 sebagai berikut:

**Tabel 4.40 Pengujian Data Salah Keranjang Belanja**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan jumlah pembelian kurang dari 1 atau lebih dari 999 | Tampilkan pesan kesalahan "Ada produk yang gagal diubah!, Jumlah pesanan harus diantara 0 dan 999" | Dapat menampilkan pesan kesalahan "Ada produk yang gagal diubah!, Jumlah pesanan harus diantara 0 dan 999" | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan jumlah pembelian yang melebihi stok tersedia | Tampilkan pesan kesalahan "stok yang dipesan melebihi stok tersedia" | Dapat menampilkan pesan kesalahan "stok yang dipesan melebihi stok tersedia" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.7. Pengujian Alamat Baru

Untuk penjelasan lebih lanjut tentang pengujian data benar Alamat Baru maka dapat dilihat pada Tabel 4.41 sebagai berikut:

**Tabel 4.41 Pengujian Data Benar Alamat Baru**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data alamat baru dengan benar<br>Contoh:<br>Nama Penerima: Andreas Saputra<br>Alamat: Jl. Santosa Asih II no 19<br>Provinsi: Jawa Barat<br>Kode Pos: 40292 | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah Alamat Baru maka dapat dilihat pada Tabel 4.42 sebagai berikut:

**Tabel 4.42 Pengujian Data Salah Alamat Baru**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan nama penerima kurang dari 3 karakter<br>Contoh:<br>Nama penerima: a | Tampilkan pesan kesalahan “nama harus lebih dari 3 karakter” | Dapat menampilkan pesan kesalahan “nama harus lebih dari 3 karakter” | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan alamat kurang dari 3 karakter<br>Contoh:<br>alamat: a | Tampilkan pesan kesalahan “alamat harus lebih dari 3 karakter” | Dapat menampilkan pesan kesalahan “alamat harus lebih dari 3 karakter” | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan kode pos kurang dari 5 karakter<br>Contoh:<br>alamat: 123 | Tampilkan pesan kesalahan “kode pos salah” | Dapat menampilkan pesan kesalahan “kode pos salah” | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan kode pos huruf<br>Contoh:<br>alamat: abcde | Tampilkan pesan kesalahan “format kode pos salah” | Dapat menampilkan pesan kesalahan “format kode pos salah” | [√ ] Diterima<br>[ ] Ditolak |

### 4.5.2.8. Pengujian Transaksi

Untuk penjelasan lebih lanjut tentang pengujian data benar transaksi maka dapat dilihat pada Tabel 4.43 sebagai berikut:

**Tabel 4.43 Pengujian Data Benar transaksi**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data transaksi dengan benar<br>Contoh:<br>Jumlah transaksi: 100000<br>Nama bank: BCA<br>Jenis transaksi: transfer<br>Tanggal transaksi: 30/06/2011<br>Nama pengirim: Andreas Saputra<br>Bank penerima: BCA 4381332084 A/N Andreas Saputra | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah transaksi maka dapat dilihat pada Tabel 4.44 sebagai berikut:

**Tabel 4.44 Pengujian Data Salah transaksi**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan jumlah transaksi kurang dari jumlah bayar | Tampilkan pesan kesalahan “jumlah bayar kurang” | Dapat menampilkan pesan kesalahan “jumlah bayar kurang” | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan jumlah transaksi dengan huruf | Tampilkan pesan kesalahan “jumlah bayar harus angka” | Dapat menampilkan pesan kesalahan “jumlah bayar harus angka” | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan tanggal transaksi diluar batas pesanan dan kadaluarsa | Tampilkan pesan kesalahan “tanggal transaksi tidak valid” | Dapat menampilkan pesan kesalahan “tanggal transaksi tidak valid” | [√ ] Diterima<br>[ ] Ditolak |

### 4.5.2.9. Pengujian login operator/admin

Untuk penjelasan lebih lanjut tentang pengujian data benar login operator/admin maka dapat dilihat pada Tabel 4.45 sebagai berikut:

**Tabel 4.45 Pengujian Data Benar login operator/admin**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data login dengan benar<br>Contoh:<br>Username: slow.abiez@gmail.com<br>Password: 123456 | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah login operator/admin maka dapat dilihat pada Tabel 4.46 sebagai berikut:

**Tabel 4.46 Pengujian Data Salah login operator/admin**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data login yang salah/tidak terdaftar | Tampilkan pesan kesalahan “username atau password salah” | Dapat menampilkan pesan kesalahan “username atau password salah” | [√ ] Diterima<br>[ ] Ditolak |
| Mengosongkan masukan data login | Tampilkan pesan kesalahan “username atau password tidak lengkap” | Dapat menampilkan pesan kesalahan “username atau password tidak lengkap” | [√ ] Diterima<br>[ ] Ditolak |

### 4.5.2.10. Pengujian tambah data barang

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah data barang maka dapat dilihat pada Tabel 4.47 sebagai berikut:

**Tabel 4.47 Pengujian Data Benar tambah data barang**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data barang dengan benar<br>Contoh:<br>Nama barang: Condor<br>Kategori: Formal<br>Harga: 100000<br>Diskon: 0<br>Deskripsi:<br>Ditampilkan: ya | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah data barang maka dapat dilihat pada Tabel 4.48 sebagai berikut:

**Tabel 4.48 Pengujian Data Salah tambah data barang**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data barang | Tampilkan pesan kesalahan “data barang gagal ditambahkan, masukan data barang dengan lengkap” | Dapat menampilkan pesan kesalahan “data barang gagal ditambahkan, masukan data barang dengan lengkap” | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.11. Pengujian ubah data barang

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah data barang maka dapat dilihat pada Tabel 4.49 sebagai berikut:

**Tabel 4.49 Pengujian Data Benar ubah data barang**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data barang dengan benar<br>Contoh:<br>Nama barang: Condor<br>Kategori: Formal<br>Harga: 100000<br>Diskon: 0<br>ditampilkan: ya | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah data barang maka dapat dilihat pada Tabel 4.50 sebagai berikut:

**Tabel 4.50 Pengujian Data Salah ubah data barang**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data barang | Tampilkan pesan kesalahan "data barang gagal diubah, masukan data barang dengan lengkap" | Dapat menampilkan pesan kesalahan "data barang gagal di ubah, masukan data barang dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.12. Pengujian tambah data kategori

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah data kategori maka dapat dilihat pada Tabel 4.51 sebagai berikut:

**Tabel 4.51 Pengujian Data Benar tambah data kategori**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data kategori dengan benar<br>Contoh:<br>Nama kategori: Denim<br>Deskripsi:<br>Ditampilkan: ya | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah data kategori maka dapat dilihat pada Tabel 4.52 sebagai berikut:

**Tabel 4.52 Pengujian Data Salah tambah data kategori**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data kategori | Tampilkan pesan kesalahan "data kategori gagal ditambahkan, masukan data kategori dengan lengkap" | Dapat menampilkan pesan kesalahan "data kategori gagal ditambahkan, masukan data kategori dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.13. Pengujian ubah data kategori

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah data kategori maka dapat dilihat pada Tabel 4.53 sebagai berikut:

**Tabel 4.53 Pengujian Data Benar ubah data kategori**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data kategori dengan benar. Contoh:<br>Nama kategori: | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| Denim | | | |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah data kategori maka dapat dilihat pada Tabel 4.54 sebagai berikut:

**Tabel 4.54 Pengujian Data Salah ubah data kategori**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data kategori | Tampilkan pesan kesalahan "data kategori gagal diubah, masukan data kategori dengan lengkap" | Dapat menampilkan pesan kesalahan "data kategori gagal di ubah, masukan data kategori dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.14. Pengujian tambah data kota

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah data kota maka dapat dilihat pada Tabel 4.55 sebagai berikut:

**Tabel 4.55 Pengujian Data Benar tambah data kota**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data kota dengan benar<br>Contoh:<br>Nama kota: Bandung<br>provinsi: jawa Barat<br>ongkos kirim: 5000 | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah data kota maka dapat dilihat pada Tabel 4.56 sebagai berikut:

**Tabel 4.56 Pengujian Data Salah tambah data kota**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data kota | Tampilkan pesan kesalahan "data kota gagal ditambahkan, masukan data kota dengan lengkap" | Dapat menampilkan pesan kesalahan "data kota gagal ditambahkan, masukan data kota dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.15. Pengujian ubah data kota

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah data kota maka dapat dilihat pada Tabel 4.57 sebagai berikut:

**Tabel 4.57 Pengujian Data Benar ubah data kota**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data kota dengan benar<br>Contoh:<br>Nama kota: Bandung<br>provinsi: Jawa Barat<br>Ongkos Kirim: 5000 | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah data kota maka dapat dilihat pada Tabel 4.58 sebagai berikut:

**Tabel 4.58 Pengujian Data Salah ubah data kota**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data kota | Tampilkan pesan kesalahan "data kota gagal diubah, masukan data kota dengan | Dapat menampilkan pesan "data kota gagal di ubah, masukan data kota | [√ ] Diterima<br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| | lengkap” | dengan lengkap” | |

### 4.5.2.16. Pengujian tambah data provinsi

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah data provinsi maka dapat dilihat pada Tabel 4.59 sebagai berikut:

**Tabel 4.59 Pengujian Data Benar tambah data provinsi**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data provinsi dengan benar, Contoh: Nama provinsi: jawa Barat | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah data provinsi maka dapat dilihat pada Tabel 4.60 sebagai berikut:

**Tabel 4.60 Pengujian Data Salah tambah data provinsi**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data provinsi | Tampilkan pesan kesalahan “data provinsi gagal ditambahkan, masukan data provinsi dengan lengkap” | Dapat menampilkan pesan kesalahan “data provinsi gagal ditambahkan, masukan data provinsi dengan lengkap” | [√ ] Diterima<br>[ ] Ditolak |

### 4.5.2.17. Pengujian ubah data provinsi

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah data provinsi maka dapat dilihat pada Tabel 4.61 sebagai berikut:

**Tabel 4.61 Pengujian Data Benar ubah data provinsi**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data provinsi dengan benar<br>Contoh:<br>Nama provinsi:<br>Jawa Barat | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah data provinsi maka dapat dilihat pada Tabel 4.62 sebagai berikut:

**Tabel 4.62 Pengujian Data Salah ubah data provinsi**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data provinsi | Tampilkan pesan kesalahan "data provinsi gagal diubah, masukan data provinsi dengan lengkap" | Dapat menampilkan pesan kesalahan "data provinsi gagal di ubah, masukan data provinsi dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.18. Pengujian tambah data petugas

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah data petugas maka dapat dilihat pada Tabel 4.63 sebagai berikut:

**Tabel 4.63 Pengujian Data Benar tambah data petugas**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data petugas dengan benar. Contoh:<br>Nama petugas:<br>Andreas saputra<br>Email petugas:<br>slow.abiez@gmail. | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| com | | | |

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah data petugas maka dapat dilihat pada Tabel 4.64 sebagai berikut:

**Tabel 4.64 Pengujian Data Salah tambah data petugas**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data petugas | Tampilkan pesan kesalahan "data petugas gagal ditambahkan, masukan data petugas dengan lengkap" | Dapat menampilkan pesan kesalahan "data petugas gagal ditambahkan, masukan data petugas dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.19. Pengujian ubah data petugas

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah data petugas maka dapat dilihat pada Tabel 4.65 sebagai berikut:

**Tabel 4.65 Pengujian Data Benar ubah data provinsi**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data petugas dengan benar<br>Contoh:<br>Nama petugas: Andreas saputra<br>Email petugas: slow.abiez@gmail.com<br>Level: admin<br>Aktif: ya | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah data petugas maka dapat dilihat pada Tabel 4.66 sebagai berikut:

**Tabel 4.66 Pengujian Data Salah ubah data petugas**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Mengosongkan masukan data petugas | Tampilkan pesan kesalahan "data petugas gagal diubah, masukan data petugas dengan lengkap" | Dapat menampilkan pesan kesalahan "data petugas gagal di ubah, masukan data petugas dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

## 4.5.2.20. Pengujian ubah password operator/admin

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah password operator/admin maka dapat dilihat pada Tabel 4.67 sebagai berikut:

**Tabel 4.67 Pengujian Data Benar ubah password operator/admin**

| Kasus dan Hasil Uji (Data Benar) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data ubah password dengan benar<br>Contoh:<br>Password lama: 123456<br>Password baru: 654321<br>Konfirmasi password baru: 654321 | Mengisi *field* data yang dimasukan dengan benar | Dapat mengisi *field* data yang dimasukan dengan benar | [√ ] Diterima<br>[ ] Ditolak |

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah data provinsi maka dapat dilihat pada Tabel 4.68 sebagai berikut:

**Tabel 4.68 Pengujian Data Salah ubah data provinsi**

| Kasus dan Hasil Uji (Data Salah) | | | |
|---|---|---|---|
| Data Masukan | Yang diharapkan | pengamatan | kesimpulan |
| Memasukan data password lama salah | Tampilkan pesan kesalahan "password salah" | Dapat menampilkan pesan kesalahan "password salah" | [√ ] Diterima<br>[ ] Ditolak |
| Memasukan data password baru dan konfirmasi password baru tidak sesuai | Tampilkan pesan kesalahan "password tidak sesuai" | Dapat menampilkan pesan kesalahan "password tidak sesuai" | [√ ] Diterima<br>[ ] Ditolak |
| Mengosongkan masukan data ubah password | Tampilkan pesan kesalahan "password gagal diubah, masukan data dengan lengkap" | Dapat menampilkan pesan kesalahan "password gagal diubah, masukan data dengan lengkap" | [√ ] Diterima<br>[ ] Ditolak |

### 4.5.3. Kesimpulan Pengujian Alpa

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

## 4. 6. Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan mengenai kepuasan user dengan kandungan poin yaitu pemenuhan kebutuhan

dari tujuan awal pembangunan *website e-commerce* dan tampilan antarmuka dari *website e-commerce* tersebut. Pengujian beta dilakukan melalui dua teknik pengambilan data, yaitu melalui wawancara dan kuesioner.

Pengujian beta melalui wawancara dilakukan untuk halaman administrator dan halaman operator pada *website e-commerce* showroom cardinal karena terbatasnya jumlah pengguna untuk administrator dan operator. Wawancara mengenai halaman administrator dilakukan terhadap kepala showroom cardinal, sedangkan untuk wawancara mengenai halaman operator dilakukan terhadap pegawai showroom cardinal.

Pengujian beta melalui kuesioner dilakukan dengan membuat kuesioner mengenai kepuasan user dengan kandungan poin syarat user friendly untuk selanjutnya dibagikan kepada sebagian user dengan mengambil sample sebanyak 10 orang. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru.

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

1. Lingkungan penjualan di showroom cardinal

Penelitian di lingkungan penjual di showroom cardinal dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

2. Lingkungan masyarakat umum

Penelitian di lingkungan pada masyarakat umum khususnya yang sudah terbiasa dengan teknologi internet, dilakukan supaya dapat mengetahui sejauh

mana sistem yang dibangun dapat memberikan kemudahan masyarakat umum untuk melakukan transaksi pemesanan dan pembelian barang secara online.

### 4.6.1. Wawancara Pengguna

Wawancara dilakukan dengan menggunakan teknik kualitatif melalui pertanyaan terbuka. Wawancara mengenai halaman administrator dilakukan kepada 1 orang yaitu kepala showroom cardinal, sedangkan untuk wawancara mengenai halaman operator dilakukan terhadap 1 orang yaitu pegawai showroom cardinal. Dari hasil wawancara tersebut akan dilakukan pengambilan kesimpulan terhadap penilaian penerapan sistem yang baru.

Wawancara terdiri dari 5 pertanyaan terbuka yang akan ditanyakan kepada narasumber.

#### 4.6.1.1. Wawancara Pengujian Beta Untuk Administrator

Wawancara dilakukan untuk pengujian beta terhadap *website e-commerce* showroom cardinal kepada pemilik showroom dengan total narasumber sebanyak satu orang. Berikut daftar pertanyaan yang diajukan kepada narasumber:

1. Secara *interface* bagaimana pendapat Anda tentang *website e-commerce showroom* Cardinal?
2. Menurut Anda, apakah fungsionalitas dari situs administrator pada *website e-commerce showroom* Cardinal sudah memenuhi kebutuhan dalam pengolahan datanya?
3. Menurut pendapat Anda, apakah cara penjualan produk yang diterapkan di *website e-commerce* sesuai dengan aturan bisnis yang ada pada *showroom* Cardinal?
4. Apakah menurut anda situs administrator pada *website e-commerce showroom* Cardinal mudah digunakan dan dipahami?

5. Apakah menurut anda cara penjualan produk di *website e-commerce showroom* Cardinal dapat meningkatkan jumlah penjualan?

Berikut ini adalah masing-masing jawaban dari pertanyaan hasil wawancara tersebut .

1. Secara *interface* bagaimana pendapat Anda tentang *website e-commerce showroom* Cardinal?

Jawaban :

| |
|---|
| Cukup menarik, *simple* dan tidak membingungkan |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa *inter*face dari *website e-commerce showroom* Cardinal .

2. Menurut Anda, apakah fungsionalitas dari situs administrator pada *website e-commerce showroom* Cardinal sudah memenuhi kebutuhan dalam pengolahan datanya?

Jawaban :

| |
|---|
| Ya sudah memenuhi, fungsionalitas berfungsi sebagai mana mestinya |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa fungsionalitas dari situs administrator pada *website e-commerce showroom* Cardinal sudah memenuhi kebutuhan dalam hal pengolahan datanya.

3. Menurut pendapat Anda, apakah cara penjualan produk yang diterapkan di *website e-commerce* sesuai dengan aturan bisnis yang ada pada *showroom* Cardinal?

Jawaban :

| |
|---|
| Ya sudah cukup baik, namun perlu dapat diatasi apabila ada aturan bisnis yang baru |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa cara penjualan produk yang diterapkan sudah sesuai dengan aturan bisnis yang ada pada *showroom* Cardinal.

4. Apakah menurut anda situs administrator pada *website e-commerce showroom* Cardinal mudah digunakan dan dipahami?

Jawaban :

| |
|---|
| Sangat mudah dipahami dan digunakan karena aplikasi yang dibangun sederhana namun berfungsi sebagai mana mestinya |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa halaman administrator *website e-commerce showroom* Cardinal cukup mudah digunakan dan dipahami.

5. Apakah menurut anda halaman administrator *website e-commerce showroom* Cardinal mudah dipahami dan digunakan?

Jawaban :

| |
|---|
| Secara keseluruhan halaman administrator ini mudah dipahami dan digunakan sehingga mempermudah saya dalam melakukan proses pengelolaan data. |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa dengan adanya *website e-commerce showroom* Cardinal, mudah dipahami dan digunakan.

### 4.6.1.2. Wawancara Pengujian Beta Untuk Operator

Wawancara dilakukan untuk pengujian beta terhadap *website e-commerce* showroom Cardinal kepada pegawai showroom dengan total narasumber sebanyak 1 orang. Berikut daftar pertanyaan yang diajukan kepada narasumber :

1. Secara *interface* bagaimana pendapat Anda tentang *website e-commerce showroom* Cardinal?
2. Menurut Anda, apakah fungsionalitas dari situs operator pada *website e-commerce showroom* Cardinal sudah memenuhi kebutuhan dalam pengolahan datanya?
3. Menurut pendapat Anda, apakah situs operator dalam *website e-commerce showroom* Cardinal dapat membantu Anda dalam mengelola data produk?
4. Menurut pendapat Anda, apakah halaman aperator dalam *website* ini dapat membantu Anda dalam mengelola data penjualan secara *online*?
5. Menurut pendapat Anda, apakah fungsionalitas dari halaman operator ini mudah digunakan dan dipahami?

Berikut ini adalah masing-masing jawaban dari pertanyaan hasil wawancara tersebut . Wawancara ini dilakukan kepada 1 orang.

1. Secara *interface* bagaimana pendapat Anda tentang *website e-commerce showroom* Cardinal?

Jawaban :

| |
|---|
| Bagus, tidak membuat bingung |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa halaman operator pada *website e-commerce* showroom Cardinal mudah digunakan.

2. Menurut Anda, apakah fungsionalitas dari situs operator pada *website e-commerce showroom* Cardinal sudah memenuhi kebutuhan dalam pengolahan datanya?

Jawaban :

| |
|---|
| Ya, sudah cukup |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa halaman operator dalam *website* ini dapat membantu operator dalam mengelola data barang.

3. Menurut pendapat Anda, apakah situs operator dalam *website e-commerce showroom* Cardinal dapat membantu Anda dalam mengelola data produk?

Jawaban :

| |
|---|
| Ya mudah sekali mengelola data produk |

Berdasarkan hasil prosentase diatas maka dapat disimpulkan bahwa halaman operator dalam *website* ini dapat membantu operator dalam mengelola data produk.

4. Menurut pendapat Anda, apakah situs aperator dalam *website* ini dapat membantu Anda dalam mengelola data penjualan secara *online*?

Jawaban :

| |
|---|
| Ya sangat membantu, apalagi dengan basic saya yang tidak memahami teknologi. Aplikasi ini mudah mengelola data penjualan |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa halaman aperator dalam *website* ini dapat membantu operator dalam pembuatan laporan penjualan.

5. Menurut pendapat Anda, apakah fungsionalitas dari halaman operator ini mudah digunakan dan dipahami?

Jawaban :

| |
|---|
| Sangat mudah digunakan dan dipahami karna ditampilkan dengan tampilan yang sederhana namun menarik |

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa fungsionalitas dari halaman operator ini mudah dipahami dan digunakan / *User friendly*.

### 4.6.2. Kuesioner Pengguna

Kuesioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 10 pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru.

#### 4.6.2.1. Kuesioner Pengujian Beta Untuk Pengguna

Kuisioner ini terdiri dari 5 dengan menggunakan skala likert dengan skala 1 sampai 5, dengan ketentuan skala untuk tiap pertanyaan sebagai berikut:

1. Menurut anda, apakah *website e-commerce showroom* Cardinal layak dalam hal pelayanan transaksi jual-beli secara *online*?

   A. Sangat layak
   B. layak
   C. Cukup layak
   D. Kurang layak
   E. Tidak layak

2. Apakah anda setuju dengan adanya *website e-commerce showroom* Cardinal showroom dapat memperluas cakupan pemasarannya?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

3. Apakah anda setuju dengan adanya *website e-commerce showroom* Cardinal memudahkan anda untuk melakukan pembelian barang tanpa harus datang langsung ke showroom Cardinal?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

4. Apakah anda setuju bahwa *website e-commerce showroom* Cardinal dapat membantu anda untuk mendapatkan informasi tentang produk terbaru yang ada pada showroom Cardinal?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

5. Apakah anda setuju bahwa tampilan *website e-commerce* showroom Cardinal cukup menarik dan mudah dipahami / *User friendly*?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

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

Keterangan :

P = Banyaknya jawaban responden tiap soal.

Q = Jumlah responden

Y = Nilai persentase

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

1. Menurut anda, apakah *website e-commerce showroom* Cardinal layak dalam hal pelayanan transaksi jual-beli secara *online*?

**Tabel 4.69 Hasil Pengujian Kuisioner Soal Nomor 1**

| **Pertanyaan** | **No.** | **Keterangan** | **Responden** | **Prosentase (%)** |
|---|---|---|---|---|
| 1 | 1 | Sangat layak | 3 | 30 |
| | 2 | layak | 6 | 60 |
| | 3 | Cukup layak | 1 | 10 |
| | 4 | Kurang layak | 0 | 0 |
| | 5 | Tidak layak | 0 | 0 |
| **Jumlah** | | | **10** | **100** |

Berdasarkan hasil prosentase pada Tabel 4.65 maka dapat disimpulkan sebanyak 3 atau 30% menyatakan sangat layak, 6 atau 60% menyatakan layak dan 1 atau 10% menyatakan cukup layak bahwa *website e-commerce* showroom Cardinal ini sudah layak dalam hal pelayanan transaksi jual-beli secara *online*.

2. Apakah anda setuju dengan adanya *website e-commerce* ini pemilik dapat memperluas cakupan pemasarannya ?

**Tabel 4.70 Hasil Pengujian Kuisioner Soal Nomor 2**

| Pertanyaan | No. | Keterangan | Responden | Prosentase (%) |
|---|---|---|---|---|
| 2 | 1 | Sangat Setuju | 3 | 30 |
| | 2 | Setuju | 6 | 60 |
| | 3 | Cukup Setuju | 1 | 10 |
| | 4 | Kurang Setuju | 0 | 0 |
| | 5 | Tidak Setuju | 0 | 0 |
| **Jumlah** | | | **10** | **100** |

Berdasarkan hasil prosentase pada Tabel 4.66 maka dapat disimpulkan sebanyak 3 orang atau 30% menyatakan sangat setuju, 6 orang atau 60% menyatakan setuju dan 1 orang atau 10% menyatakan cukup setuju bahwa dengan adanya *website e-commerce* ini pemilik dapat memperluas cakupan pemasarannya.

3. Apakah anda setuju dengan adanya *website e-commerce* ini dapat memudahkan anda untuk melakukan pembelian barang tanpa harus datang langsung ke showroom Cardinal ?

**Tabel 4.71 Hasil Pengujian Kuisioner Soal Nomor 3**

| Pertanyaan | No. | Keterangan | Responden | Prosentase (%) |
|---|---|---|---|---|
| 3 | 1 | Sangat Setuju | 2 | 20 |
| | 2 | Setuju | 6 | 60 |
| | 3 | Cukup Setuju | 2 | 20 |
| | 4 | Kurang Setuju | 0 | 0 |
| | 5 | Tidak Setuju | 0 | 0 |
| **Jumlah** | | | **10** | **100** |

Berdasarkan hasil prosentase pada Tabel 4.65 maka dapat disimpulkan sebanyak 2 atau 20% menyatakan sangat setuju dan 6 orang atau 60% menyatakan setuju dan, 2 atau 20% menyatakan cukup setuju bahwa *website e-commerce* ini

dapat mempermudah member dalam melakukan pembelian barang tanpa harus datang langsung ke showroom Cardinal.

4. Apakah anda setuju bahwa *website e-commerce* ini dapat membantu anda untuk mendapatkan informasi tentang produk terbaru yang ada pada showroom Cardinal ?

**Tabel 4.72 Hasil Pengujian Kuisioner Soal Nomor 4**

| Pertanyaan | No. | Keterangan | Responden | Prosentase (%) |
|---|---|---|---|---|
| 4 | 1 | Sangat Setuju | 6 | 60 |
| | 2 | Setuju | 4 | 40 |
| | 3 | Cukup Setuju | 0 | 0 |
| | 4 | Kurang Setuju | 0 | 0 |
| | 5 | Tidak Setuju | 0 | 0 |
| **Jumlah** | | | **10** | **100** |

Berdasarkan hasil prosentase pada Tabel 4.68 maka dapat disimpulkan sebanyak 6 orang atau 60% menyatakan sangat setuju dan 4 orang atau 40% menyatakan setuju bahwa *website e-commerce* ini dapat membantu untuk mendapatkan informasi tentang produk terbaru yang ada pada showroom Cardinal.

5. Apakah anda setuju bahwa tampilan *website e-commerce* showroom Cardinal cukup menarik dan mudah dipahami / *User friendly* ?

**Tabel 4.73 Hasil Pengujian Kuisioner Soal Nomor 5**

| Pertanyaan | No. | Keterangan | Responden | Prosentase (%) |
|---|---|---|---|---|
| 5 | 1 | Sangat Setuju | 3 | 30 |
| | 2 | Setuju | 7 | 70 |
| | 3 | Cukup Setuju | 0 | 0 |
| | 4 | Kurang Setuju | 0 | 0 |
| | 5 | Tidak Setuju | 0 | 0 |
| **Jumlah** | | | **10** | **100** |

Berdasarkan hasil prosentase pada Tabel 4.69 maka dapat disimpulkan sebanyak 3 orang atau 30% menyatakan sangat setuju dan 7 orang atau 70% menyatakan bahwa tampilan *website e-commerce* showroom Cardinal cukup menarik dan mudah dipahami / *User friendly*.

### 4.6.3. Kesimpulan Pengujian Beta

Berdasarkan hasil wawancara dan kuesioner yang didapatkan dari pengujian beta administrator, operator dan pengguna, maka sistem penerapan *e-commerce* untuk sistem penjualan ini sudah sesuai dengan tujuan yaitu untuk memenuhi tuntutan ekonomi digital sehingga dapat meningkatkan pelayanan kepada konsumen, memperluas cakupan pemasaran *showroom* Cardinal, memudahkan konsumen untuk melakukan pembelian barang tanpa harus datang langsung ke *showroom* Cardinal dan mempermudah *showroom* Cardinal dalam mempromosikan produk-produk kepada konsumen tanpa harus membangun cabang-cabang baru yang memakan banyak biaya.