BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

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

4.1.1 Perangkat Keras Yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam pembangunan *E-commerce* pada toko Mega Batik adalah tercantum pada tabel 4.1 sebagai berikut :

Tabel 4.1. Perangkat keras yang digunakan

Perangkat	Spesifikasi
Processor	Core 2 Duo 2.6 GHz
RAM	2 GHz
Harddisk	160 Gb
Monitor	Resolusi 1280 x 800
VGA	512 Mb

4.1.2 Perangkat Lunak Yang Digunakan

Spesifikasi perangkat lunak yang digunakan dalam membangun *E-commerce* pada toko Mega Batik adalah sebagai berikut:

Tabel 4.2. Perangkat Lunak yang digunakan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP SP 2
2	Bahasa Pemrograman	PHP
3	*DBMS*	MySQL 5.0.27
4	*Web browser*	Mozilla firefox 3.5.5
5	*Code editor*	Macromedia Dreamweaver 8
6	DFD modeler	Microsoft Visio 2007

4.2 Implementasi database

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

1. Tabel Admin

```
CREATE TABLE IF NOT EXISTS `admin` (
  `id_admin` int(11) NOT NULL AUTO_INCREMENT,
  `nama_admin` varchar(50) NOT NULL,
  `alamat_admin` text NOT NULL,
  `email_admin` varchar(50) NOT NULL,
  `telp_admin` varchar(20) NOT NULL,
  `password_admin` varchar(32) NOT NULL,
  `verification_admin` varchar(32) NOT NULL,
  PRIMARY KEY (`id_admin`)
)ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

2. Tabel Member

```
CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(11) NOT NULL AUTO_INCREMENT,
  `id_kota` int(11) NOT NULL,
  `nama_member` varchar(50) NOT NULL,
  `alamat_member` text NOT NULL,
  `telp_member` varchar(20) NOT NULL,
  `kodepos_member` varchar(6) NOT NULL,
  `email_member` varchar(50) NOT NULL,
  `password_member` varchar(32) NOT NULL,
  `verificationcode_member` varchar(32) NOT NULL,
  `status_member` enum('0','1') NOT NULL,
  PRIMARY KEY (`id_member`),
  FOREIGN KEY `id_kota` REFERENCES kota (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```

3. Tabel Kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `nama_kategori` varchar(50) NOT NULL,
  `deskripsi_kategori` text NOT NULL,
  PRIMARY KEY (`id_kategori`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

4. Tabel Barang

```
CREATE TABLE IF NOT EXISTS `barang` (
  `id_barang` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_kategori` int(11) NOT NULL,
  `id_jenis` int(11) NOT NULL,
  `id_motif` int(11) NOT NULL,
  `nama_barang` varchar(50) NOT NULL,
  `harga_barang` int(11) NOT NULL,
  `berat_barang` float NOT NULL,
  `deskripsi_barang` text NOT NULL,
  `promo_barang` enum('0','1') NOT NULL,
  `diskon_barang` int(11) NOT NULL,
  `rating_barang` float NOT NULL,
  `voterrating_barang` int(11) NOT NULL,
  `viewcounter_barang` int(11) NOT NULL,
  `gambar_barang` varchar(200) NOT NULL,
  PRIMARY KEY (`id_barang`),
  FOREIGN KEY `id_kategori` REFERENCES kategori
(`id_kategori`),
  FOREIGN KEY 2 `id_jenis` REFERENCES jenis (`id_jenis`),
  FOREIGN KEY 3 `id_motif` REFERENCES motif (`id_motif`),
  FOREIGN KEY 4 `id_pegawai` REFERENCES
pegawai(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;
```

5. Tabel Detail Barang

```
CREATE TABLE IF NOT EXISTS `detailbarang` (
  `id_detailbarang` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_barang` int(11) NOT NULL,
  `id_ukuran` int(11) NOT NULL,
  `stok_barang` int(11) NOT NULL,
  PRIMARY KEY (`id_detailbarang`),
  FOREIGN KEY  `id_barang` REFERENCES barang (`id_barang`),
  FOREIGN KEY 2 `id_ukuran` REFERENCES ukuran (`id_ukuran`),
  FOREIGN KEY 3 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
)ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=68 ;
```

6. Tabel Detail Pembelian

```
CREATE TABLE IF NOT EXISTS `detailpembelian` (
  `id_detailpembelian` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_pembelian` int(11) NOT NULL,
  `id_detailbarang` int(11) NOT NULL,
  `harga_temp` int(11) NOT NULL,
  `berat_temp` float NOT NULL,
```

```
  `diskon_temp` int(11) NOT NULL,
  `qty` int(11) NOT NULL,
  PRIMARY KEY (`id_detailpembelian`),
  FOREIGN KEY `id_pembelian` REFERENCES pembelian
(`id_pembelian`),
  FOREIGN KEY 2 `id_detailbarang` REFERENCES detailbarang
(`id_detailbarang`),
  FOREIGN KEY 3 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=39 ;
```

7. Tabel Jenis Pengiriman

```
CREATE TABLE IF NOT EXISTS `jenispengiriman` (
  `id_jenispengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_jasapengiriman` int(11) NOT NULL,
  `nama_jenispengiriman` varchar(50) NOT NULL,
  `deskripsi_jenispengiriman` text NOT NULL,
  PRIMARY KEY (`id_jenispengiriman`),
  FOREIGN KEY `id_jasapengiriman` REFERENCES  jasapengiriman
(`id_jasapengiriman`),
  FOREIGN KEY 2 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

8. Tabel Kota

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_provinsi` int(11) NOT NULL,
  `nama_kota` varchar(50) NOT NULL,
  PRIMARY KEY (`id_kota`),
  FOREIGN KEY  `id_provinsi` REFERENCES provinsi
(`id_provinsi`),
  FOREIGN KEY 2 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=441 ;
```

9. Tabel Kurs

```
CREATE TABLE IF NOT EXISTS `kurs` (
  `id_kurs` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `kode_kurs` varchar(5) NOT NULL,
  `simbol_kurs` varchar(5) NOT NULL,
  `harga_kurs` int(11) NOT NULL,
  `deskripsi_kurs` text NOT NULL,
```

```
  `tanggal_kurs` date NOT NULL,
  PRIMARY KEY (`id_kurs`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

10. Tabel Jenis

```
CREATE TABLE IF NOT EXISTS `jenis` (
  `id_jenis` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `nama_jenis` varchar(50) NOT NULL,
  PRIMARY KEY (`id_jenis`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=7 ;
```

11. Tabel Ongkos Kirim

```
CREATE TABLE IF NOT EXISTS `ongkoskirim` (
  `id_ongkoskirim` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_kota` int(11) NOT NULL,
  `id_jenispengiriman` int(11) NOT NULL,
  `harga_ongkoskirim` int(11) NOT NULL,
  PRIMARY KEY (`id_ongkoskirim`),
  FOREIGN KEY  `id_kota` REFERENCES kota (`id_kota`),
  FOREIGN KEY 2 `id_jenispengiriman` REFERENCES
jenispengiriman (`id_jenispengiriman`),
  FOREIGN KEY 3 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=12 ;
```

12. Tabel Pembelian

```
CREATE TABLE IF NOT EXISTS `pembelian` (
  `id_pembelian` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_member` int(11) NOT NULL,
  `id_jenispengiriman` int(11) NOT NULL,
  `tanggal_pembelian` datetime NOT NULL,
  `status_pengiriman`
enum('dipesan','dikonfirmasi','dibayar','dikirim','diterima'
) NOT NULL,
  `keterangan_pembelian` text NOT NULL,
  `nama_pemesan` varchar(50) NOT NULL,
  `alamat_pemesan` text NOT NULL,
  `kota_pemesan` varchar(100) NOT NULL,
```

```
  `email_pemesan` varchar(50) NOT NULL,
  `no_telp_pemesan` varchar(20) NOT NULL,
  `kodepos_pemesan` varchar(6) NOT NULL,
  `ongkoskirim_pembelian` int(11) NOT NULL,
  `totalharga_pembelian` int(11) NOT NULL,
  `no_resi_pemesan` varchar(32) NOT NULL,
  `status_pemesanan` enum('ok','cancel') NOT NULL,
  `pencetakan_label` varchar(50) NOT NULL,
  PRIMARY KEY (`id_pembelian`),
  FOREIGN KEY `id_member` REFERENCES member (`id_member`),
  FOREIGN KEY 2 `id_jenispengiriman` REFERENCES
jenispengiriman (`id_jenispengiriman`),
  FOREIGN KEY 3 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1260;
```

13. Tabel Provinsi

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `nama_provinsi` varchar(50) NOT NULL,
  PRIMARY KEY (`id_provinsi`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=34 ;
```

14. Tabel Rekening

```
CREATE TABLE IF NOT EXISTS `rekening` (
  `id_rekening` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `nama_rekening` varchar(50) NOT NULL,
  `bank_rekening` varchar(50) NOT NULL,
  `cabang_rekening` varchar(100) NOT NULL,
  `no_rekening` varchar(32) NOT NULL,
  `gambar_rekening` varchar(200) NOT NULL,
  PRIMARY KEY (`id_rekening`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;
```

15. Tabel Retur

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_pembelian` int(11) NOT NULL,
```

```
  `tanggal_retur` date NOT NULL,
  `status_retur` enum('0','1','2','3','4') NOT NULL DEFAULT
'0',
  PRIMARY KEY (`id_retur`),
  FOREIGN KEY `id_pegawai` (`id_pegawai`),
  FOREIGN KEY 2 `id_pembelian` REFERENCES pembelian
(`id_pembelian`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

16. Tabel Detail Retur

```
CREATE TABLE IF NOT EXISTS `detailretur` (
  `id_detailretur` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_retur` int(11) NOT NULL,
  `id_detailbarang` int(11) NOT NULL,
  `qty_barang` int(11) NOT NULL,
  `alasan_retur` text NOT NULL,
  PRIMARY KEY (`id_detailretur`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai
(`id_pegawai`),
  FOREIGN KEY 2 `id_retur` (`id_retur`),
  FOREIGN KEY 3 `id_detailbarang` REFERENCES  detailbarang
(`id_detailbarang`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

17. Tabel Jasa Pengiriman

```
CREATE TABLE IF NOT EXISTS `jasapengiriman` (
  `id_jasapengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `nama_jasapengiriman` varchar(50) NOT NULL,
  `deskripsi_jasapengiriman` text NOT NULL,
  PRIMARY KEY (`id_jasapengiriman`),
  FOREIGN KEY `id_pegawai` REFERENCES  pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=6 ;
```

18. Tabel Motif

```
CREATE TABLE IF NOT EXISTS `motif` (
  `id_motif` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
```

```
  `nama_motif` varchar(50) NOT NULL,
  PRIMARY KEY (`id_motif`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

19. Tabel Pembayaran

```
CREATE TABLE IF NOT EXISTS `pembayaran` (
  `id_pembayaran` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_pembelian` int(11) NOT NULL,
  `id_rekening` int(11) NOT NULL,
  `tanggal_pembayaran` datetime NOT NULL,
  `jenis_pembayaran` enum('paypal','rekening') NOT NULL,
  `bank_pembayaran` varchar(50) NOT NULL,
  `no_rekening_pembayaran` varchar(20) NOT NULL,
  `atasnama_pembayaran` varchar(50) NOT NULL,
  `total_pembayaran` float NOT NULL,
  PRIMARY KEY (`id_pembayaran`),
  FOREIGN KEY `id_rekening` REFERENCES rekening
(`id_rekening`),
  FOREIGN KEY 2 `id_pembelian` REFERENCES pembelian
(`id_pembelian`),
  FOREIGN KEY 3 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=15 ;
```

20. Tabel Pengiriman

```
CREATE TABLE IF NOT EXISTS `pengiriman` (
  `id_pengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `id_pembelian` int(11) NOT NULL DEFAULT '1',
  `id_retur` int(11) NOT NULL DEFAULT '1',
  `no_resi_pengiriman` varchar(30) NOT NULL,
  `tujuan_pengiriman` varchar(50) NOT NULL,
  `penerima_pengiriman` varchar(50) NOT NULL,
  `waktu_pengiriman` varchar(30) NOT NULL,
  `status_pengiriman` varchar(30) NOT NULL,
  PRIMARY KEY (`id_pengiriman`),
  FOREIGN KEY `id_retur` REFERENCES retur (`id_retur`),
  FOREIGN KEY 2 `id_pembelian` REFERENCES pembelian
(`id_pembelian`),
  FOREIGN KEY 3 `id_pegawai` REFERENCES pegawai
(`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=16 ;
```

21. Tabel Ukuran

```
CREATE TABLE IF NOT EXISTS `ukuran` (
  `id_ukuran` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
```

```
  `nama_ukuran` varchar(50) NOT NULL,
  PRIMARY KEY (`id_ukuran`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

22. Tabel Operator

```
CREATE TABLE IF NOT EXISTS `operator` (
  `id_operator` int(11) NOT NULL AUTO_INCREMENT,
  `nama_operator` varchar(50) NOT NULL,
  `alamat_operator` text NOT NULL,
  `email_operator` varchar(50) NOT NULL,
  `telp_operator` varchar(20) NOT NULL,
  `password_operator` varchar(32) NOT NULL,
  `verification_operator` varchar(32) NOT NULL,
  PRIMARY KEY (`id_operator`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

23. Tabel Kontak

```
CREATE TABLE IF NOT EXISTS `kontak` (
  `id_kontak` int(11) NOT NULL AUTO_INCREMENT,
  `id_pegawai` int(11) NOT NULL DEFAULT '1',
  `email` varchar(50) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `telepon` varchar(15) NOT NULL,
  `pesan` text NOT NULL,
  `tanggal` datetime NOT NULL,
  `status` enum('Dibaca','Belum dibaca') NOT NULL DEFAULT
'Belum dibaca',
  PRIMARY KEY (`id_kontak`),
  FOREIGN KEY `id_pegawai` REFERENCES pegawai (`id_pegawai`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

24. Tabel Pegawai

```
CREATE TABLE IF NOT EXISTS `pegawai` (
  `id_pegawai` int(11) NOT NULL AUTO_INCREMENT,
  `jenis` enum('admin','operator') NOT NULL,
  `id_admin` int(11) NOT NULL,
  `id_operator` int(11) NOT NULL,
  PRIMARY KEY (`id_pegawai`),
  FOREIGN KEY `id_admin` REFERENCES admin (`id_admin`),
  FOREIGN KEY 2 `id_operator` REFERENCES operator
(`id_operator`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;
```

4.3 Implementasi Antar Muka

Implementasi antar muka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna. Adapun untuk tampilan implementasi Pembangunan *E-commerce* pada toko Mega Batik akan dilampirkan pada halaman lampiran.

Tabel 4.3 Implementasi antar muka

Menu	Deskripsi	Nama File
Login administrator	Digunakan sebagai halaman login administrator	login.php
kota	Digunakan untuk melihat, menambah, dan merubah data kota	kota.php
provinsi	Digunakan untuk melihat, menambah, dan merubah data provinsi	provinsi.php
kategori	Digunakan untuk melihat, menambah, dan merubah data kategori	kategori.php
barang	Digunakan untuk melihat, menambah, dan merubah data barang	barang.php
Lihat transaksi	Digunakan untuk melihat data transaksi, data detail transaksi, dan mengubah status transaksi.	datatransaksi.php
Registrasi *member*	Digunakan untuk melakukan pendaftaran *member*	register.php
Login *member*	Digunakan sebagai halaman login *member*	login.php
Ganti *password member*	Digunakan jika *member* ingin merubah *password*	account.password.php
Ubah Profil	Digunakan untuk mengubah profil *member*	account.change.php
Barang	Digunakan untuk melihat seluruh data barang yang ada	produk.php
Profil	Digunakan untuk melihat data profil perusahaan	about.php
help	Digunakan untuk melihat	help.php

	data help	
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus detail transaksi yang akan dilakukan	cart.php
History	Digunakan untuk melihat *history* transaksi yang pernah dilakukan	account.history.php
Logout	Digunakan User (operator, administrator, *member*)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 Alpa

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

4.5.1 Skenario Pengujian Alpa

Tabel 4.4. menjelaskan skenario pengujian aplikasi *member*.

Tabel 4.4. 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*
Ganti 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 profil	Menampilkan profil perusahaan	*Black Box*
Menu help	Menampilkan data hubungi	*Black Box*
Menu cara pemesanan	Menampilkan cara pemesanan	*Black Box*
Menu keranjang belanja	Menampilkan data detail transaksi	*Black Box*
	Mengubah data detail transaksi	*Black Box*
	Menghapus data detail transaksi	*Black Box*
Form konfirmasi	Menampilkan konfirmasi transaksi	*Black Box*

Form pengiriman	Menampilkan form pengiriman	*Black Box*
	Isi form pengiriman	*Black Box*
Form cara pembayaran	Menampilkan cara pembayaran	*Black Box*
Menu history	Menampilkan data transaksi yang pernah dilakukan	*Black Box*
	Menampilkan data detail tiap transaksi	*Black Box*

Tabel 4.5. menjelaskan skenario pengujian aplikasi administrator.

Tabel 4.5. Skenario pengujian aplikasi administrator

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Login	Isi data login	
	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Logout	Menghapus *session user*	*Black Box*
Ubah *password*	Menampilkan *form* ubah *password*	*Black Box*
	Verifikasi *password* lama	
	Verifikasi *password* baru	*Black Box*
	Verfikasi konfirmasi *password*	*Black Box*
Jenis pengiriman	Tampilkan jenis pengiriman	*Black Box*
	Tambah jenis pengiriman	*Black Box*
	Ubah jenis pengiriman	*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*
Data transaksi	Tampilkan data transaksi	*Black Box*
	Ubah status data transaksi	*Black Box*
	Lihat data detail transaksi	*Black Box*

Data kategori	Tampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
Data produk	Tampilkan data produk	*Black Box*
	Tambah data produk	*Black Box*
	Ubah data produk	*Black Box*
Data Laporan	Tampilkan data laporan	*Black Box*
	Print data laporan	*Black Box*

4.5.2 Kasus dan Hasil Pengujian

4.5.2.1 Pengujian Registrasi member

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

Tabel 4.6. Pengujian Registrasi *member*

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu registrasi *member*	Tampil form registrasi *member*	Dapat menampilkan form registrasi *member*	[√] Diterima [ ] Ditolak
Memasukan data pendaftaran, contoh masukan : Nama: Ricki Pratama Dijaya Alamat Email : if.10106110@yahoo.co.id Provinsi : Jawa Barat Kota : Cimahi Alamat : Cimahi Kodepos : 40511 Telepon : 085294493552 Password : delete Konfirmasi password : delete	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	Jika salah satu field tidak terisi atau salah	Tampil pesan “Anda telah berhasil	[√] Diterima [ ] Ditolak

	dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi. Jika data masukan benar muncul pesan "Anda telah berhasil melakukan registrasi"	melakukan registrasi" jika data masukan benar dan dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	
Memasukan data kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*field* ini harus diisi"	[√] Diterima [ ] Ditolak

4.5.2.2 Pengujian Login *Member*

Berikut dapat dilihat pengujian login *member* pada tabel 4.7 dibawah ini:

Tabel 4.7 Pengujian login *member*

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan alamat *email* : if.10106110@yahoo.co.id	Mengisikan data alamat *email* pada *field* alamat *email*. Jika salah maka akan muncul pesan "invalid alamat email"	Dapat mengisi *field* alamat *email* dan muncul pesan "invalid alamat email" jika salah dalam pengisian data	[√] Diterima [ ] Ditolak
Memasukan *Password* : delete	Mengisikan data *password* pada *field password*. Jika	Dapat mengisi *field password* dan muncul pesan	[√] Diterima [ ] Ditolak

	salah maka akan muncul pesan "diantara 6 sampai 11 karakter diperbolehkan"	"diantara 6 sampai 11 karakter diperbolehkan" jika salah dalam pengisian data	
Klik tombol login	Data *email* dan data *password* dicari	Tombol login dapat berfungsi	[√] Diterima [ ] Ditolak
Memasukan data *email* atau *password* yang tidak terdaftar	Tidak dapat login	Muncul pesan "*email* atau *password* salah"	[√] Diterima [ ] Ditolak
mengosongkan data *email* atau *password*	Tidak dapat login	Dapat menampilkan pesan kesalahan "*field* ini harus diisi"	[√] Diterima [ ] Ditolak

4.5.2.3 Pengujian Logout

Berikut dapat dilihat pengujian dari logout pada tabel 4.8. dibawah ini:

Tabel 4.8. Pengujian logout

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik logout	Dapat keluar dan menghapus session pengguna	*User* keluar dari aplikasi	[√] Diterima [ ] Ditolak

4.5.2.4 Pengujian Ubah profil member

Berikut dapat dilihat pengujian dari ubah profil member pada tabel 4.9. dibawah ini:

Tabel 4.9. Pengujian ubah profil member

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik ubah profil member	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 ubah	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Mengosongkan salah satu fied	Menampilkan pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan bahwa field tersebut harus terisi.	[√] Diterima [ ] Ditolak
Mengosongkan semua field	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*field* ini harus diisi"	[√] Diterima [ ] Ditolak

4.5.2.5 Pengujian ganti *password member*

Berikut dapat dilihat pengujian dari ganti *password member* pada tabel 4.10. dibawah ini:

Tabel 4.10. Pengujian ganti *password member*

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik ubah *password*	Menampilkan *form* ubah *password*	Dapat menampilkan *form* ubah *password*	[√] Diterima [ ] Ditolak
Memasukan data yang benar	*Password* baru berhasil	Dapat menyimpan data	[√] Diterima [ ] Ditolak

	disimpan di *database*	*password* yang baru dengan muncul pesan " Data berhasil diubah"	
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
Data *password* lama salah	muncul pesan "password salah"	Dapat menampilkan pesan kesalahan "*Password* salah"	[√] Diterima [ ] Ditolak
Data *password* baru dah konfirmasi *password* baru 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

4.5.2.6 Pengujian menu barang

Berikut dapat dilihat pengujian dari menu barang pada tabel 4.11. dibawah ini:

Tabel 4.11. Pengujian menu barang

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu barang	Menampilkan data barang yang ada di *database*	Dapat menampilkan data barang yang ada di *data base*	[√] Diterima [ ] Ditolak
Klik detail pada data barang	Menampilkan detail barang	Dapat menampilkan detail barang	[√] Diterima [ ] Ditolak

Klik pesan	Menuju ke keranjang belanja	Dapat menampilkan keranjang belanja	[√] Diterima [ ] Ditolak

4.5.2.7 Pengujian menu profil

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

Tabel 4.12. Pengujian menu profil

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik Profil	Menampilkan menu profil	Tampil menampilkan menu profil	[√] Diterima [ ] Ditolak

4.5.2.8 Pengujian menu help

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

Tabel 4.13. Pengujian menu help

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik help	Menampilkan menu help	Tampil menampilkan menu hubungi	[√] Diterima [ ] Ditolak
Klik selangkapnya	Menampilkan menu help selangkapnya	Menampilkan menu hubungi selengkapnya	[√] Diterima [ ] Ditolak

4.5.2.9 Pengujian menu kategori

Berikut dapat dilihat pengujian menu kategori pada tabel 4.14. dibawah ini:

Tabel 4.14. Pengujian menu kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu kategori	Menampilkan data barang yang ada di *database*	Dapat menampilkan data barang	[√] Diterima [ ] Ditolak

		yang ada di*data base*	
Klik detail pada data barang	Menampilkan detail barang	Dapat menampilkan detail barang	[√] Diterima [ ] Ditolak
Klik pesan	Menuju ke keranjang belanja	Dapat menampilkan keranjang belanja	[√] Diterima [ ] Ditolak

4.5.2.10 Pengujian menu keranjang belanja

Berikut dapat dilihat pengujian dari menu keranjang belanja pada tabel 4.15. dibawah ini:

Tabel 4.15. Pengujian meu *shopping cart*

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu keranjang belanja	Menampilkan *form* keranjang belanja	Dapat menampilkan *form* keranjang belanja	[√] Diterima [ ] Ditolak
Menampilkan data jumlah pesanan	Tercantum pada text box data yang akan diubah	Dapat mengisi data jumlah dengan benar	[√] Diterima [ ] Ditolak
Klik tombol *edit*	Data jumlah yang diubah tersimpan *didatabase*	Dapat menampilkan perubahan data jumlah	[√] 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

4.5.2.11 Pengujian *form* pengiriman

Berikut dapat dilihat pengujian form pengiriman pada tabel 4.16. dibawah ini:

Tabel 4.16. Pengujian form pengiriman

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data yang benar lalu mengklik tombol ganti	Data pengirimah baru akan disimpan di dalam *database*	Dapat data pengirimah baru akan disimpan di dalam *database*	[√] Diterima [ ] Ditolak
Mengosongkan salah satu *field*	Menampilkan pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Dapat kembali ke menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.12 Pengujian *form checkout*

Berikut dapat dilihat pengujian dari *form checkout* pada tabel 4.17 dibawah ini:

Tabel 4.17 Pengujian *form checkout*

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik *checkout*	Data transaksi dan detail transaksi tersimpan di *database*, Menampilkan *form* terimakasih	Dapat menyimpan data transaksi, dapat menampilkan *form* terima kasih	[√] Diterima [ ] Ditolak

4.5.2.13 Pengujian *form* konfirmasi pembayaran

Berikut dapat dilihat pengujian dari *form* konfirmasi pembayaran pada tabel 4.18 dibawah ini:

Tabel 4.18 Pengujian *form* konfirmasi pembayaran

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh No. invoice : 2325657 No. transfer : 24 Nama Bank : BCA Atas Nama : Ricki Pratama DIjaya Jumlah Transfer : 100000	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
Mengosongkan semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.5.2.14 Pengujian menu *history*

Berikut dapat dilihat pengujian dari *history* pada tabel 4.19. dibawah ini:

Tabel 4.19. Pengujian menu *history*

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

4.5.2.15 Pengujian Login Administrator

Berikut dapat dilihat pengujian login administrator pada tabel 4.20 dibawah ini:

Tabel 4.20 Pengujian login administrator

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
email: megabatik@yahoo.com	Tercantum pada text box *username*	Dapat mengisi field *username* dengan benar	[√] Diterima [ ] Ditolak
Password : delete	Tercantum pada text box *password*	Dapat mengisi field *password* dengan benar	[√] Diterima [ ] Ditolak
Klik tombol login	Data *username* dan data *password* dicari	Tombol login dapat berfungsi	[√] Diterima [ ] Ditolak
Memasukan data *username* atau *password* yang tidak terdaftar	Tidak dapat login	Muncul pesan "Tidak Sama Untuk Username Dan/Atau Password."	[√] Diterima [ ] Ditolak
Tidak memasukan data *username* atau *password*	Tidak dapat login	Dapat menampilkan pesan kesalahan "Tidak Sama Untuk Username Dan/Atau Password."	[√] Diterima [ ] Ditolak

4.5.2.16 Pengujian data pengiriman

Berikut dapat dilihat pengujian data pengiriman pada tabel 4.21 dibawah ini:

Tabel 4.21 Pengujian data pengiriman

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik pengiriman	Menampilkan data kota pengiriman	Dapat menampilkan data kota pengiriman	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah kota pengiriman	Dapat menampilkan *form* kota pengiriman	[√] Diterima [ ] Ditolak
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	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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.17 Pengujian tambah jenis pengiriman

Berikut dapat dilihat pengujian dari tambah pengiriman pada tabel 4.22 dibawah ini:

Tabel 4.22 Pengujian tambah jenis pengiriman

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik baru	Menampilkan *form* tambah jenis pengiriman	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data pengiriman Contoh: Provinsi : Jawa Barat Kota : Bandung harga: 1000	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
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] 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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.18 Pengujian data provinsi

Berikut dapat dilihat pengujian dari data provinsi pada tabel 4.23 dibawah ini:

Tabel 4.23 Pengujian data provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik provinsi	Menampilkan data provinsi	Dapat menampilkan data provinsi	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah provinsi	Dapat menampilkan *form* ubah provinsi	[√] Diterima [ ] Ditolak
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	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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.19 Pengujian tambah provinsi

Berikut dapat dilihat pengujian dari tambah provinsi pada tabel 4.24 dibawah ini:

Tabel 4.24 Pengujian tambah provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	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 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
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] 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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.20 Pengujian data kota

Berikut dapat dilihat pengujian dari data kota pada tabel 4.25 dibawah ini:

Tabel 4.25 Pengujian lihat kota

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik kota	Menampilkan data kota	Dapat menampilkan data kota	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah kota	Dapat menampilkan *form* ubah kota	[√] Diterima [ ] Ditolak
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	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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.21 Pengujian tambah kota

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

Tabel 4.26 Pengujian tambah kota

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik baru	Menampilkan *form* tambah kota	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota : Subang	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
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] 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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.22 Pengujian data kategori

Berikut dapat dilihat pengujian dari lihat kategori pada tabel 4.27 dibawah ini:

Tabel 4.27 Pengujian lihat kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik kategori	Menampilkan *form* lihat kategori	Dapat menampilkan *form* lihat kategori	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah kategori	Dapat menampilkan *form* ubah kategori	[√] Diterima [ ] Ditolak
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	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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.23 Pengujian tambah kategori

Berikut dapat dilihat pengujian dari tambah kategori pada tabel 4.28 dibawah ini:

Tabel 4.28 Pengujian tambah kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik baru	Menampilkan *form* tambah kategori	Dapat menampilkan *form* tambah kategori	[√] Diterima [ ] Ditolak
Mengisi data kategori Contoh: Nama kategori: memory	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
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] 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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.24 Pengujian data barang

Berikut dapat dilihat pengujian dari data barang pada tabel 4.29 dibawah ini:

Tabel 4.29 Pengujian lihat barang

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik barang	Menampilkan data barang	Dapat menampilkan data barang	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah barang	Dapat menampilkan *form* ubah barang	[√] Diterima [ ] Ditolak
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	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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.25 Pengujian tambah produk

Berikut dapat dilihat pengujian dari tambah barang pada tabel 4.30 dibawah ini:

Tabel 4.30 Pengujian tambah barang

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik baru	Menampilkan *form* tambah barang	Dapat menampilkan *form* tambah barang	[√] Diterima [ ] Ditolak
Mengisi data produk Contoh: kategori: Wanita Nama : Gaun Sania WA Motif : Batik Solo Jenis: Batik Tulis Ukuran : M Jumlah : 10 Berat : 0,5 Harga : 70000 Gambar : Gaun Sania WA.jpg	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
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[√] 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 batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.5.2.26 Pengujian data member

Berikut dapat dilihat pengujian dari data member pada tabel 4.31 dibawah ini:

Tabel 4.31 Pengujian lihat member

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	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

4.5.2.27 Pengujian data transaksi

Berikut dapat dilihat pengujian dari lihat transaksi pada tabel 4.32 dibawah ini:

Tabel 4.32 Pengujian lihat transaksi

Kasus dan Hasil Uji Benar (Data Benar)			
Data yang dimasukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik transaksi	Menampilkan *form* transaksi	Dapat menampilkan *form* lihat transaksi	[√] Diterima [ ] Ditolak
Klik tabel transaksi pemesanan baru	Menampilkan *form* transaksi pemesanan baru	Dapat menampilkan *form* transaksi pemesanan baru	[√] Diterima [ ] Ditolak
Klik tabel transaksi	Menampilkan *form*	Dapat	[√] Diterima

pembayaran	transaksi pembayaran	menampilkan *form* transaksi pembayaran	[ ] Ditolak
Klik tabel transaksi pengiriman	Menampilkan *form* transaksi pengiriman	Dapat menampilkan *form* transaksi pengiriman	[√] Diterima [ ] Ditolak
Klik tabel transaksi berhasil	Menampilkan *form* transaksi berhasil	Dapat menampilkan *form* transaksi berhasil	[√] Diterima [ ] Ditolak
Klik tabel transaksi gagal	Menampilkan *form* transaksi gagal	Dapat menampilkan *form* transaksi gagal	[√] Diterima [ ] Ditolak
Klik tombol pembayaran pada tab transaksi pemesanan baru	Ubah status transaksi pada tabel transaksi di *database* menjadi telah dibayar	Data status transaksi telah diubah di *database* menjadi telah dibayar	[√] Diterima [ ] Ditolak
Klik tombol detail pada tab transaksi pemesanan baru	Menampilkan *form* detail transaksi	Dapat menampilkan *form* detail transaksi	[√] Diterima [ ] Ditolak
Klik tombol pengiriman pada tab transaksi pembayaran	Ubah status transaksi pada tabel transaksi di *database* menjadi telah dikirim	Data status transaksi telah diubah di *database* menjadi telah dikirim	[√] Diterima [ ] Ditolak
Klik tombol detail pada tab transaksi pembayaran	Menampilkan *form* detail transaksi	Dapat menampilkan *form* detail transaksi	[√] Diterima [ ] Ditolak
Klik tombol berhasil pata tab transaksi pengiriman	Ubah status transaksi pada tabel transaksi di *database* menjadi telah terkirim	Data status transaksi telah diubah di *database* menjadi telah terkirim	[√] Diterima [ ] Ditolak
Klik tombol detail pata tab transaksi pengiriman	Menampilkan *form* detail transaksi	Dapat menampilkan *form* detail transaksi	[√] Diterima [ ] Ditolak

4.5.3 Kesimpulan Pengujian Alpha

Berdasarkan hasi pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan *E-commerce* pada toko Mega Batik bebas kesalahan penulisan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.6 Pengujian Beta

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 petugas Mega Batik

   Penelitian di lingkungan petugas Mega Batik dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

2. Lingkungan Masyarakat Umum

   Penelitian di lingkungan pada masyarakat umum khususnya yang sudah terbiasa dengan teknologi *internet*, dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat memberikan kemudahan masyarakat umum untuk melakukan transaksi pembelian barang.

Adapun beberapa poin yang diteliti adalah :

a. Tampilan antar muka yang *user friendly*
b. *Website* dapat memenuhi kebutuhan informasi pelanggan
c. Memudahkan pelanggan dalam melakukan transaksi pembelian barang

d. Pendokumentasikan data *master* dan data transaksi penjualan akan lebih teratur bagi Mega Batik

e. Sistem pembayaran yang mudah digunakan.

f. *Website* menjadi sarana memperluas pangsa pasar Mega Batik

g. Kelayakan *website* untuk dipublikasikan

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

Keterangan :

P = Banyaknya jawaban responden tiap soal.

Q = Jumlah responden

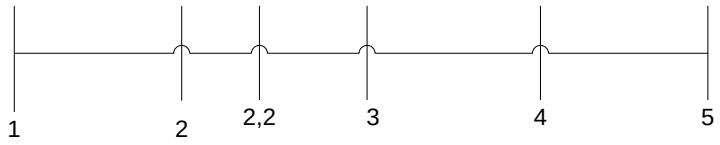
Y = Nilai persentase

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

1. Apakah anda setuju bahwa tampilan *website* cukup menarik dan mudah dipahami *user friendly*?

Tabel 4.33 Hasil pengujian kuesioner soal nomor 1

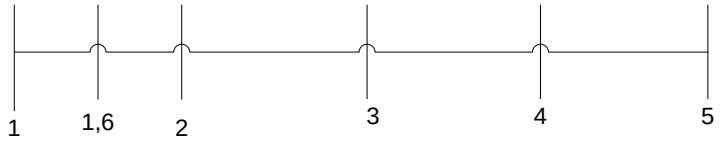
Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
1	1	Sangat Setuju	0	0
	2	Setuju	8	80
	3	Cukup Setuju	2	20
	4	Biasa Saja	0	0
	5	Tidak Setuju	0	0
Jumlah			**10**	**100**



2. Apakah anda setuju bahwa informasi barang yang diberikan oleh *website* ini sesuai dengan informasi yang anda butuhkan?

Tabel 4.34 Hasil pengujian kuesioner soal nomor 2

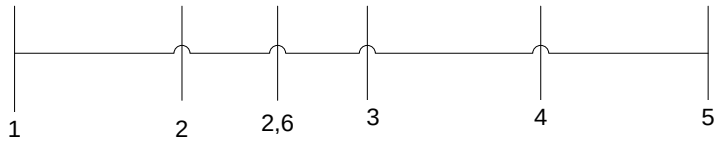
Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
2	1	Sangat Setuju	4	40
	2	Setuju	6	60
	3	Cukup Setuju	0	0
	4	Biasa Saja	0	0
	5	Tidak Setuju	0	0
Jumlah			**10**	**100**



3. Apakah anda setuju bahwa *website* ini akan lebih memudahkan pelanggan dalam melakukan transaksi pembelian barang?

Tabel 4.35 Hasil pengujian kuesioner soal nomor 3

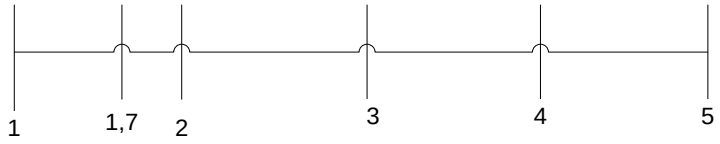
Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
3	1	Sangat Setuju	1	10
	2	Setuju	2	20
	3	Cukup Setuju	7	70
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
Jumlah			**10**	**100**



4. Apakah anda setuju bahwa dengan adanya *website* ini pihak Mega Batik akan lebih mudah mengelola data *master* dan data transaksi penjualannya?

Tabel 4.36 Hasil pengujian kuesioner soal nomor 4

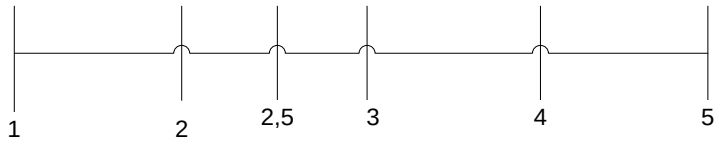
Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
4	1	Sangat Setuju	3	30
	2	Setuju	7	70
	3	Cukup Setuju	0	0
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
Jumlah			**10**	**100**



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

Tabel 4.37 Hasil pengujian kuesioner soal nomor 5

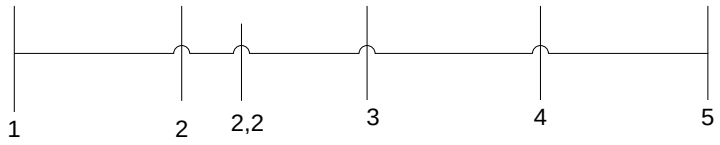
Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
5	1	Sangat Setuju	0	0
	2	Setuju	7	70
	3	Cukup Setuju	1	10
	4	Biasa Saja	2	0
	5	Kurang Setuju	0	0
Jumlah			**10**	**100**



6. Apakah anda setuju dengan adanya website Mega Batik ini dapat memperluas pangsa pasarnya?

Tabel 4.38 Hasil pengujian kuesioner soal nomor 6

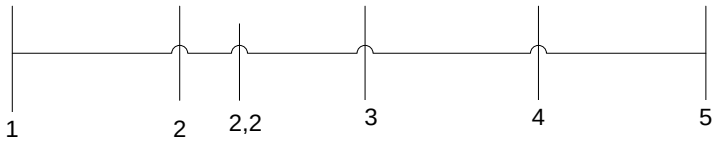
Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
6	1	Sangat Setuju	0	0
	2	Setuju	8	80
	3	Cukup Setuju	2	20
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
Jumlah			**10**	**100**



7. Apakah anda setuju bahwa *website* ini telah layak untuk di-*online*-kan?

Tabel 4.39 Hasil pengujian kuesioner soal nomor 7

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
7	1	Sangat Setuju	0	0
	2	Setuju	8	80
	3	Cukup Setuju	2	20
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
Jumlah			**10**	**100**



4.6.2 Kesimpulan Pengujian Beta

Berdasarkan pengujian beta diatas, bahwa dapat diambil kesimpulan bahwa:

1. Tampilan *website* cukup menarik dan mudah dipahami.
2. Informasi barang yang ada di *website* ini sesuain dengan yang konsumen butuhkan.
3. Proses pembayaran yang digunakan cukup mudah digunakan bagi para pelanggan.
4. Aplikasi ini juga memudahkan pihak toko dalam mengola data master dan data transaksi penjualan barangnya.
5. Aplikasi ini juga dapat memperluas pangsa pasar Mega Batik karena menjadi tidak terbatas secara geografis.