BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penciptaan perangkat lunak, tahap kelanjutan dari kegiatan perancangan sistem. Tahap ini merupakan menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sesungguhnya.

4.1.1 Perangkat keras yang digunakan

Spesifikasi perangkat keras (*hardware*) yang digunakan dalam pembangunan *E-Commerce* di Batagoro store adalah tercantum pada tabel 4.1 sebagai berikut :

Tabel 4.1 Perangkat keras yang digunakan

No	Perangkat Keras	Spesifikasi
1	Processor	Dual Core T42000 @2.20GHZ
2	RAM	2.00 Gb
3	Harddisk	250 Gb
4	Mouse	-

4.1.2 Perangkat Lunak yang digunakan

Spesifikasi perangkat lunak (*Software*) yang dipasang pada sistem komputer yang digunakan untuk mengembangkan sistem penerapan *E-commerce* untuk sistem penjualan ini adalah tercantum pada tabel 4.2 sebagai berikut :

Tabel 4.2 Perangkat lunak yang digunakan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP profesional
2	PHP	Bahasa Pemrograman
3	*MySQL*	Penyimpanan *database*
4	*PhpMyUser*	*interface database*
5	*WampServer 2.0*	*web server*
6	Macromedia Dreamweaver 8	Kode dan perancangan tampilan

7	*Mozilla Firefox*	*Web browser*

4.2 Implementasi Basis Data

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

1. Tabel user

```
 CREATE TABLE IF NOT EXISTS `user` (
  `id_user` int(3) NOT NULL AUTO_INCREMENT,
  `user` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `nama_lengkap` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `no_telp` varchar(20) NOT NULL,
  `level` varchar(20) NOT NULL,
  `blokir` enum('Y','N') NOT NULL,
  `id_session` varchar(100) NOT NULL,
  PRIMARY KEY (`user`),
  UNIQUE KEY `id_ user ` (`id_ user `)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

2. Tabel detail retur

```
CREATE TABLE IF NOT EXISTS `detail_retur` (
  `id_detail_retur` int(5) NOT NULL AUTO_INCREMENT,
  `id_retur` int(5) NOT NULL,
  `id_produk` int(5) NOT NULL,
  `jumlah` int(10) NOT NULL,
  `alasan` text NOT NULL,
  `status_produk_retur` varchar(20) NOT NULL,
  PRIMARY KEY (`id_detail_retur`),
  KEY `id_produk` (`id_produk`),
  KEY `id_retur` (`id_retur`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

3. Tabel gambar

```
CREATE TABLE IF NOT EXISTS `gambar` (
  `id_gambar` int(5) NOT NULL AUTO_INCREMENT,
  `id_produk` int(5) NOT NULL,
  `nama_gambar` varchar(100) NOT NULL,
  `photo` varchar(100) NOT NULL,
  PRIMARY KEY (`id_gambar`),
  KEY `id_produk` (`id_produk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

4. Tabel jasa pengiriman

```
CREATE TABLE IF NOT EXISTS `jasa_pengiriman` (
  `jasa_kirim` varchar(25) NOT NULL,
  `perusahaan` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`jasa_kirim`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. Tabel Sub kategori

```
CREATE TABLE IF NOT EXISTS `sub kategori` (
  `id_sub kategori` int(5) NOT NULL AUTO_INCREMENT,
  `nama_sub kategori` varchar(100) NOT NULL,
  `sub kategori_seo` varchar(100) NOT NULL,
  PRIMARY KEY (`id_sub kategori`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

6. Tabel komentar

```
 CREATE TABLE IF NOT EXISTS `komentar` (
  `id_komentar` int(5) NOT NULL AUTO_INCREMENT,
  `id_produk` int(5) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `komentar` text NOT NULL,
  `tgl_komentar` date NOT NULL,
  `jam_komentar` time NOT NULL,
  `aktif` enum('Y','N') NOT NULL,
  PRIMARY KEY (`id_komentar`),
  KEY `id_produk` (`id_produk`)
```

```
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

7. Tabel Kota

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(10) NOT NULL AUTO_INCREMENT,
  `id_prov` int(10) DEFAULT NULL,
  `nama_kota` varchar(50) DEFAULT NULL,
  `kabkota` varchar(20) DEFAULT NULL,
  UNIQUE KEY `id_kota` (`id_kota`),
  KEY `id_prov` (`id_prov`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=428 ;
```

8. Tabel kurs

```
CREATE TABLE IF NOT EXISTS `kurs` (
  `kurs_id` int(11) NOT NULL AUTO_INCREMENT,
  `kurs_rupiah` int(7) NOT NULL DEFAULT '0',
  `kurs_user` int(6) NOT NULL DEFAULT '0',
  `kurs_update` varchar(25) NOT NULL,
  `kurs_flag` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP
ON UPDATE CURRENT_TIMESTAMP,
  PRIMARY KEY (`kurs_id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

9. Tabel member

```
CREATE TABLE IF NOT EXISTS `members` (
  `id_member` int(5) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `nama_lengkap` varchar(100) NOT NULL,
  `alamat` text NOT NULL,
  `jk` enum('P','W') NOT NULL,
  `email` varchar(30) NOT NULL,
  `no_telp` varchar(13) NOT NULL,
  `kota` int(10) NOT NULL,
```

```
  `provinsi` int(10) NOT NULL,
  `kode_pos` varchar(5) NOT NULL,
  `level` enum('member') NOT NULL,
  `blokir` enum('Y','N') NOT NULL,
  `id_session` varchar(100) NOT NULL,
  PRIMARY KEY (`username`),
  UNIQUE KEY `id_member` (`id_member`),
  KEY `kota` (`kota`),
  KEY `provinsi` (`provinsi`)\
  KEY ` user ` (`user `)\
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

10. Tabel Merk

```
CREATE TABLE IF NOT EXISTS `merk` (
  `id_merk` int(5) NOT NULL AUTO_INCREMENT,
  `nama_merk` varchar(50) NOT NULL,
  PRIMARY KEY (`id_merk`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

```
CREATE TABLE IF NOT EXISTS `ongkoskirim` (
  `id_ongkos` int(5) NOT NULL AUTO_INCREMENT,
  `id_kota` int(10) NOT NULL,
  `ongkos_kirim` int(10) NOT NULL,
  `perusahaan_jasa` varchar(40) NOT NULL DEFAULT 'Jalur
Nugraha Ekakurir',
  `jasa_kirim` varchar(25) NOT NULL,
  PRIMARY KEY (`id_ongkos`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1278 ;
```

```
CREATE TABLE IF NOT EXISTS `orders_detail` (
  `id_orders` int(11) NOT NULL,
  `id_produk` int(11) NOT NULL,
  `jumlah` int(5) NOT NULL,
  `total` int(10) NOT NULL,
  `diskon` int(3) NOT NULL,
  `harga` int(10) NOT NULL,
  `pkt_id` int(11) DEFAULT NULL,
  KEY `id_produk` (`id_produk`),
  KEY `id_orders` (`id_orders`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

```
CREATE TABLE IF NOT EXISTS `jenis_pengiriman` (
  `id_jenis_pengiriman` int(11) NOT NULL,
  `nama_jenis_pengiriman` varchar(30) NOT NULL,
  `id_jasa_pengiriman` int(11) NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

11. Order tempt

```
CREATE TABLE IF NOT EXISTS `orders_temp` (
  `id_orders_temp` int(5) NOT NULL AUTO_INCREMENT,
  `id_produk` int(5) NOT NULL,
  `username` varchar(50) NOT NULL,
  `jumlah` int(5) NOT NULL,
  `harga` int(10) NOT NULL,
  `diskon` int(3) NOT NULL,
  `tgl_order_temp` date NOT NULL,
  `jam_order_temp` time NOT NULL,
  `stok_temp` int(5) NOT NULL,
  `id_session` varchar(50) NOT NULL,
  `pkt_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`id_orders_temp`),
  KEY `id_produk` (`id_produk`),
  KEY `username` (`username`)
```

```
CREATE TABLE IF NOT EXISTS `produk` (
  `id_produk` int(5) NOT NULL AUTO_INCREMENT,
  `id_sub kategori` int(5) NOT NULL,
  `id_subsub kategori` int(5) NOT NULL,
  `nama_produk` varchar(100) NOT NULL,
  `id_merk` int(5) NOT NULL,
  `produk_seo` varchar(100) NOT NULL,
  `deskripsi` text NOT NULL,
  `harga` int(20) NOT NULL,
  `diskon` int(3) NOT NULL,
  `berat` decimal(5,2) NOT NULL,
  `tgl_masuk` date NOT NULL,
  `gambar` varchar(100) NOT NULL,
  `stok` int(5) NOT NULL DEFAULT '0',
  `dibeli` int(5) NOT NULL,
  `dilihat` int(5) NOT NULL,
  PRIMARY KEY (`id_produk`),
  KEY `id_sub kategori` (`id_sub kategori`),
  KEY `id_merk` (`id_merk`),
  KEY `id_subsub kategori` (`id_subsub kategori`)
  KEY ` user ` (`user`)\
```

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_prov` int(10) NOT NULL AUTO_INCREMENT,
  `nama_prov` varchar(100) DEFAULT NULL,
  KEY `id_prov` (`id_prov`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO INCREMENT=34 ;
```

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(5) NOT NULL AUTO_INCREMENT,
  `username` varchar(50) NOT NULL,
  `id_orders` int(5) NOT NULL,
  `tgl_retur` date NOT NULL,
  `status_retur` varchar(20) NOT NULL,
  `no_resi_retur` varchar(20) NOT NULL,
  `biaya` varchar(12) DEFAULT NULL,
  PRIMARY KEY (`id_retur`),
  KEY `username` (`username`),
  KEY `id_orders` (`id_orders`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ;
```

12. Tabel kategori

```
CREATE TABLE IF NOT EXISTS  kategori` (
  `id_ kategori` int(5) NOT NULL AUTO_INCREMENT,
  `id_ kategori` int(5) NOT NULL,
  `nama_ kategori` varchar(20) NOT NULL,
  ` kategori_seo` varchar(20) NOT NULL,
  PRIMARY KEY (`id_ kategori`),
  KEY `id_kategori` (`id_ kategori`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

```
CREATE TABLE IF NOT EXISTS `vote` (
  `id` int(5) NOT NULL,
  `username` varchar(50) NOT NULL,
  `date` datetime NOT NULL,
  `vote` int(11) NOT NULL
    ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13. Tabel *Order*

```
TABLE IF NOT EXISTS `orders` (
  `id_orders` int(5) NOT NULL AUTO_INCREMENT,
  `jasa_kirim` varchar(20) NOT NULL,
  `no_transaksi` varchar(20) NOT NULL,
  `username` varchar(50) NOT NULL,
  `nama_kustomer` varchar(50) NOT NULL,
  `alamat` text NOT NULL,
  `telpon` varchar(13) NOT NULL,
  `email` varchar(30) NOT NULL,
  `status_order` varchar(30) NOT NULL,
  `status_bayar` varchar(30) NOT NULL,
  `status_pengiriman` varchar(30) NOT NULL,
  `kode_pos` int(5) NOT NULL,
  `tgl_order` date NOT NULL,
  `tgl_bayar` date NOT NULL,
  `tgl_batal` date NOT NULL,
  `tgl_sampai` date NOT NULL,
  `tot_order` int(10) NOT NULL,
  `dibayar_dlr` double(10,2) NOT NULL,
  `dibayar_rp` int(10) NOT NULL,
  `jam_order` time NOT NULL,
  `id_prov` int(10) NOT NULL,
  `id_kota` int(10) NOT NULL,
  `no_resi` varchar(20) NOT NULL,
  `jenis_bayar` varchar(20) NOT NULL,
  `ongkos` double NOT NULL,
  `atas_nama` varchar(50) NOT NULL DEFAULT 'Pemesan',
  `kode_trx` varchar(20) DEFAULT NULL,
  `no_rek` int(20) NOT NULL,
  `track_table` text NOT NULL,
  `track_tujuan` varchar(256) DEFAULT NULL,
  `track_operator` varchar(256) DEFAULT NULL,
  `track_tgl_terima` varchar(75) DEFAULT NULL,
  `track_penerima` varchar(256) DEFAULT NULL,
  `track_status` varchar(25) DEFAULT NULL,
  `track_tgl_kirim` date NOT NULL,
  `type_trx` char(1) NOT NULL DEFAULT 'R',
  `paket_name` varchar(100) NOT NULL DEFAULT 'REGULER',
  PRIMARY KEY (`id_orders`),
  KEY `username` (`username`),
  KEY `id_prov` (`id_prov`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2
```

4.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna pada. Adapun untuk tampilan implementasi pembangunan *E-commerce* di Batagoro Store berbasis *Web* akan dilampirkan pada halaman lampiran.

Tabel 4.3 Implementasi antarmuka Pengunjung dan Member

Menu	Deskripsi	Nama File
Registrasi member	Digunakan untuk melakukan pendaftaran *member* baru.	Register.php
Login *member*	Digunakan Digunakan sebagai halaman login *member*	account.php
Lupa *password*	Digunakan jika *member* lupa dengan *password*	Lupapass. Php
produk	Digunakan untuk melihat data produk	produk.php
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus data order yang akan dilakukan.	keranjangbelanja.php
Ganti password	Digunakan jika *member* ingin merubah *password*	ubahpass.php
Sub kategori	Digunakan untuk melihat produk persub kategori	sub kategori. Php
Tentang kami	Digunakan untuk melihat, data perusahaan	Tentang kami.php
Hubungi	Digunakan untuk melakukan tanya jawab secara langsung	hubungi. Php
Member History	Digunakan untuk melihat *history* pembelian yang pernah dilakukan.	history.php
Logout	Digunakan *member* untuk keluar dari akunnya.	logout.php

Tabel 4.4 Implementasi Antarmuka User

Menu	Deskripsi	Nama File
Login user	Digunakan sebagai halaman login user	index.php
Lupa Password	Digunakan jika user lupa terhadap passwordnya	forgotpassword.php
Ubah Profil User	Digunakan jika user ingin merubah	ubahprofile.php

	profil	
Ubah Password user	Digunakan jika user ingin merubah *password*	ubahpassword.php
Kota	Digunakan untuk melihat, menambah, dan merubah data kota	Kota. Php
Provinsi	Digunakan untuk melihat, menambah, dan merubah data provinsi	provinsi. Php
sub kategori	Digunakan untuk melihat, menambah, dan merubah data sub kategori	sub kategori. Php
Produk	Digunakan untuk melihat, menambah, dan merubah data produk	produk. Php
Merk	Digunakan untuk melihat, menambah, dan merubah data merk	merk. Php
Data Order	Digunakan untuk melihat data order yang masuk dan meubah statusnya.	dataorder. Php
Ongkos Kirim	Digunakan untuk melihat data ongkos kirim, merubah, menghapus ongkos kirim	ongkoskirim. Php
Laporan	Digunakan untuk melihat data laporan harian,bulanan maupun tahunan kemudian dicetak ke pdf	laporan.php
Hubungi	Digunakan untuk melihat data hubungi yang akan dibaca.	hubungi.php
Logout	Digunakan user Untuk dapat keluar dari aplikasi	logout.php

4.4 Pengujian Perangkat Lunak

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

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

4.5 Pengujian *Alpha*

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

4.5.1 Skenario Pengujian Alpa

Tabel 4.5 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 *Email*	*Black Box*
	Verifikasi *password*	*Black Box*
Ubah profil	Menampilkan *form* ubah profil	*Black Box*
	Ubah data *member*	*Black Box*
	Simpan perubahan	*Black Box*
ubah *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 produk	*Black Box*
	Menampilkan data produk	*Black Box*
	Cari produk	*Black Box*
Menu Kontak	Menampilkan profil perusahaan	*Black Box*
Menu Hubungi	Menampilkan form	*Black Box*
	Isi form hubungi	*Black Box*
Menu keranjang belanja	Menampilkan data pembelian	*Black Box*
	Mengupdate data pembelian	*Black Box*
	Menghapus data pembelian	*Black Box*
Form konfirmasi	Menampilkan konfirmasi order	*Black Box*
Form pengiriman	Menampilkan form pengiriman	*Black Box*
	Pilih jenis pengiriman	*Black Box*
Logout	Menghapus *session user*	*Black Box*

Tabel 4.6 Menjelaskan Skenario pengujian aplikasi user.

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Isi data login	*Black Box*
	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Lupa *password*	Menampilkan *form* Lupa *password*	*Black Box*
Data Order	Tampilkan data order	*Black Box*
	Ubah status order	*Black Box*
Sub kategori	Tampilkan data sub kategori	*Black Box*
	Tambah Data katgori	*Black Box*
	Ubah data sub kategori	*Black Box*
Provinsi	Tampilkan data Provinsi	*Black Box*
	Tambah data Provinsi	*Black Box*
	Ubah data Provinsi	*Black Box*
Kota	Tampilkan data kota	*Black Box*
	Tambah data kota	*Black Box*
	Ubah data kota	*Black Box*
Merk	Tampilkan data Merk	*Black Box*
	Tambah data Merk	*Black Box*
	Ubah data merk	*Black Box*
Data Laporan	Tampilkan data laporan	*Black Box*
	Cari data laporan	*Black Box*
	Cetak data laporan	*Black Box*
Data Order	Tampilkan data Order	*Black Box*
	Ubah data order	*Black Box*
Member	Tampilan Data Member	*Black Box*
Data sub kategori	Tampilkan data sub kategori	*Black Box*
	Tambah data sub kategori	*Black Box*
	Ubah data sub kategori	*Black Box*
Data Produk	Tampilkan data produk	*Black Box*
	Tambah data produk	*Black Box*
	Ubah data produk	*Black Box*
Logout	Menghapus *session user*	*Black Box*

4.5.2 Kasus dan Hasil Pengujian

4.5.2.1 Pengujian Registrasi Member

Berikut dapat dilihat pengujian dari registrasi *member* yang terdapat pada tabel 4.7 dibawah ini:

Tabel 4.7. Pengujian Registrasi *member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik Menu Register	Tampil form registrasi	Dapat menampilkan form registrasi	[√] Diterima [ ] Ditolak
Masukan data pendaftaran, contoh masukan : Nama : rizal Alamat Email : rizalvela@yahoo.co.id Alamat : Jl. Sekeloa Tengah No.I56 Telepon : 085642552625 Provinsi : Jawa Barat Kota : Bandung Kodepos : 40514 Password : 12345678 Konfirmasi password : 12345678	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
Klik tombol Daftar	Jika data masukan benar muncul pesan “Terima Kasih Anda Telah Berhasil Registrasi. Silahkan cek email untuk melakukan Aktifasi”	data masukan benar muncul pesan “Terima Kasih Anda Telah Berhasil Registrasi. Silahkan cek email untuk melakukan Aktifasi”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data kosong	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa	[√] Diterima [ ] Ditolak

	bahwa field tersebut harus terisi.	field tersebut harus terisi.	
Email tidak diisi	Tampilkan pesan kesalahan “Email harus diisi”	Dapat menampilkan pesan kesalahan “Email harus diisi”	[√] Diterima [ ] Ditolak
Nama Lengkap tidak diisi	Tampilkan pesan kesalahan”Nama lengkap harus diisi”	Dapat menampilkan Pesan kesalahan “Nama lengkap harus diisi”	[√] Diterima [ ] Ditolak
Alamat tidak diisi	Tampilkan pesan kesalahan”Alamat harus diisi”	Dapat menampilkan pesan kesalahan ”Alamat harus diisi”	[√] Diterima [ ] Ditolak
Provinsi tidak diisi	Tampilkan pesan kesalahan ”Provinsi harus diisi”	Dapat menampilkan pesan kesalahan ”Provinsi harus diisi”	[√] Diterima [ ] Ditolak
kota tidak diisi	Tampilkan pesan kesalahan ”kota harus diisi”	Dapat menampilkan Pesan kesalahan ”kota harus diisi”	[√] Diterima [ ] Ditolak
kode pos tidak diisi	Tampilkan pesan kesalahan ”kode pos harus diisi”	Dapat menampilkan pesan kesalahan ”kode pos harus diisi”	[√] Diterima [ ] Ditolak
password tidak diisi	Tampilkan pesan kesalahan ”password harus diisi”	Dapat menampilkan pesan kesalahan ”password harus diisi”	[√] Diterima [ ] Ditolak
confirm password tidak diisi	Tampilkan pesan kesalahan ”confirm harus diisi”	Dapat menampilkan pesan kesalahan ”confirm harus diisi”	[√] Diterima [ ] Ditolak
Kode pos tidak diisi	Tampilkan pesan kesalahan ” kode pos harus diisi”	Dapat menampilkan pesan kesalahan ”kode pos harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.2 Pengujian *Login Member*

Berikut dapat dilihat pengujian *login member* pada tabel 4.8 dibawah ini:

Tabel 4.8 Pengujian login *member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat *email* : rizalvela@yahoo.co.id	Mengisikan data alamat *email* pada *field* alamat *email*.	Dapat mengisi *field* alamat *email* dan dapat mengisi data login dengan benar	[√] Diterima [ ] Ditolak
Memasukan *Password* : 12345678	Mengisikan data *password* pada *field password*.	Mengisikan data *password* pada *field password*.	[√] Diterima [ ] Ditolak
Klik tombol login	Data *email* dan data *password* benar maka akan masuk ke menu utama	Data *email* dan data *password* benar maka akan masuk ke menu utama	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data *email* atau *password* yang tidak terdaftar	Tampilkan pesan kesalahan “Maaf account tidak di temukan ”	Muncul pesan “Maaf account tidak di temukan ”	[√] Diterima [ ] Ditolak
mengosongkan data *email* atau *password*	Tampilkan pesan kesalahan “Maaf Email dan password harus diisi”	Dapat menampilkan pesan kesalahan ” Maaf Email dan password harus diisi”	[√] Diterima [ ] Ditolak
alamat Email tidak diisi	Tampilkan pesan kesalahan ”email harus diisi”	Dapat menampilkan pesan kesalahan ” email harus diisi”	[√] Diterima [ ] Ditolak
Password tidak diisi	Tampilkan pesan kesalahan ”password harus diisi”	Dapat menampilkan pesan kesalahan”password harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.3 Pengujian *Logout*

Berikut dapat dilihat pengujian dari *logout* pada tabel 4.9. dibawah ini:

Tabel 4.9 Pengujian *logout*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu logout	Dapat keluar dari menu member dan muncul pesan ”terimaksih sudah berkunjung”	Dapat keluar dari menu member dan akan muncul pesan”terimaksih sudah berkunjung”	[√] Diterima [ ] Ditolak

4.5.2.4 Pengujian Ubah Profile Member

Berikut dapat dilihat pengujian dari ubah profil member pada tabel 4.10. dibawah ini:

Tabel 4.10 Pengujian Ubah Profil member

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu ubah profil	Menampilkan *form* ubah profil	Dapat menampilkan *form* ubah profil	[√] Diterima [ ] Ditolak
Data menu profil yang akan diubah	Dapat mengisi *field* data yang akan diubah	Dapat mengisi *field* data yang akan diubah	[√] Diterima [ ] Ditolak
Klik tombol update	Data berhasil di update di*database*	Data berhasil di update di*database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan ada yang tidak diisi	Menampilkan pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan bahwa field tersebut harus terisi.	[√] Diterima [ ] Ditolak
Email tidak diisi	Muncul pesan kesalahan “email harus diisi”	Dapat menampilkan pesan kesalahan email harus diisi	[√] Diterima [ ] Ditolak
Nama tidak diisi	Muncul pesan kesalahan “Nama harus diisi”	Dapat menampilkan pesan kesalahan Nama harus diisi	[√] Diterima [ ] Ditolak

Alamat tidak diisi	Muncul pesan kesalahan "Alamat harus diisi"	Dapat menampilkan pesan kesalahan Alamat harus diisi	[√] Diterima [ ] Ditolak
Telp tidak diisi	Muncul pesan kesalahan "telepon harus diisi"	Dapat menampilkan pesan kesalahan telepon harus diisi	[√] Diterima [ ] Ditolak
Provinsi tidak diisi	Muncul pesan kesalahan "Provinsi harus diisi"	Dapat menampilkan pesan kesalahan Provinsi harus diisi	[√] Diterima [ ] Ditolak
Kota tidak diisi	Muncul pesan kesalahan "kota harus diisi"	Dapat menampilkan pesan kesalahan kota harus diisi	[√] Diterima [ ] Ditolak
Kode pos tidak diisi	Muncul pesan kesalahan "kota harus diisi"	Dapat menampilkan pesan kesalahan kota harus diisi	[√] Diterima [ ] Ditolak

4.5.2.5 Pengujian Ganti *Password* Member

Berikut dapat dilihat pengujian dari ganti *password member* pada tabel 4.11 dibawah ini:

Tabel 4.11 Pengujian ganti password member

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu ubah *password*	Menampilkan *form* ubah *password*	Dapat menampilkan *form* ubah *password*	[√] Diterima [ ] Ditolak
data yang dimasukkan semuanya benar	*Password* baru berhasil disimpan di *database*	Dapat menyimpan data *password* yang baru di *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu field belum terisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan "*field* tersebut harus diisi"	[√] Diterima [ ] Ditolak

Data *password* lama salah	muncul pesan "password salah"	Dapat menampilkan pesan kesalahan ”*Password* salah”	[√] Diterima [ ] Ditolak
Password lama tidak diisi	muncul pesan kesalahan "password lama harus diisi”	Dapat menampilkan pesan kesalahan "password lama harus diisi”	[√] Diterima [ ] Ditolak
Confirm *password* tidak diisi	muncul pesan kesalahan "confirm password harus diisi”	Dapat menampilkan pesan kesalahan "confirm password harus diisi”	
Code tidak diisi	Muncul pesan kesalahan”kode spam harus diisi”	Dapat menampilkan pesan kesalahan Kode spam harus diisi	[√] Diterima [ ] Ditolak
Data kosong	Muncul pesan kesalahan field harus diisi	Dapat menampilkan pesan kesalahan field harus diisi	[√] Diterima [ ] Ditolak

4.5.2.6 Pengujiaan Menu Kontak

Berikut dapat dilihat pengujian dari menu kontak pada tabel 4.12 dibawah ini:

Tabel 4.12 Pengujian Menu Kontak

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu Kontak	Menampilkan profil perusahaan	Dapat menampilkan profil perusahaan	[√] Diterima [ ] Ditolak

4.5.2.7 Pengujian Keranjang Belanja

Berikut dapat dilihat pengujian dari menu keranjang belanja pada tabel 4.13 dibawah ini:

Tabel 4.13 Pengujian Menu Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu keranjang belanja	Menampilkan *form* keranjang belanja	Dapat menampilkan *form* keranjang belanja	[√] Diterima [ ] Ditolak
Data Qty	Tercantum pada	Dapat mengisi data	[√] Diterima

	text box data yang akan diubah	jumlah dengan benar	[ ] Ditolak
Klik tombol *Update*	Dapat menampilkan perubahan data jumlah (qty)	Dapat menampilkan perubahan data jumlah (qty)	[√] Diterima [ ] Ditolak
Klik tombol hapus	Data yang dihapus terhapus dari keranjang belanja	Dapat menghapus data pada keranjang belanja	[√] Diterima [ ] Ditolak
Klik *checkout*	Menampilkan form pengiriman	Dapat menampilkan form pengiriman	[√] Diterima [ ] Ditolak

4.5.2.8 Pengujian Menu Produk

Berikut dapat dilihat pengujian dari menu produk pada tabel 4.14 dibawah ini:

Tabel 4.14 Pengujian Menu Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu produk	Menampilkan data produk yang ada di *database*	menampilkan data produk yang ada di *data base*	[√] Diterima [ ] Ditolak
Klik add to chart pada data produk	Menampilkan detail produk	Dapat menampilkan detail produk	[√] Diterima [ ] Ditolak
Klik add to chart	Menuju ke keranjang belanja	Dapat menampilkan keranjang belanja	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
merk tidak dipilih	Menampilkan pesan “merk belum dipilih”	Pesan “merk belum dipilih”	[√] Diterima [ ] Ditolak
Memasukan jumlah (qty) produk lebih dari stok	Menampilkan pesan “stok tidak mencukupi”	Dapat menampilkan pesan ““stok tidak mencukupi”	[√] Diterima [ ] Ditolak

4.5.2.9 Pengujian Menu Sub kategori

Berikut dapat dilihat pengujian dari menu sub kategori pada tabel 4.15 dibawah ini:

Tabel 4.15 Pengujian Menu Sub kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu sub kategori	Menampilkan data sub kategori yang ada di *database*	Dapat menampilkan data sub kategori yang ada di*data base*	[√] Diterima [ ] Ditolak
Klik detail pada data produk	Menampilkan detail produk	Dapat menampilkan detail produk	[√] Diterima [ ] Ditolak
Klik add to cart	Menuju ke keranjang belanja	Dapat menampilkan keranjang belanja	[√] Diterima [ ] Ditolak

4.5.2.10 Pengujian *Form CheckOut*

Berikut dapat dilihat pengujian dari *Form Checkout* pada tabel 4.16 dibawah ini:

Tabel 4.16 Pengujian *Form CheckOut*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tombol *checkout*	Data pembelian yang disimpan di dalam *databese*	Dapat menyimpan data pembelian yang sisimpan di dalam *database*.	[√] Diterima [ ] Ditolak

4.5.2.11 Pengujian Form Pengiriman

Berikut dapat dilihat pengujian dari Form pengiriman pada tabel 4.17 dibawah ini:

Tabel 4.17 Pengujian form Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Pilih jasa pengiriman	Menampilkan form jasa pengiriman	Dapat menampilkan jasa pengiriman	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Jenis pengiriman tidak diisi	Menampilkan pesan kesalahan "Anda belum memilih Jenis pengiriman!"	Dapat menampilkan pesan kesalahan "Anda belum memilih Jenis pengiriman!"	[√] Diterima [ ] Ditolak

4.5.2.12 Pengujian Menu Histori

Berikut dapat dilihat pengujian dari menu history pembayaran pada tabel 4.18 dibawah ini:

Tabel 4.18 Pengujian Menu *History*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu *history*	Menampilkan *history*	Dapat menampilkan form *history*	[√] Diterima [ ] Ditolak
Klik *view*	Menampilkan *detail order*	Dapat menampilkan *detail order*	[√] Diterima [ ] Ditolak

4.5.2.13 Pengujian Form Konfirmasi Pembayaran

Berikut dapat dilihat pengujian dari *Form* konfirmasi pembayaran pada tabel 4.19. dibawah ini:

Tabel 4.19 Pengujian *Form* konfirmasi pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh No. order : 2 Nama Bank :	Menampilkan *form* konfirmasi	Dapat memunculkan pesan form konfirmasi.	[√] Diterima [ ] Ditolak

BCA Atas Nama : Rizal Jumlah Transfer : 125000			
Memasukan data kosong	Tampilkan pesan kesalahan ”*field* ini harus diisi”	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.14 Pengujian Menu Tentang kami

Berikut dapat dilihat pengujian dari menu hubungi pada tabel 4.20 dibawah ini:

Tabel 4.20 Pengujian Menu Tentang kami

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu Tentang kami	Menampilkan form tentang kami	menampilkan form Tentang kami	[√] Diterima [ ] Ditolak
data yang dimasukkan semuanya benar	Menampilkan pesan “Terima Kasih Telah Mengtentang kami”	Dapat menampilkan pesan “Terima Kasih Telah Mengtentang kami”	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji(Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan ada yang tidak diisi	Muncul pesan kesalahan bahwa “field tersebut harus terisi”	Menampilkan pesan kesalahan “field tersebut harus terisi”	[√] Diterima [ ] Ditolak
Nama tidak diisi	Menampilkan pesan kesalahan “nama harus diisi”	Menampilkan pesan kesalahan “nama harus diisi”	[√] Diterima [ ] Ditolak
email tidak diisi	Menampilkan pesan kesalahan “email harus diisi”	Menampilkan pesan kesalahan “email harus diisi”	[√] Diterima [ ] Ditolak
Isi pesan/saran tidak diisi	Menampilkan pesan kesalahan “pesan/saran harus diisi”	Menampilkan pesan kesalahan “pesan/saran harus diisi”	[√] Diterima [ ] Ditolak
Kode belum diisi	Menampilkan pesan kesalahan	Menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

	“kode spam harus diisi”	“kode sapam harus diisi”	

4.5.2.15 Pengujian Login User

Berikut dapat dilihat pengujian dari login user pada tabel 4.21 dibawah ini:

Tabel 4.21 Pengujian login User

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
nama : user	Tercantum pada text box *username*	Dapat mengisi field *username* dengan benar	[√] Diterima [ ] Ditolak
Password : rizal	Tercantum pada text box *password*	Dapat mengisi field *password* dengan benar	[√] Diterima [ ] Ditolak
Klik tombol login	jika sama dengan data login maka user masuk ke menu utama user	Tombol login dapat berfungsi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *email* atau *password* tidak terdaftar	Tidak dapat login	Muncul pesan “maaf account tidak ditemukan.”	[√] Diterima [ ] Ditolak
Data *emaikl* atau *password* kosong	Tidak dapat login	Muncul pesan kesalahan “account tidak ditemukan.”	[√] Diterima [ ] Ditolak
Email tidak diisi	Menampilkan pesan kesalahan “email harus diisi”	Dapat menampilkan pesan kesalahan email harus diisi”	[√] Diterima [ ] Ditolak
Password tidak diisi	Menampilkan pesan kesalahan “*password* harus diisi”	Dapat menampilkan pesan kesalahan “*password* harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.16 Pengujian Ganti Password User

Berikut dapat dilihat pengujian dari ganti *password* user pada tabel 4.22 dibawah ini:

Tabel 4.22 Pengujian ganti *password* user

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu ubah *password*	Menampilkan *form* ubah *password*	Dapat menampilkan *form* ubah *password*	[√] Diterima [ ] Ditolak
Memasukan data yang benar	*Password* baru berhasil disimpan di *database*	Dapat menyimpan data *password* yang baru dengan muncul pesan " Data berhasil diubah"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengosongkan salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan "*field* ini harus diisi" bila ada field yang belum diisi	[√] Diterima [ ] Ditolak
Memasukan *password* yang salah	muncul pesan "password salah"	Dapat menampilkan pesan kesalahan "*Password* salah"	[√] Diterima [ ] Ditolak
Memasukan data *password* baru dan konfirmasi *password* baru yang tidak sama	Muncul pesan "password baru dan konfirmasi password harus sama"	Dapat menampilkan pesan "password baru dan konfirmasi password harus sama"	[√] Diterima [ ] Ditolak
Mengosongkan semua *field*	Muncul pesan kesalahan *field* harus diisi	Dapat menampilkan pesan kesalahan *field* harus diisi	[√] Diterima [ ] Ditolak

4.5.2.17 Pengujian Ubah Profil User

Berikut dapat dilihat pengujian dari ubah profil user pada tabel 4.23 dibawah ini:

Tabel 4.23 Pengujian Ubah Profil User

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik edit profil User	Menampilkan *form* ubah profil	Dapat menampilkan *form* ubah profil	[√] Diterima [ ] Ditolak
Data profil yang akan diubah	Tercantum pada text box data yang akan diubah	Dapat mengisi *field* data yang akan diubah dengan benar	[√] Diterima [ ] Ditolak
Klik tombol ubah	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan ada yang tidak diisi	Menampilkan pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan bahwa field tersebut harus terisi.	[√] Diterima [ ] Ditolak
Email tidak diisi	Muncul pesan kesalahan *email* harus diisi	Muncul pesan kesalahan *email* harus diisi	[√] Diterima [ ] Ditolak
Nama tidak diisi	Muncul pesan kesalahan nama harus diisi	Muncul pesan kesalahan nama harus diisi	[√] Diterima [ ] Ditolak
Alamat tidak diisi	Muncul pesan kesalahan alamat harus diisi	Muncul pesan kesalahan alamat harus diisi	[√] Diterima [ ] Ditolak

4.5.2.18 Pengujian Pengolahan Data Sub kategori

Berikut dapat dilihat pengujian dari pengolahan data sub kategori pada tabel 4.24 dibawah ini:

Tabel 4.24 Pengujian lihat sub kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu sub kategori	Menampilkan *form* lihat sub kategori	Dapat menampilkan *form* lihat sub kategori	[√] Diterima [ ] Ditolak
Klik tombol ubah	Menampilkan *form* ubah sub kategori	Dapat menampilkan *form* ubah sub kategori	[√] Diterima [ ] Ditolak
Klik tombol simpan	Tampil pesan "Data berhasil diubah"	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan *field* harus diisi	Dapat menampilkan pesan kesalahan *field* harus diisi	[√] Diterima [ ] Ditolak

4.5.2.19 Pengujian Tambah kategori

Berikut dapat dilihat pengujian dari tambah kategori pada tabel 4.25 dibawah ini :

Tabel 4.25 Pengujian Tambah kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu Tambah	Menampilkan *form* tambah kategori	Dapat menampilkan *form* tambah kategori	[√] Diterima [ ] Ditolak
Mengisi data sub kategori Contoh: Nama kategori: Lampu	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses	[√] Diterima [ ] Ditolak

	proses tambah kategori	tambah kategori	
Klik tombol simpan	Data sub kategori akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	Data sub kategori akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data nama kategori masih kosong	Tampilkan pesan kesalahan “nama kategori harus isi”	Dapat menampilkan pesan kesalahan “nama kategori harus diisi”	[√] Diterima [ ] Ditolak
Data produk kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

4.5.2.20 Pengujian Pengolahan Data Produk

Berikut dapat dilihat pengujian dari pengolahan data Produk pada tabel 4.26 dibawah ini :

Tabel 4.26 Pengujian Lihat Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu Produk	Menampilkan data Produk	Dapat menampilkan data Produk	[√] Diterima [ ] Ditolak
Klik tombol ubah	Menampilkan *form* ubah Produk	Dapat menampilkan *form* ubah Produk	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

Data produk kosong	Tampilkan pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

4.5.2.21 Pengujian Tambah Data Produk

Berikut dapat dilihat pengujian dari pengolahan data produk pada tabel 4.27 dibawah ini :

Tabel 4.27 Pengujian Tambah Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu Tambah	Menampilkan *form* tambah produk	Dapat menampilkan *form* tambah produk	[√] Diterima [ ] Ditolak
Mengisi data produk Contoh: Sub kategori: Lampu Nama : Lampu butterfly lamp Harga : 100 Warna : biru Diskon : Deskrispsi Gambar :  	Jika semua field terisi semua maka akan muncul pesan “ data berhasil ditambah”	Jika semua field terisi semua maka akan muncul pesan “ data berhasil ditambah”	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data produk akan tersimpan didalam *database* muncul pesan ”data berhasil di tambah”	Data produk akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukkan	Yang diharapkan	Pengamatan	Kesimpulan
Data produk masih kosong	Tampilkan pesan kesalahan “*field*	Dapat menampilkan pesan kesalahan”	[√] Diterima [ ] Ditolak

	tersebut harus diisi"	*field* tersebut harus diisi"	
Nama produk tidak diisi	Muncul pesan kesalahan"nama baramg harus diisi"	Dapat menampilkan pesan kesalahan "nama produk harus diisi"	[√] Diterima [ ] Ditolak
Harga tidak diisi	Muncul pesan kesalahan"harga harus diisi"	Dapat menampilkan pesan kesalahan "harga harus diisi"	[√] Diterima [ ] Ditolak
Warna tidak diisi	Muncul pesan kesalahan"warna harus diisi"	Dapat menampilkan pesan kesalahan "warna harus diisi"	[√] Diterima [ ] Ditolak

4.5.2.22Pengujian Pengolahan Data Provinsi

Berikut dapat dilihat pengujian dari pengolahan data provinsi pada tabel 4.28 dibawah ini:

Tabel 4.28 Pengujian Lihat Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu provinsi	Menampilkan data provinsi	Dapat menampilkan data provinsi	[√] Diterima [ ] Ditolak
Klik tombol ubah	Menampilkan *form* ubah provinsi	Dapat menampilkan *form* ubah provinsi	[√] Diterima [ ] Ditolak
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Data provinsi kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "ada kesalahan didalam pengisian data, silahkan cek kembali"	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.23 Pengujian Tambah Provinsi

Berikut dapat dilihat pengujian dari tambah provinsi pada tabel 4.29 dibawah ini:

Tabel 4.29 Pengujian Tambah Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah provinsi	Menampilkan *form* tambah provinsi	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data provinsi Contoh: Nama Provinsi : Jawa Barat	Jika semua field terisi maka akan muncul pesan "data berhasil ditambah"	Jika semua field terisi maka akan muncul pesan "data berhasil ditambah".	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Nama provinsi masih kosong	Tampilkan pesan kesalahan "nama provinsi harus diisi"	Dapat menampilkan pesan kesalahan " nama provinsi harus diisi"	[√] Diterima [ ] Ditolak

4.5.2.24 Pengujian Pengolahan Data Kota

Berikut dapat dilihat pengujian dari pengolahan data kota pada tabel 4.30 dibawah ini:

Tabel 4.30 Pengujian Lihat Kota

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu kota	Menampilkan data kota	Dapat menampilkan data kota	[√] Diterima [ ] Ditolak
Klik tombol ubah	Menampilkan *form* ubah kota	Dapat menampilkan *form* ubah kota	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

nama kota masih kosong	Tampilkan pesan kesalahan nama kota harus disi	Dapat menampilkan pesan kesalahan nama kota harus disi	[√] Diterima [ ] Ditolak

4.5.2.25 Pengujian Tambah Kota

Berikut dapat dilihat pengujian dari tambah kota pada tabel 4.31 dibawah ini:

Tabel 4.31 Pengujian Tambah Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah kota	Menampilkan *form* tambah kota	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota : Bandung	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
nama kota masih kosong	Tampilkan pesan kesalahan nama kota harus disi	Dapat menampilkan pesan kesalahan nama kota harus disi	[√] Diterima [ ] Ditolak

4.5.2.26 Pengujian Pengolahan Data Merk

Berikut dapat dilihat pengujian dari pengolahan data merk pada tabel 4.32 dibawah ini:

Tabel 4.32 Pengujian Pengolahan Data Merk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik merk	Menampilkan data merk	Dapat menampilkan data merk	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah merk	Dapat menampilkan *form* ubah merk	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi nama merk	Tampilkan pesan kesalahan nama merk harus diisi	Dapat menampilkan pesan kesalahan nama merk harus diisi	[√] Diterima [ ] Ditolak

4.5.2.27 Pengujian Tambah Merk

Berikut dapat dilihat pengujian dari tambah merk pada tabel 4.33 dibawah ini:

Tabel 4.33 Pengujian Tambah Merk

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah merk	Menampilkan *form* tambah merk	Dapat menampilkan *form* tambah merk	[√] Diterima [ ] Ditolak
Mengisi data merk Contoh: Nama merk : S Keterangan : Small	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak

	terisi.		
Klik tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Tidak mengisi nama merk	Tampilkan pesan kesalahan nama merk harus diisi	Dapat menampilkan pesan kesalahan nama merk harus diisi	[√] Diterima [ ] Ditolak

4.5.2.28 Pengujian Data Member

Berikut dapat dilihat pengujian dari pengolahan data member pada tabel 4.34 dibawah ini:

Tabel 4.34 Pengujian Lihat Member

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik member	Menampilkan data member	Dapat menampilkan data member	[√] Diterima [ ] Ditolak
Klik *check box* pada data yang akan dihapus hapus	Memilih data yang akan dihapus	Data yang akan dihapus telah dipilih	[√] Diterima [ ] Ditolak
Klik hapus	Data akan dihapus dari *database*. Jika tidak ada data yang dipilih akan muncul pesan kesalahan	Data telah dihapus dari database. Muncul pesan “pilih data yang akan dihapus” bila data belum dipilih	[√] Diterima [ ] Ditolak

4.5.2.29 Pengujian Pengolahan Data Order

Berikut dapat dilihat pengujian dari lihat order pada tabel 4.35 dibawah ini:

Tabel 4.35 Pengujian Lihat Data Order

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik pengeolahan data order	Menampilkan *form* order	Dapat menampilkan *form* lihat order	[√] Diterima [ ] Ditolak
Klik tabel order pemesanan baru	Menampilkan *form* order pemesanan baru	Dapat menampilkan *form* order pemesanan baru	[√] Diterima [ ] Ditolak
Klik tabel order pembayaran	Menampilkan *form* order pembayaran	Dapat menampilkan *form* order pembayaran	[√] Diterima [ ] Ditolak
Klik tabel order pengiriman	Menampilkan *form* order pengiriman	Dapat menampilkan *form* order pengiriman	[√] Diterima [ ] Ditolak
Klik tabel order berhasil	Menampilkan *form* order berhasil	Dapat menampilkan *form* order berhasil	[√] Diterima [ ] Ditolak
Klik tombol pembayaran pada tab order pemesanan baru	Ubah status order pada tabel order di *database* menjadi telah dibayar	Data status order telah diubah di *database* menjadi telah dibayar	[√] Diterima [ ] Ditolak
Klik tombol detail pada tab order pemesanan baru	Menampilkan *form* detail order	Dapat menampilkan *form* detail order	[√] Diterima [ ] Ditolak
Klik tombol pengiriman pada tab order pembayaran	Ubah status order pada tabel order di *database* menjadi telah dikirim	Data status order telah diubah di *database* menjadi telah dikirim	[√] Diterima [ ] Ditolak
Klik tombol detail pada tab order pembayaran	Menampilkan *form* detail order	Dapat menampilkan *form* detail order	[√] Diterima [ ] Ditolak
Klik tombol berhasil pata tab order pengiriman	Ubah status order pada tabel order di *database* menjadi telah terkirim	Data status order telah diubah di *database* menjadi telah terkirim	[√] Diterima [ ] Ditolak
Klik tombol detail pada tabel order pengiriman	Menampilkan *form* detail order	Dapat menampilkan *form* detail order	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Klik tabel order gagal	Menampilkan *form* order gagal	Dapat menampilkan *form* order gagal	[√] Diterima [ ] Ditolak

4.5.2.30 Pengujian Pengolahan Data Ongkos Kirim

Berikut dapat dilihat pengujian pengolahan data ongkos kirim pada tabel 4.36 dibawah ini:

Tabel 4.36 Pengujian Lihat Data ongkos kirim

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik pengolahan data ongkos kirim	Menampilkan data ongkos kirirm	Dapat menampilkan data kota ongkos kirim	[√] Diterima [ ] Ditolak
Klik tombol ubah	Menampilkan *form* ubah ongkos kirim	Dapat menampilkan *form*ongkos kirim	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Data ongkos kirim kosong	Tampilkan pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

4.5.2.31 Pengujian Tambah Ongkos Kirim

Berikut dapat dilihat pengujian dari tambah ongkos kirim pada tabel 4.37 dibawah ini:

Tabel 4.37 Pengujian Lihat Data Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu Tambah	Menampilkan *form* tambah ongkos kirim	Dapat menampilkan *form* tambah ongkos kirim	[√] Diterima [ ] Ditolak
Mengisi data ongkos kirim Contoh: Provinsi : Jawa Barat Kota : Bandung Jenis pengiriman : Regular service harga: Rp.8000	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah onkos kirim	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah onkos kirim	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data ongkos kirim akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	Data ongkos kirim akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data ongkos kirim masih kosong	Tampilkan pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

4.5.3 Keimpulan Hasil Pengujian Alpa

Berdasarkan hasil pengujian alpha(fungsional) dengan kasus dan *sample* uji yang telah dilakukan, dapat diambil kesimpulan bahwa setiap fungsional proses yang ada pada aplikasi *e-commerce* Toko Batagoro Store telah sesuai. Penyaringan kesalahan proses dalam bentuk tampilan info dan pesan sudah cukup

maksimal. Secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan.

4.6 Pengujian Betha

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuisioner mengenai kepuasan pengguna terhadap aplikasi yang telah dibangunapakah sesuai dengan yang diharapkan.

4.6.1 Skenario Pengujian Beta

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

1. Lingkungan Penjual di batagoro store

Penelitian di lingkungan penjual di Batagoro store 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 order pembelian produk secara *online*.

Berikut ini adalah hasil wawancara dari *user* Toko Batagoro Store , dengan menanyakan pertanyaan dan menyimpulkan jawaban dengan empat kemungkinan jawaban (ya, cukup, kurang, tidak).

4.6.2 Wawancara *Betha* Untuk *User*

1. Bagaimana tampilan perangkat luak untuk pengolahan data?

Jawaban : bagus, dapat mempermudah toko dalam memberi informasi barang yang ada pada Toko Batagoro Store.

2. Apakah Anda mengerti dengan instruksi yang terdapat dalam *website* tersebut mudah dimengerti?

Jawaban : iya, mudah dimengerti dan dapat dipahami dengan baik.

3. Apakah *website* ini dapat membantu dalam pengolahan data produk?

Jawaban : iya, memudahkan dalam pembuatan laporan transaksi dan produk.

4. Apakah website ini membantu dalam pengoalahan data penjualan secara *online?*

Jawaban : membantu pengolahan jadi tidak adanya penumpukan dokumen.

5. Apakah *website* ini dapat membantu dalam membuat laporan penjualan ?

Jawaban : memudahkan dalam pembuatan laporan penjualan produk yang keluar.

4.6.3 Wawancara Pengujian *Beta* Untuk operator

1. Menurut anda apakah halaman operator pada *website e-commerce* Batagoro Store ini sudah mampu memenuhi kebutuhan anda dalam pengelolahan data?

Jawaban: Ya, dengan adanya halaman operator ini saya merasa terbantu untuk mengelola berbagai macam data yang ada.

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa halaman operator pada *website e-commerce* Batagoro Store ini sudah mampu memenuhi kebutuhan operator dalam pengelolaan data.

2. Menurut anda apakah tampilan antarmuka halaman operator pada *website E-Commerce* Batagoro Store mudah untuk anda pahami ?

Jawaban: Ya, tampilan halaman operator mudah untuk saya pahami.

Berdasarkan hasil wawancara diatas maka dapat disimpulkan bahwa halaman operator dalam *website* ini mudah untuk dipahami operator.

3. Menurut pendapat Anda, apakah halaman operator dalam *website* ini dapat membantu Anda dalam mengelola data penjualan secara *online*?

Jawaban: Ya, halaman operator dalam *website* ini sangat membantu dalam mengelola data penjualan secara *online.*

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

4. Menurut pendapat Anda, apakah halaman operator dalam *website* inidapat membantu Anda dalam pembuatan laporan produk dan laporan transaksi penjualan ?

Jawaban: Ya, dengan *website* ini membatu saya dalam pembuatan laporan produk dan transaksi penjualan. Membuat laporan menjadi lebih mudah.

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

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

Jawaban: Saya tidak banyak menemukan kesulitan dalam mengoprasikan halaman operator ini, penjelasannya pun mudah dimengerti.

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

4.6.4 Kuesioner Pengujian *Beta* Untuk Pengguna

Kuesioner disebarkan menggunakan teknik *quota sampling* dimana pemilihan sampel dengan memilih sejumlah tertentu (kuota) dan paling mudah diperoleh seperti yang dikehendaki oleh peneliti. Kuesioner disebarkan kepada 10 orang, dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat mengambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuesioner terdiri dari 5 pertanyaan dengan menggunakan skala 1 sampai 5 (skala likert).

Tabel 4. 38 Tabel Bobot

Keterangan	**Bobot**
Sangat Setuju	5
Cukup Setuju	4
Setuju	3
Kurang Setuju	2
Tidak Setuju	1

Untuk mencari nilai persentase dari masing – masing jawaban kuesioner digunakan rumus skala likert sebagai berikut :

$$Y = \frac{S}{skorideal} \times 100 \qquad (4.1)$$

Keterangan :

Y = Nilai persentase yang dicari

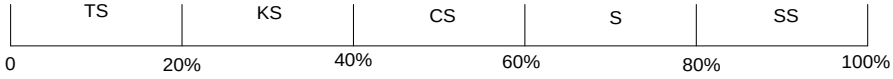
S = Total yang diperoleh dari S = $\sum_{i=1}^{5} n_i . r_i$

r_i = Banyak responden yang menjawab jawaban ke-i

n = Nilai jawaban ke-I

skor ideal = Nilai tertinggi dikalikan dengan jumlah sampel atau responden

Dibawah ini merupakan skala untuk mengmbil kesimpulan dari setiap petanyaan



Berikut ini adalah hasil persentase hasil masing – masing jawaban yang sudah dihitung nilainya dengan menggunakan rumus diatas. Kuesioner ini diuji kepada 10 orang.

1. Apakah Anda setuju *bahwa tampilan website* Batagoro Store mudah untuk dipahami / *user friendly*?

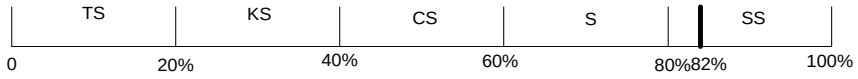
Tabel 4. 39 Hasil Pengujian Kuesioner soal nomor 1 (Situs *Frontend*)

Pertanyaan	Keterangan	Nilai	Responden	$n_i.r_i$
1	Sangat Setuju	5	4	20
	Setuju	4	3	12
	Cukup Setuju	3	3	9
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**10**	**41**

Berdasarkan data dari hasil kuesioner tersebut, dapat dicari prosentase jawaban dengan menggunakan rumus :

$$Y = \frac{41}{10\ x\ 5} \times 100 = \ 82\ \%$$

Skala



Berdasarkan hasil persentase skala diatas maka dapat disimpulkan 82% menyatakan sangat setuju bahwa tampilan *website* Batagoro Store sudah cukup bagus.

2. Apakah anda setuju bahwa *website* E-Commerce ini dapat membantu Anda untuk mendapatkan informasi terbaru tentang produk Toko Batagoro Store?

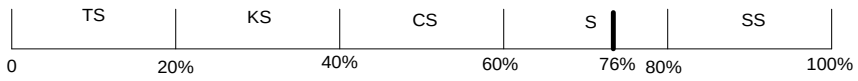
Tabel 4. 40 Hasil Pengujian Kuesioner soal nomor 2 (Situs *Frontend*)

Pertanyaan	**Keterangan**	**Nilai**	**Responden**	**$n_i.r_i$**
2	Sangat Membantu	5	2	10
	Membantu	4	5	20
	Cukup Membantu	3	2	6
	Kurang Membantu	2	1	2
	Tidak Membantu	1	0	0
Jumlah			**10**	**38**

Berdasarkan data dari hasil kuesioner tersebut, dapat dicari prosentase jawaban dengan menggunakan rumus :

$$Y = \frac{38}{10\ x\ 5} \times 100 = \ 76\ \%$$

Skala



Berdasarkan hasil persentase skala diatas maka dapat disimpulkan 76% menyatakan *website* E-Commerce Batagoro Store dapat membantu untuk mendapatkan informasi terbaru tentang produk Toko Batagoro Store.

3. Apakah Anda setuju bahwa website ini akan lebih mempermudah anda dalam melakukan transaksi pembelian dibandingkan transaksi manual?

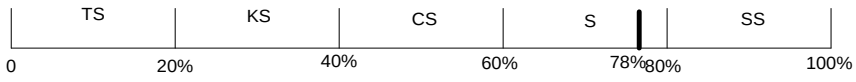
Tabel 4. 41 Hasil Pengujian Kuesioner soal nomor 3 (Situs *Frontend*)

Pertanyaan	Keterangan	Nilai	Responden	$n_i.r_i$
3	Sangat Setuju	5	3	15
	Setuju	4	4	16
	Cukup Setuju	3	2	6
	Kurang Setuju	2	1	2
	Tidak Setuju	1	0	0
Jumlah			**10**	**39**

Berdasarkan data dari hasil kuesioner tersebut, dapat dicari prosentase jawaban dengan menggunakan rumus :

$$Y = \frac{39}{10\ x\ 5} \times 100 = \ 78\ \%$$

Skala



Berdasarkan hasil presentase skala diatas maka dapat disimpulkan 78% menyatakan setuju informasi yang ada pada *website E-Commerce* Batagoro Store .

4. Apakah Anda setuju *website* ini pemilik dapat memperluas pangsa pasarnya ?

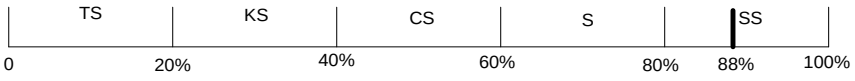
Tabel 4. 42 Hasil Pengujian Kuesioner soal nomor 4 (Situs *Frontend*)

Pertanyaan	Keterangan	Nilai	Responden	$n_i.r_i$
4	Sangat Setuju	5	5	25
	Setuju	4	4	16
	Cukup Setuju	3	1	3
	Kurang Setuju	2	0	0
	Tidak Setuju	1	0	0
Jumlah			**10**	**44**

Berdasarkan data dari hasil kuesioner tersebut, dapat dicari prosentase jawaban dengan menggunakan rumus :

$$Y = \frac{44}{10\ x\ 5} \times 100 = \ 88\ \%$$

Skala



Berdasarkan hasil persentase skala diatas maka dapat disimpulkan 88% menyatakan sangat setuju ini pemilik dapat memperluas pangsa pasarnya

5. Apakah Anda setuju bahwa sistem pembayaran yang digunakan lebih memudahkan anda ?

Tabel 4. 43 Hasil Pengujian Kuesioner soal nomor 5 (Situs *Frontend*)

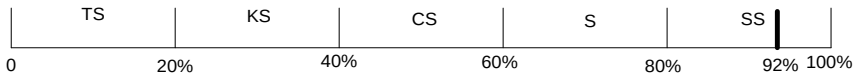
Pertanyaan	Keterangan	Nilai	Respo nden	$n_i.r_i$
5	Sangat Setuju	5	6	30
	Setuju	4	4	16
	Cukup Setuju	3	0	0
	Kurang Setuju	2	0	0

	Tidak Setuju	1	0	0
Jumlah			**10**	**46**

Berdasarkan data dari hasil kuesioner tersebut, dapat dicari prosentase jawaban dengan menggunakan rumus :

$$Y = \frac{46}{10\ x\ 5} \times 100 = \ 92\ \%$$

Skala



Berdasarkan hasil persentase skala diatas maka dapat disimpulkan 92% menyatakan sangat setuju bahwa *website e-commerce* batagoro store dapat memudahkan pembayaran.

4.6.5 Kesimpulan Pengujian Betha

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

1. *Website e-commerce* ini memudahkan konsumen untuk bertransaksi *online.*
2. Melalui Website *E-commrce* Batagoro Store sudah bisa memberikan informasi dan melakukan promosi atas produk-produk yang dijualnya kepada konsumen.
3. *Website e-commerce* Batagoro Store membantu konsumen dalam melakukan pembelian produk terutama yang berada diluar kota Bandung.