BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Tahap implementasi sistem merupakan tahap menterjemahkan perancangan, berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya. Untuk mendukung aplikasi yang akan diterapkan pada lingkungan implementasi, maka dalam hal ini menggunakan perangkat keras dan perangkat lunak yang menunjang dalam Membangun Aplikasi E-commerce pada M.VITO JAYA SHOES.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan oleh user baik itu dari sisi pengunjung, member operator, dan admin bersifat relatif, tergantung dari spesifikasi yang dimiliki dan akan digunakan, namun sebagai gambaran berikut sistem minimal yang dapat dipakai:

a. *Processor* dengan kecepatan minimal 2 GHz + *Motherboard*
b. Kapasitas *Harddisk* minimal 80 GB
c. RAM minimal 512 GB

d. VGA *Card* minimal 128 MB

e. *Monitor*

f. *Lan Card 10/100Mbps*

g. Koneksi *internet* minimal 64 kbps

h. *Mouse / keyboard*

4.1.2 Implementasi Perangkat Lunak

Spesifikasi perangkat lunak yang digunakan untuk membangun aplikasi website e-commerce adalah sebagai berikut:

1. XAMPP sebagai *Web Server*
2. PHP
3. Macromedia Dreamweaver 8 sebagai editor PHP.
4. MySQL sebagai DBMS
5. Mozila Firefox 3.6.13 sebagai *browser*

2.1 4.1.3 Implementasi Database

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

1. Tabel Admin

```
CREATE TABLE IF NOT EXISTS `admin` (
  `username` varchar(15) NOT NULL,
  `password` varchar(32) NOT NULL,
  `email` varchar(40) NOT NULL,
  `hak_akses` enum('admin','kasir','petugas') NOT NULL,
  PRIMARY KEY (`username`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Tabel Buku Tamu

```
CREATE TABLE IF NOT EXISTS `buku_tamu` (
  `id_tamu` int(5) NOT NULL AUTO_INCREMENT,
  `username` varchar(15) NULL,
  `nama` varchar(30) NOT NULL,
  `komentar` longtext NOT NULL,
  `tgl_input` datetime NOT NULL,
  `email` varchar(40) NOT NULL,
  PRIMARY KEY (`id_tamu`)
 CONSTRAINT  `buku_tamu_ibfk_1`  FOREIGN  KEY  (`username`)
 REFERENCES `admin` (`username`) ON UPDATE CASCADE;
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

3. Tabel Detail Pembelian

```
CREATE TABLE IF NOT EXISTS `detail_pembelian` (

  `id_detailbeli` int(11) NOT NULL AUTO_INCREMENT,

  `id_pembelian` int(11) NOT NULL,

  `id_detailp` int(11) NOT NULL,

  `username` varchar(15) NULL,

  `hargabeli` int(20) NOT NULL,

  `qty` int(11) NOT NULL,

  `diskon` int(11) NOT NULL,

  PRIMARY KEY (`id_detailbeli`)

CONSTRAINT      `detail_pembelian_ibfk_3`      FOREIGN      KEY
(`id_pembelian`)  REFERENCES  `pembelian`  (`id_pembelian`)  ON
UPDATE CASCADE,

CONSTRAINT      `detail_pembelian_ibfk_2`      FOREIGN      KEY
(`id_detailp`)  REFERENCES  `detail_produk`  (`id_detailp`)  ON
UPDATE CASCADE,

CONSTRAINT `detail_pembelian_ibfk_1` FOREIGN KEY (`username`)



REFERENCES `admin` (`username`) ON UPDATE CASCADE;

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

4. Tabel Detail Produk

```
CREATE TABLE IF NOT EXISTS `detail_produk` (

  `id_detailp` int(11) NOT NULL AUTO_INCREMENT,

  `id_produk` int(8) NOT NULL,

  `id_ukuran` int(11) NOT NULL,

  `stok` int(11) NOT NULL,

  `berat` int(11) NOT NULL,

  PRIMARY KEY (`id_detailp`)
```

```
CONSTRAINT `detail_produk_ibfk_2` FOREIGN KEY (`id_produk`)
REFERENCES `produk` (`id_produk`) ON UPDATE CASCADE,

CONSTRAINT `detail_produk_ibfk_1` FOREIGN KEY (`id_ukuran`)
REFERENCES `ukuran` (`id_ukuran`) ON UPDATE CASCADE;

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

5. Tabel Detail Retur

```
CREATE TABLE IF NOT EXISTS `detail_retur` (

  `id_detail_retur` int(11) NOT NULL AUTO_INCREMENT,

  `qty` int(11) NOT NULL,

  `ket` varchar(255) NOT NULL,

  `id_retur` int(11) NOT NULL,

  `id_detailbeli` int(11) NOT NULL,

  PRIMARY KEY (`id_detail_retur`)

CONSTRAINT `detail_retur_ibfk_2` FOREIGN KEY (`id_detailbeli`)
REFERENCES `detail_pembelian` (`id_detailbeli`) ON UPDATE
CASCADE,

CONSTRAINT `detail_retur_ibfk_1` FOREIGN KEY (`id_retur`)
REFERENCES `retur` (`id_retur`) ON UPDATE CASCADE,

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

6. Tabel Jasa Pengiriman

```
CREATE TABLE IF NOT EXISTS `jasapengiriman` (

  `id_jasapengiriman` int(11) NOT NULL AUTO_INCREMENT,

  `nama_jasapengiriman` varchar(20) NOT NULL,

  `deskripsi_jasapengiriman` text NOT NULL,
```

```
  PRIMARY KEY (`id_jasapengiriman`),
  UNIQUE KEY `nama_jasapengiriman` (`nama_jasapengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

7. Tabel Jenis Pengiriman

```
CREATE TABLE IF NOT EXISTS `jenispengiriman` (
  `id_jenispengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_jasapengiriman` int(11) NOT NULL,
  `nama_jenispengiriman` varchar(20) NOT NULL,
  `deskripsi_jenispengiriman` text NOT NULL,
  PRIMARY KEY (`id_jenispengiriman`)
CONSTRAINT      `jenispengiriman_ibfk_1`      FOREIGN      KEY
(`id_jasapengiriman`)       REFERENCES       `jasapengiriman`
(`id_jasapengiriman`) ON UPDATE CASCADE;
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

8. Tabel Kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL AUTO_INCREMENT,
  `kategori` varchar(15) NOT NULL,
  `h_dasar` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_kategori`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

9. Tabel Kurs

```
CREATE TABLE IF NOT EXISTS `kurs` (
  `id_kurs` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(15) NULL,
  `nilai` int(11) NOT NULL,
  `tanggal` date NOT NULL,
  PRIMARY KEY (`id_kurs`)
CONSTRAINT `kurs_ibfk_1` FOREIGN KEY (`username`) REFERENCES
`admin` (`username`) ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

10. Tabel Member

```
CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(11) NOT NULL AUTO_INCREMENT,
  `nama` varchar(50) NOT NULL,
  `alamat` text NOT NULL,
  `id_kabkota` int(10) NOT NULL,
  `kodepos` varchar(10) NOT NULL,
  `telepon` varchar(20) NOT NULL,
  `email` varchar(40) NOT NULL,
  `password` varchar(32) NOT NULL,
  `tgl_reg` datetime NOT NULL,
  `status` int(11) NOT NULL,
  PRIMARY KEY (`id_member`),
```

```
  UNIQUE KEY `email` (`email`)

CONSTRAINT   `member_ibfk_1`   FOREIGN   KEY   (`id_kabkota`)
REFERENCES `member` (`id_kabkota`) ON UPDATE CASCADE;

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

11. Tabel Ongkos Kirim

```
CREATE TABLE IF NOT EXISTS `ongkoskirim` (

  `id_ongkoskirim` int(11) NOT NULL AUTO_INCREMENT,

  `id_kabkota` int(11) NOT NULL,

  `id_jenispengiriman` int(11) NOT NULL,

  `harga_ongkoskirim` int(11) NOT NULL,

  PRIMARY KEY (`id_ongkoskirim`)

CONSTRAINT        `ongkoskirim_ibfk_2`        FOREIGN        KEY
(`id_jenispengiriman`)       REFERENCES       `jenispengiriman`
(`id_jenispengiriman`) ON UPDATE CASCADE,

CONSTRAINT  `ongkoskirim_ibfk_1`  FOREIGN  KEY  (`id_kabkota`)
REFERENCES `t_kabkota` (`id_kabkota`) ON UPDATE CASCADE;

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

12. Tabel Pembelian

```
CREATE TABLE IF NOT EXISTS `pembelian` (
  `id_pembelian` int(11) NOT NULL AUTO_INCREMENT,
  `session_id` varchar(32) NOT NULL,
  `tgl_beli` datetime NOT NULL,
  `tgl_bayar` datetime NOT NULL,
  `status` enum('pesan','bayar','konfirmasi','kirim','selesai')
NOT NULL,
  `id_member` int(11) NOT NULL,
  `pembayaran` enum('rekening','paypal') NOT NULL,
  `kirim_nama` varchar(30) NOT NULL,
  `kirim_alamat` text NOT NULL,
  `kirim_telp` varchar(12) NOT NULL,
  `id_ongkoskirim` int(11) NOT NULL,
  `kirim_kota` int(11) NOT NULL,
  `kirim_kdpos` varchar(6) NOT NULL,
  `kirim_ongkos` int(10) NOT NULL,
  `kirim_resi` varchar(15) NOT NULL,
  `transfer_jumlah` int(10) NOT NULL,
  `transfer_tgl` date NOT NULL,
  `penerima_bank` varchar(15) NOT NULL,
  `penerima_rekening` varchar(30) NOT NULL,
  `totalbayar` float(12,2) NOT NULL,
  `id_kurs` int(11) NOT NULL,
```

```
  PRIMARY KEY (`id_pembelian`)

CONSTRAINT `pembelian_ibfk_3` FOREIGN KEY (`id_ongkoskirim`)
REFERENCES `ongkoskirim` (`id_ongkoskirim`) ON UPDATE CASCADE,

CONSTRAINT `pembelian_ibfk_2` FOREIGN KEY (`id_kurs`)
REFERENCES `kurs` (`id_kurs`) ON UPDATE CASCADE,

CONSTRAINT `pembelian_ibfk_1` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`) ON UPDATE CASCADE;

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

13. Tabel Retur

```
CREATE TABLE IF NOT EXISTS `retur` (

  `id_retur` int(11) NOT NULL AUTO_INCREMENT,

  `id_pembelian` int(10) unsigned NOT NULL,

  `id_member` int(11) NOT NULL,

  `tgl_retur` date DEFAULT NULL,

  `status_retur` varchar(11) NOT NULL,

  `tgl_kirim` date DEFAULT NULL,

  `resi_kirim` varchar(20) DEFAULT NULL,

  `resi_retur` int(11) DEFAULT NULL,

  `norek` varchar(20) DEFAULT NULL,

  `bank` varchar(20) DEFAULT NULL,

  `namarek` varchar(50) DEFAULT NULL,

  PRIMARY KEY (`id_retur`)

  CONSTRAINT `retur_ibfk_2` FOREIGN KEY (`id_member`)
  REFERENCES `member` (`id_member`) ON UPDATE CASCADE,

  CONSTRAINT `retur_ibfk_1` FOREIGN KEY (`id_pembelian`)
  REFERENCES `pembelian` (`id_pembelian`) ON UPDATE CASCADE;

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

14. Tabel Kabkota

```
CREATE TABLE IF NOT EXISTS `t_kabkota` (
  `id_kabkota` int(10) NOT NULL AUTO_INCREMENT,
  `id_prov` int(10) DEFAULT NULL,
  `nama` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`id_kabkota`)
CONSTRAINT   `t_kabkota_ibfk_1`   FOREIGN   KEY   (`id_prov`)
REFERENCES `member` (`id_kabkota`) ON UPDATE CASCADE;
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

15. Tabel Provinsi

```
CREATE TABLE IF NOT EXISTS `t_provinsi` (
  `id_prov` int(10) NOT NULL AUTO_INCREMENT,
  `nama` varchar(30) DEFAULT NULL,
  PRIMARY KEY (`id_prov`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

16. Tabel ukuran

```
CREATE TABLE IF NOT EXISTS `ukuran` (
  `id_ukuran` int(11) NOT NULL AUTO_INCREMENT,
  `nama_ukuran` varchar(10) NOT NULL,
  PRIMARY KEY (`id_ukuran`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

17. Tabel produk

```
CREATE TABLE IF NOT EXISTS `produk` (
  `id_produk` int(11) NOT NULL AUTO_INCREMENT,
  `id_kategori` int(11) NOT NULL,
  `produk` varchar(30) NOT NULL,
  `image` varchar(100) DEFAULT NULL,
  `harga` int(20) NOT NULL,
  `diskon` int(3) unsigned NOT NULL DEFAULT '0',
  `tgl_input` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_produk`),
  UNIQUE KEY `produk` (`produk`),
  KEY `id_kategori` (`id_kategori`)
  CONSTRAINT `produk_ibfk_1` FOREIGN KEY (`id_kategori`)
  REFERENCES `kategori` (`id_kategori`) ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2.2 4.1.4 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna. Adapun untuk tampilan implementasi pembangunan sistem penjualan *online* di *website* M.VITO JAYA SHOES akan dilampirkan pada halaman lampiran

Tabel 4.1 Implementasi antar muka.

Menu	Deskripsi	Nama File
Login admin	Digunakan sebagai halaman login admin	login.php
Member	Digunakan untuk melihat, menambah, dan merubah data member	member.php
Kota	Digunakan untuk melihat, menambah, dan merubah data kota	kota. php
Kecamatan	Digunakan untuk melihat, menambah, dan merubah data kecamatan	Kecamatan. php
Provinsi	Digunakan untuk melihat, menambah, dan merubah data provinsi	provinsi. php
Produk	Digunakan untuk melihat, menambah, dan merubah data produk	produk. php
Ongkos kirim	Digunakan untuk melihat,	ongkoskirim. php

	menambah, dan merubah data ongkos kirim	
Kategori	Digunakan untuk melihat, menambah, dan merubah data kategori produk	kategori.php
Retur	Digunakan untuk melihat data retur dan mengubah status retur	admin_retur.php
Laporan produk	Digunakan untuk melihat laporan produk	laporan.php
Laporan penjualan	Digunakan untuk melihat laporan penjualan	laporan.php
Laporan retur barang	Digunakan untuk melihat laporan retur	laporan.php
Halaman Depan	Digunakan untuk menampilkan halaman utama	lib_func.php
Registrasi *member*	Digunakan untuk melakukan pendaftaran *member*	register.php
About	Digunakan untuk menampilkan halaman about	about.php
Hubungi kami	Digunakan untuk menampilkan halaman hubungi kami	kontak.php
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus detail transaksi yang akan dilakukan	pemesanan.php

4.2 Pengujian Perangkat Lunak

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

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

2.3 4.2.1 Pengujian Alpa

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

4.2.1.1 Skenario Pengujian Alpa

Skenario pengujian alpa menjelaskan skenario pengujian website sistem *ecommerce* adalah sebagai berikut:

Tabel 4.2 menjelaskan skenario pengujian aplikasi *member*.

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Registrasi *member*	Menampilkan *form* user baru	*Black Box*
	Isi *form* pendaftaran	*Black Box*

	Simpan data baru	*Black Box*
Login	Isi data login	*Black Box*
	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Logout	Menghapus *session user*	*Black Box*
Edit profil	Menampilkan *form* ubah profil	*Black Box*
	Ubah data *costumer*	*Black Box*
	Simpan perubahan	*Black Box*
Ganti *password*	Menampilkan *form* ubah *password*	*Black Box*
	Verifikasi *password* lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verifikasi konfirmasi *password*	*Black Box*
Menu produk	Menampilkan data barang	*Black Box*
	Menampilkan data detail barang	*Black Box*
	Cari barang	*Black Box*
Menu about	Menampilkan profil perusahaan	*Black Box*
Menu cara pembayaran	Menampilkan petunjuk pembayaran	*Black Box*
Menu keranjang belanja	Menampilkan data detail transaksi	*Black Box*

Tabel 4.3 menjelaskan skenario pengujian aplikasi admin.

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Login	Isi data login	
	Verifikasi *username*	*Black Box*
	Verivikasi *password*	*Black Box*
Logout	Menghapus *session user*	*Black Box*
Ganti P*assword*	Menampilkan *form* ganti *password*admin	*Black Box*
	Verifikasi *password*lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verfikasi konfirmasi *password*	*Black Box*
Member	Tampilkan data member	*Black Box*
Kota	Tampilkan data kota	*Black Box*
	Tambah data kota	*Black Box*
	Ubah data kota	*Black Box*
Provinsi	Tampilkan data Provinsi	*Black Box*
	Tambah data Provinsi	*Black Box*
	Ubah data Provinsi	*Black Box*
Kategori	Tampilkan data Kategori	*Black Box*
	Tambah data Kategori	*Black Box*
	Ubah data Kategori	*Black Box*
Retur	Tampilkan data Retur	*Black Box*
	Tambah data Retur	*Black Box*

	Ubah data Retur	*Black Box*
Laporan Penjualan	Tampilkan Laporan Penjualan	*Black Box*
Laporan Produk	Tampilkan Laporan Produk	*Black Box*

4.2.1.2 Butir Pengujian

Pengujian dilakukan dengan menguji setiap proses *data flow diagram* dan kemungkinan kesalahan yang terjadi untuk setiap proses. Pengujian ini dilakukan secara *black box*, yaitu pengujian dilakukan dengan hanya memperhatikan masukan ke sistem dan keluaran ke sistem.

1. Pengujian Registrasi

Tabel 4.4 Pengujian Registrasi *Member* (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data pendaftaran, contoh masukan: Nama: pandu yudha satria Alamat : Jl. Ciheulang Baru no.53 Telpon: 083829303029 Email : panduyudhas@gmail.com Password : 123456 Konfirmasi password : 123456 Provinsi : Jawa Barat Kota : Bandung Kodepos : 40134 KodeCaptcha: hay61	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses registrasi.	Form registrasi terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses registrasi.	[√] Diterima [ ] Ditolak

Tabel 4.5 Pengujian Registrasi *Member* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama: (kosong) Alamat : (kosong)	Tampilkan pesan kesalahan	Pesan kesalahan	[√] Diterima

Telpon: (kosong) Email : (kosong) Password : (kosong) Konfirmasi password : (kosong) Provinsi : (kosong) Kota : (kosong) Kodepos : (kosong) KodeCaptcha: (kosong)	dalam pengisian, maka akan muncul pesan kesalahan bahwa beberapa kolom yang tersedia harus terisi.	dapat tampil dan muncul pesan kesalahan bahwa beberapa kolom yang tersedia harus terisi.	[ ] Ditolak

2. Pengujian Login *Member*

Berikut dapat dilihat pengujian login *member* pada tabel 4.6 dibawah ini:

Tabel 4.6 Pengujian Login *Member* (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data login, contoh masukan: *email* : panduyudhas@gmail.com password: 123456	Mengisikan data alamat *email* pada *textbox* email dan password	Dapat mengisi *textbox* email dan password	[√] Diterima [ ] Ditolak

Tabel 4.7 Pengujian Login *Member* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data login, contoh masukan: *email* : (kosong) password: (kosong)	Tampilkan pesan kesalahan ”Maaf, Akun Tidak Ditemukan Masukan Email dan password dengan benar”	Dapat menampilkan pesan kesalahan ”Maaf, Akun Tidak Ditemukan Masukan Email dan password dengan benar”	[√] Diterima [ ] Ditolak

3. Pengujian Edit Profile *Member*

Berikut dapat dilihat pengujian dari *edit profile* member pada tabel 4.8 dibawah ini:

Tabel 4.8 Pengujian *Edit Profil Member* (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data *edit profile*, contoh masukan: Nama Lengkap: pandu Email: panduyuhdhas@gmail.com Password Lama: 123456 Password Baru: qwerty Konfirmasi Password: qwerty Alamat: Sekeloa Provinsi: Jawa Barat Kota: Bandung Kode Pos: 45122 Telepon: 081234567	Dapat mengubah isi *text box edit profil* yang akan diubah dan dapat melanjutkan proses *edit profile*	Form *edit profile* terisi atau benar dalam pengisian, maka akan dapat melanjutkan *edit profile*.	[√] Diterima [ ] Ditolak

Tabel 4.9 Pengujian *Edit Profil Member* (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data *edit profile*, contoh masukan: Nama Lengkap: (kosong) Email: (kosong) Password Lama: (kosong) Password Baru: (kosong) Konfirmasi Password: (kosong) Alamat: (kosong) Provinsi: (kosong) Kota: (kosong)	Muncul pesan kesalahan "kolom yang tersedia tidak boleh kosong"	Dapat menampilkan pesan kesalahan kolom yang tersedia tidak boleh kosong	[√] Diterima [ ] Ditolak

Kode Pos: (kosong) Telepon: (kosong)			

4. Pengujian Menu Kontak

Berikut dapat dilihat pengujian dari menu kontak pada tabel 4.10 dibawah ini:

Tabel 4.10 Pengujian Menu Kontak (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data kontak kami, contoh masukan: Nama: pandu Email: panduyudhas@gmail.com Komentar: Terima kasih Kode Captcha: 1Ofd0	Mengisikan data kontak kami pada *textbox*	Dapat mengisi *textbox*	[√] Diterima [ ] Ditolak

Tabel 4.11 Pengujian Menu Kontak (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data kontak kami, contoh masukan: Nama: (kosong) Email: (kosong) Komentar: (kosong) Kode Captcha: (kosong)	Muncul pesan kesalahan "kolom yang tersedia tidak boleh kosong"	Dapat menampilkan pesan kesalahan kolom yang tersedia tidak boleh kosong	[√] Diterima [ ] Ditolak

5. Pengujian Login Admin

Berikut dapat dilihat pengujian login admin pada tabel 4.12 dibawah ini:

Tabel 4.12 Pengujian Login Admin (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data login admin, contoh masukan: *username* : admin password: qwerty	Mengisikan data *username* pada *textbox username* dan password	Dapat mengisi *textbox username* dan password	[√] Diterima [ ] Ditolak

Tabel 4.13 Pengujian Login Admin (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data login admin, contoh masukan: *username* : (kosong) password: (kosong)	**Tampilkan pesan kesalahan "Username atau Password Salah"**	**Dapat menampilkan pesan kesalahan " Username atau Password Salah"**	[√] Diterima [ ] Ditolak

6. Pengujian Edit Admin

Berikut dapat dilihat pengujian dari *edit* admin pada tabel 4.14 dibawah ini:

Tabel 4.14 Pengujian *Edit* Admin (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data *edit* admin, contoh masukan: *username* : admin	Dapat mengubah isi *text box edit*	Form *edit* admin terisi atau benar dalam	[√] Diterima [ ] Ditolak

Email: ndoe_djay@yahoo.co.id Password Lama: qwerty Password: asdfgh Confirm Password: asdfgh	*admin* yang akan diubah dan dapat melanjutkan proses *edit* admin	pengisian, maka akan dapat melanjutkan *edit* admin.	

Tabel 4.15 Pengujian *Edit* Admin (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data *edit* admin, contoh masukan: *Username* : (kosong) Email: (kosong) Password Lama: (kosong) Password: (kosong) Confirm Password: (kosong)	Muncul pesan kesalahan "kolom yang tersedia tidak boleh kosong"	Dapat menampilkan pesan kesalahan kolom yang tersedia tidak boleh kosong	[√] Diterima [ ] Ditolak

7. Pengujian Data Tambah kategori

Berikut dapat dilihat pengujian data tambah kategori pada tabel 4.16 dibawah ini:

Tabel 4.16 Pengujian Data Tambah Kategori (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah Kategori, contoh masukan : Kategori : Caterpillar	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah kategori	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses tambah Kategori	[√] Diterima [ ] Ditolak

Tabel 4.17 Pengujian Data Tambah Kategori(Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah Kategori, contoh masukan: Kategori: (kosong)	Tampilkan pesan kesalahan "Kategori tidak boleh kosong"	Dapat menampilkan pesan kesalahan "Kategori tidak boleh kosong"	[√] Diterima [ ] Ditolak

8. Pengujian Data Edit Kategori

Berikut dapat dilihat pengujian data tambah kategori pada tabel 4.18 dibawah

Tabel 4.18 Pengujian Data Edit Kategori (Data Benar)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit Kategori, contoh masukan: Kategori:Marten	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses edit Kategori	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses edit Kategori	[√] Diterima [ ] Ditolak

Tabel 4.19 Pengujian Data Edit Kategori (Data Salah)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit Kategori, contoh masukan:	Tampilkan pesan kesalahan "Kategori tidak	Dapat menampilkan pesan	[√] Diterima [ ] Ditolak

Kategori: (kosong)	boleh kosong”	kesalahan “Kategori tidak boleh kosong”	

9. Pengujian Tambah Data Provinsi

Berikut dapat dilihat pengujian dari tambah data provinsi pada tabel 4.20 dibawah ini:

Tabel 4.20 Pengujian Tambah Provinsi(Data Benar)

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah provinsi, contoh masukan: Provinsi: Jawa Barat	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah provinsi	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses tambah provinsi	[√] Diterima [ ] Ditolak

Tabel 4.21 Pengujian Tambah Provinsi (Data Salah)

Kasus dan Hasil Uji Benar (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah provinsi, contoh masukan: Provinsi: (kosong)	Tampilkan pesan kesalahan “Provinsi tidak boleh kosong”	Dapat menampilkan pesan kesalahan “Provinsi tidak boleh kosong”	[√] Diterima [ ] Ditolak

10. Pengujian Edit Data Provinsi

Berikut dapat dilihat pengujian dari edit data provinsi pada tabel 4.22 dibawah ini:

Tabel 4.22 Pengujian Edit Data Provinsi(Data Benar)

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit provinsi, contoh masukan: Provinsi : Jawa Barat	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses edit provinsi	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses edit provinsi	[√] Diterima [ ] Ditolak

Tabel 4.23 Pengujian Edit Data Provinsi(Data Salah)

Kasus dan Hasil Uji Benar (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit provinsi, contoh masukan: Provinsi : (kosong)	Tampilkan pesan kesalahan "Provinsi tidak boleh kosong"	Dapat menampilkan pesan kesalahan "Provinsi tidak boleh kosong"	[√] Diterima [ ] Ditolak

11. Pengujian Tambah Data Kota

Berikut dapat dilihat pengujian data kota pada tabel 4.24 dibawah ini:

Tabel 4.24 Pengujian Tambah Data Kota

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah kota, contoh masukan: Nama Kota: Cirebon Provinsi: Jawa Barat Kab/Kota: Kota	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah kota	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses tambah kota	[√] Diterima [ ] Ditolak

Tabel 4.25 Pengujian Tambah Data Kota

Kasus dan Hasil Uji Benar (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah kota, contoh masukan: Nama Kota: (kosong) Provinsi: (kosong) Kab/Kota: (kosong)	Tampilkan pesan kesalahan "Kota, Provinsi dan Kab/Kota tidak boleh kosong"	Dapat menampilkan pesan kesalahan " Kota, Provinsi dan Kab/Kota tidak boleh kosong"	[√] Diterima [ ] Ditolak

12. Pengujian Edit Data Kota

Berikut dapat dilihat pengujian dari edit data kota pada tabel 4.26 dibawah ini:

Tabel 4.26 Pengujian Edit Data Kota (Data Benar)

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit kota, contoh masukan: Nama Kota: Cirebon Provinsi: Jawa Barat Kab/Kota: Kota	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses edit kota	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses edit kota	[√] Diterima [ ] Ditolak

Tabel 4.27 Pengujian Edit Data Kota (Data Salah)

Kasus dan Hasil Uji Benar (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit kota, contoh masukan: Nama Kota: (kosong) Provinsi: (kosong) Kab/Kota: (kosong)	Tampilkan pesan kesalahan "Kota, Provinsi dan Kab/Kota tidak boleh kosong"	Dapat menampilkan pesan kesalahan " Kota, Provinsi dan Kab/Kota tidak boleh kosong"	[√] Diterima [ ] Ditolak

13. Pengujian Data Tambah Produk

Berikut dapat dilihat pengujian data kategori pada tabel 4.38 dibawah ini:

Tabel 4.28 Pengujian Data Tambah Produk (Data Benar)

Kasus dan Hasil Uji Benar (Data Benar)

Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data tambah produk, contoh masukan: Nama Produk: Vilong Harga Produk: 48000 Diskon: 10% Kategori: Marten Masukan Gambar: C../vilong.jpg	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah produk	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses tambah produk	[√] Diterima [ ] Ditolak

Tabel 4.29 Pengujian Data Tambah Produk (Data Salah)

Kasus dan Hasil Uji Benar (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah produk, contoh masukan: Nama Produk: (kosong) Harga Produk: (kosong) Diskon: (kosong) Kategori: (kosong) Masukan Gambar: (kosong)	Tampilkan pesan kesalahan "Data tidak boleh kosong"	Dapat menampilkan pesan kesalahan "Data tidak boleh kosong"	[√] Diterima [ ] Ditolak

14. Pengujian Edit Data Produk

Tabel 4.30 Pengujian Data Tambah Produk (Data Benar)

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit produk, contoh masukan: Nama Produk: vilong Harga Produk: 48000 Diskon: 10% Kategori: marten Masukan Gambar: C../vilong.jpg	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses edit produk	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses edit produk	[√] Diterima [ ] Ditolak

Tabel 4.31 Pengujian Data Tambah Produk (Data Salah)

Kasus dan Hasil Uji Benar (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit produk, contoh masukan: Nama Produk: (kosong) Harga Produk: (kosong) Diskon: (kosong) Kategori: (kosong) Masukan Gambar: (kosong)	Tampilkan pesan kesalahan "Data tidak boleh kosong"	Dapat menampilkan pesan kesalahan "Data tidak boleh kosong"	[√] Diterima [ ] Ditolak

15. Pengujian Tambah Data Detail Produk

Berikut dapat dilihat pengujian data retur pada tabel 4.32 dibawah ini:

Tabel 4.32 Pengujian Tambah Data Detail Produk (Data Benar)

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah detail produk, contoh masukan: Ukuran: 30 Berat: 320gr Stok: 10pcs	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah detail produk	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses tambah detail produk	[√] Diterima [ ] Ditolak

Tabel 4.33 Pengujian Tambah Data Detail Produk (Data Salah)

Kasus dan Hasil Uji Benar (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data tambah detail produk, contoh	Tampilkan pesan kesalahan "Data	Dapat menampilkan	[√] Diterima

masukan: Ukuran: (kosong) Berat: (kosong) Stok: (kosong)	tidak boleh kosong"	pesan kesalahan "Data tidak boleh kosong"	[ ] Ditolak

16. Pengujian Edit Data Detail Produk

Tabel 4.34 Pengujian Edit Data Detail Produk (Data Benar)

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit detail produk, contoh masukan: Ukuran: 30 Berat: 320gr Stok: 10pcs	Jika form terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses edit detail produk	Jika semua form terisi atau benar dalam pengisian, dapat melanjutkan proses edit detail produk	[√] Diterima [ ] Ditolak

Tabel 4.35 Pengujian Edit Data Detail Produk (Data Benar)

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data edit detail produk, contoh masukan: Ukuran: (kosong) Berat: (kosong) Stok: (kosong)	Tampilkan pesan kesalahan "Data tidak boleh kosong"	Dapat menampilkan pesan kesalahan "Data tidak boleh kosong"	[√] Diterima [ ] Ditolak

4.2.1.3 Kesimpulan Pengujian Alpa

Berdasarkan hasi pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan sistem penjualan *online* di *website* M.VITO JAYA SHOES bebas kesalahan penulisan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.3 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan dengan membuat kuesioner mengenai kepuasan *user* dengan kandungan poin syarat *user friendy* untuk selanjutnya dibagikan kepada sebagian *user*. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru.

4.3.1 Wawancara Pengujian Beta Untuk Admin

1. Apakah *website* ini dapat membantu dalam pengolahan data *admin*?
2. Apakah anda setuju dengan adanya *website* ini, toko M.VITO JAYA SHOES dapat memperluas pangsa pasar penjualannya ?
3. Apakah anda mengerti dengan instruksi yang terdapat pada *website* tersebut ?
4. Apakah anda setuju dengan tampilan *website* admin untuk pengolahan data *user* ?

4.3.2 Hasil Pengujian Beta Untuk Admin

1. Apakah *website* ini dapat membantu dalam pengolahan data *admin*?

Jawaban: Sangat membantu, karena memudahkan kami untuk mengelolanya.

2. Apakah anda setuju dengan adanya *website* ini, toko M.VITO JAYA SHOES dapat memperluas pangsa pasar penjualannya?

Jawaban: Setuju, karena dengan adanya website ini toko saya akan semakin berkembang.

3. Apakah anda mengerti dengan instruktur yang terdapat pada *website* tersebut?

Jawaban: Setelah dijelaskan oleh anda, saya dapat mengerti bagaimana cara mengelolanya, ternyata sangat mudah.

4. Apakah anda setuju degan tampilan *website admin* untuk pengolahan data *admin*?

Jawaban: Sangat setuju.

4.3.3 Wawancara Pengujian Beta Untuk Bagian Kasir

1. Apakah *website* ini dapat membantu anda dalam pengolahan data transaksi dan retur?
2. Apakah *website* ini dapat membantu dalam pengolahan data penjualan secara *online* ?
3. Apakah *website* ini dapat membantu dalam pembuatan laporan penjualan, laporan produk dan laporan retur?

4. Apakah anda mengerti dengan instruksi yang terdapat pada *website* tersebut?
5. Apakah anda setuju dengan tampilan *website* pada halaman bagian kasir?

4.3.4 Hasil Pengujian Beta Untuk Bagian Kasir

1. Apakah *website* ini dapat membantu anda dalam pengolahan data *transaksi* dan *retur*?

Jawaban: Sangat membantu saya, karena dengan begini saya lebih mudah untuk mengelola *transaksi*, dan saya juga mudah untuk mengelola *retur* jika suatu saat *system retur* sudah berjalan di toko ini.

2. Apakah *website* ini dapat membantu dalam pengolahan data penjualan secara *online* ?

Jawaban: Tentu.

3. Apakah *website* ini dapat membantu dalam pembuatan laporan penjualan, laporan produk dan laporan retur?

Jawaban: Tentu saja sangat membantu, dengan begini saya tidak usah capek-capek untuk menulis laporan di kertas lagi.

4. Apakah anda mengerti dengan instruksi yang terdapat pada *website* tersebut?

Jawaban: Sangat mengerti sekali.

5. Apakah anda setuju dengan tampilan *website* pada halaman bagian kasir?

Jawaban: Ya, setuju.

4.3.5 Wawancara Pengujian Beta Untuk Bagian Petugas

1. Apakah *website* ini dapat membantu anda untuk mengelola data master?
2. Apakah anda mengerti dengan instruksi yang terdapat pada *website* tersebut?
3. Apakah anda setuju dengan tampilan *website* pada halaman bagian petugas?

4.3.6 Hasil Pengujian Beta Untuk Bagian Petugas

1. Apakah *website* ini dapat membantu anda untuk mengelola data master?

Jawaban: Sangat membantu, dengan begini membantu kerja saya agar lebih mudah untuk mengelola produk.

2. Apakah anda mengerti dengan instruksi yang terdapat pada *website* tersebut ?

Jawaban: Ya saya mengerti setelah tadi dijelaskan oleh anda.

3. Apakah anda setuju dengan tampilan *website* pada halaman bagian petugas?

Jawaban: **S**aya setuju.

4.3.7 Kuesioner Pengujian Beta Untuk *Member*

1. Apakah anda setuju bahwa *website e-commerce* ini memberikan kemudahan dalam mendapatkan informasi produk?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

   E. Tidak Setuju

2. Apakah anda setuju bahwa *website* ini akan lebih memudahkan anda dalam melakukan transaksi pembelian produk pada M.VITO JAYA SHOES?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

   E. Tidak Setuju

3. Apakah anda setuju bahwa sistem pembayaran yang akan digunakan akan lebih memudahkan anda?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

   E. Tidak Setuju

4. Apakah anda setuju bahwa *website* ini dapat memperluas penjualan produk ke luar kota Bandung ?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

   E. Tidak Setuju

5. Apakah anda setuju bahwa tampilan *website* M.VITO JAYA SHOES cukup mudah dipahami / *user friendly*?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

E. Tidak Setuju

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

$$P = \frac{S}{skorideal} \times 100$$

Keterangan :

P = Nilai persentase yang dicari

S = Jumlah frekuensi dikalikan dengan nilai yang di tetapkan jawaban

Skor ideal= Nilai tertinggi dikalikan dengan jumlah sampel

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

1. Apakah anda setuju bahwa *website e-commerce* ini memberikan kemudahan dalam mendapatkan informasi produk?

Tabel 4.36 Hasil pengujian kuesioner soal nomor 1

Pertanyaan	Keterangan	Skor	Frekuensi	S
1	Sangat Setuju	5	7	35
	Setuju	4	13	52
	Ragu-ragu	3	-	-
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			**20**	**87**

$$\mathrm{P} = \frac{87}{100} \mathrm{x}\, 100\% = 87\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tampilan aplikasi *e-commerce* memberikan kemudahan dalam mendapatkan informasi produk adalah 87% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.

2. Apakah anda setuju bahwa *website* ini akan lebih memudahkan anda dalam melakukan transaksi pembelian produk pada M.VITO JAYA SHOES?

Tabel 4.37 Hasil pengujian kuesioner soal nomor 2

Pertanyaan	Keterangan	Skor	Frekuensi	S
2	Sangat Setuju	5	15	75
	Setuju	4	5	20
	Ragu-ragu	3	-	-
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			**20**	**95**

$$P = \frac{95}{100} \times 100\% = 95\%$$

Berdasarkan hasil persentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan aplikasi *e-commerce* memberikan kemudahan dalam

melakukan transaksi pembelian produk pada M.VITO JAYA SHOES adalah 95% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.

3. Apakah anda setuju bahwa sistem pembayaran yang akan digunakan akan lebih memudahkan anda?

Tabel 4.38 Hasil pengujian kuesioner soal nomor 3

Pertanyaan	Keterangan	Skor	Frekuensi	S
3	Sangat Setuju	5	10	50
	Setuju	4	10	40
	Ragu-ragu	3	-	-
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			**20**	**90**

$$P = \frac{90}{100} \times 100\% = 90\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan aplikasi *e-commerce* memberikan kemudahan dalam sistem pembayaran adalah 90% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.

4. Apakah anda setuju bahwa *website* ini dapat memperluas penjualan produk ke luar kota Bandung?

Tabel 4.39 Hasil pengujian kuesioner soal nomor 4

Pertanyaan	Keterangan	Skor	Frekuensi	S
5	Sangat Setuju	5	5	25
	Setuju	4	15	60
	Ragu-ragu	3	-	-

	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			**20**	**85**

$$P = \frac{85}{100} \times 100\% = 85\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan mudah atau tidaknya aplikasi *e-commerce* dapat memperluas penjualan produk ke luar kota Bandung adalah 85% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.

5. Apakah anda setuju bahwa tampilan *website* M.VITO JAYA SHOES cukup mudah dipahami / *user friendly*?

Tabel 4.40 Hasil pengujian kuesioner soal nomor 5

Pertanyaan	Keterangan	Skor	Frekuensi	S
4	Sangat Setuju	5	8	40
	Setuju	4	12	48
	Ragu-ragu	3	-	-
	Tidak Setuju	2	-	-
	Sangat Tidak Setuju	1	-	-
Jumlah			**20**	**88**

$$P = \frac{88}{100} \times 100\% = 88\%$$

Berdasarkan hasil presentase nilai di atas, maka dapat disimpulkan bahwa tingkat persetujuan aplikasi *e-commerce cukup mudah dipahami / user friendly* adalah 88% dari yang diharapkan 100%, atau dapat dikategorikan sebagai setuju.

4.3.8 Kesimpulan Pengujian Beta

Berdasarkan pengujian beta diatas, bahwa dapat diambil kesimpulan bahwa:

1. Aplikasi ini memudahkan pelanggan dalam mendapatkan informasi barang yang tersedia.

2. Pelanggan dapat melakukan transaksi pembelian barang di *website* M.VITO JAYA SHOES.
3. Proses pembayaran yang digunakan cukup mudah digunakan bagi para pelanggan.
4. Aplikasi ini juga memudahkan pihak toko dalam mendokumentasikan data master dan data transaksi penjualan barangnya.
5. Aplikasi *e-commerce* ini menarik.