BAB IV

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 aplikasi *E-Commerce* di Moist Clothing adalah tercantum pada tabel 4.1 sebagai berikut:

Tabel 4. 1 Perangkat keras yang digunakan

No	Perangkat Keras	Spesifikasi
1	Processor	Dual Core T4500 @2.30GHZ
2	RAM	1.00 Gb
3	Harddisk	320 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	*PhpMyAdmin*	*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

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

1. Tabel admin

```
CREATE TABLE IF NOT EXISTS `admin` (
  `ADMIN_ID` varchar(50) NOT NULL,
  `ADMIN_NAMA` varchar(50) DEFAULT NULL,
  `PASSWORD`   varchar(100)   DEFAULT   NULL,PRIMARY   KEY
(`ADMIN_ID`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Tabel detail retur

```
CREATE TABLE IF NOT EXISTS `retur` (
  `retur_detail_id` int(10) NOT NULL AUTO_INCREMENT,
  `RETUR_ID` int(11) NOT NULL,
  `produkdetail_id` int(11) NOT NULL,
  `Deskripsi` text NOT NULL,
  PRIMARY KEY (`retur_detail_id`),
  KEY `FK_detail_retur_1` (`RETUR_ID`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `detail_retur`
  ADD    CONSTRAINT    `FK_detail_retur_1`    FOREIGN    KEY
(`RETUR_ID`) REFERENCES `retur` (`RETUR_ID`);
```

3. Tabel retur

```
CREATE TABLE IF NOT EXISTS `retur` (
  `RETUR_ID` int(10) NOT NULL AUTO_INCREMENT,
  `TRANSAKSI_ID` int(11) DEFAULT NULL,
  `TGL_RETUR` datetime DEFAULT NULL,
  `TGL_PROSES` datetime NOT NULL,
  PRIMARY KEY (`RETUR_ID`))
ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `retur`
  ADD CONSTRAINT `retur_ibfk_1` FOREIGN KEY (`TRANSAKSI_ID`)
REFERENCES `transaksi` (`TRANSAKSI_ID`);
```

4. Tabel gambar

```
CREATE TABLE IF NOT EXISTS `gambar` (
  `GAMBAR_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PRODUK_ID` int(11) DEFAULT NULL,
  `GAMBAR` varchar(100) DEFAULT NULL,
  `GAMBAR_NAMA` varchar(50) DEFAULT NULL,
  `DIHAPUS` enum('T','Y') NOT NULL DEFAULT 'T',
  PRIMARY KEY (`GAMBAR_ID`),
  KEY `FK_RELATIONSHIP_18` (`PRODUK_ID`)
)ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `gambar`
  ADD CONSTRAINT `gambar_ibfk_1` FOREIGN KEY (`PRODUK_ID`)
REFERENCES `produk` (`PRODUK_ID`);
```

5. Tabel jasa pengiriman

```
CREATE TABLE IF NOT EXISTS `jasa_pengiriman` (
  `JASAPNGIRIMAN_ID` varchar(11) NOT NULL,
  `ADMIN_ID` varchar(50) DEFAULT NULL,
  `JASA_NAMA` varchar(20) DEFAULT NULL,
```

```
  `DIHAPUS` enum('T','Y') NOT NULL DEFAULT 'T',
  `lama_kirim` int(11) NOT NULL,
  PRIMARY KEY (`JASAPNGIRIMAN_ID`),
  UNIQUE KEY `namaproduk_uk2` (`JASA_NAMA`),
  KEY   `FK_RELATIONSHIP_20`   (`ADMIN_ID`))   ENGINE=InnoDB
DEFAULT CHARSET=latin1;
ALTER TABLE `jasa_pengiriman`
  ADD   CONSTRAINT   `jasa_pengiriman_ibfk_1`   FOREIGN   KEY
(`ADMIN_ID`) REFERENCES `admin` (`ADMIN_ID`);
```

6. Tabel kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `KATEGORI_KODE` int(11) NOT NULL AUTO_INCREMENT,
  `KATEGORI_NAMA` varchar(50) DEFAULT NULL,
  `DIHAPUS` enum('T','Y') DEFAULT 'T',
  `admin_id` varchar(50) NOT NULL,
  PRIMARY KEY (`KATEGORI_KODE`),
  UNIQUE KEY `namaproduk_uk1` (`KATEGORI_NAMA`),
  KEY `admin_idfk2` (`admin_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `kategori`
  ADD CONSTRAINT `kategori_ibfk_1` FOREIGN KEY (`admin_id`)
REFERENCES `admin` (`ADMIN_ID`);
```

7. Tabel kota

```
CREATE TABLE IF NOT EXISTS `kota` (
  `KOTA_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PROVINSI_ID` int(11) NOT NULL,
  `KOTA_NAMA` varchar(50) NOT NULL,
  `DIHAPUS` enum('T','Y') NOT NULL DEFAULT 'T',
  PRIMARY KEY (`KOTA_ID`),
  KEY `FK_RELATIONSHIP_11` (`PROVINSI_ID`)
)  ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `kota`
  ADD  CONSTRAINT  `kota_fk_1`  FOREIGN  KEY  (`provinsi_id`)
REFERENCES `provinsi` (`PROVINSI_ID`);
```

8. Tabel member

```
CREATE TABLE IF NOT EXISTS `member` (
  `KOTA_ID` int(11) DEFAULT NULL,
  `NAMA_DEPAN` varchar(50) DEFAULT NULL,
  `NAMA_BELAKANG` varchar(50) DEFAULT NULL,
  `ALAMAT_MEMBER` text,
  `TLP_MEMBER` varchar(30) DEFAULT NULL,
  `EMAIL_MEMBER` varchar(100) NOT NULL DEFAULT '',
  `KODEPOS` varchar(30) DEFAULT NULL,
  `VERIFIMEMBER` varchar(32) DEFAULT NULL,
```

```
  `PASS_MEMBER` varchar(100) DEFAULT NULL,
  `STATUS_MEMBER` enum('T','Y') NOT NULL DEFAULT 'T',
  PRIMARY KEY (`EMAIL_MEMBER`),
  KEY `FK_RELATIONSHIP_10` (`KOTA_ID`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `member`
  ADD CONSTRAINT `member_ibfk_1` FOREIGN KEY (`KOTA_ID`)
REFERENCES `kota` (`KOTA_ID`);
```

10. Tabel produk

```
CREATE TABLE IF NOT EXISTS `produk` (
  `PRODUK_ID` int(11) NOT NULL AUTO_INCREMENT,
  `KATEGORI_KODE` int(11) NOT NULL,
  `PRODUK_NAMA` varchar(50) DEFAULT NULL,
  `PRODUK_HRG` decimal(11,0) DEFAULT NULL,
  `PRODUK_DES` text,
  `PRODUK_BRT` float DEFAULT NULL,
  `admin_id` varchar(50) NOT NULL,
  `DIHAPUS` enum('T','Y') NOT NULL DEFAULT 'T',
  PRIMARY KEY (`PRODUK_ID`),
  UNIQUE KEY `namaproduk_uk` (`PRODUK_NAMA`),
  KEY `fk_admin_id2` (`admin_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `produk`
  ADD CONSTRAINT `produk_ibfk_1` FOREIGN KEY (`admin_id`)
REFERENCES `admin` (`ADMIN_ID`),
ADD CONSTRAINT `kategori_ibfk_2` FOREIGN KEY (`KATEGORI
_KODE`) REFERENCES `kategori` (`KATEGORI_KODE`);
```

11. Tabel produk_detail

```
CREATE TABLE IF NOT EXISTS `produk_detail` (
  `produkdetail_id` int(50) NOT NULL AUTO_INCREMENT,
  `UKURAN_KODE` varchar(11) NOT NULL,
  `PRODUK_ID` int(11) NOT NULL,
  `produk_diskon` float NOT NULL DEFAULT '0',
  `STOK` int(11) NOT NULL DEFAULT '0',
  `TANGGAL` date DEFAULT NULL,
  `DIHAPUS` enum('T','Y') NOT NULL DEFAULT 'T',
  `tgl_update` datetime NOT NULL,
  PRIMARY KEY (`produkdetail_id`),
  KEY `FK_RELATIONSHIP_2` (`PRODUK_ID`),
  KEY `fk_UKURAN_KODE` (`UKURAN_KODE`)
)  ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `produk_detail`
  ADD   CONSTRAINT   `produk_detail_ibfk_1`   FOREIGN   KEY
(`PRODUK_ID`) REFERENCES `produk` (`PRODUK_ID`),
  ADD   CONSTRAINT   `produk_detail_ibfk_2`   FOREIGN   KEY
(`UKURAN_KODE`) REFERENCES `ukuran` (`UKURAN_KODE`);
```

12. Tabel provinsi

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `PROVINSI_ID` int(11) NOT NULL AUTO_INCREMENT,
  `PROVINSI_NAMA` varchar(50) DEFAULT NULL,
  `DIHAPUS` enum('T','Y') DEFAULT 'T',
  PRIMARY KEY (`PROVINSI_ID`),
  UNIQUE KEY `namaproduk_uk3` (`PROVINSI_NAMA`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

13. Tabel shipping

```
CREATE TABLE IF NOT EXISTS `shipping` (
  `JASAPNGIRIMAN_ID` varchar(11) NOT NULL,
  `KOTA_ID` int(11) NOT NULL,
  `ID_ONGKIR` int(30) NOT NULL AUTO_INCREMENT,
  `ONGKIR` decimal(11,0) DEFAULT NULL,
  PRIMARY KEY (`ID_ONGKIR`),
  KEY `FK_RELATIONSHIP_16` (`KOTA_ID`),
  KEY `jp_id_fk` (`JASAPNGIRIMAN_ID`)
)ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `shipping`
  ADD    CONSTRAINT    `shipping_ibfk_1`    FOREIGN    KEY
(`JASAPNGIRIMAN_ID`)      REFERENCES      `jasa_pengiriman`
(`JASAPNGIRIMAN_ID`),
  ADD CONSTRAINT `shipping_ibfk_2` FOREIGN KEY (`KOTA_ID`)
REFERENCES `kota` (`KOTA_ID`);
```

14. Tabel testimonial

```
CREATE TABLE IF NOT EXISTS `testimonial` (
  `TES_ID` int(11) NOT NULL AUTO_INCREMENT,
  `email_member` varchar(50) DEFAULT NULL,
  `ADMIN_ID` varchar(50) DEFAULT NULL,
  `KOMENTAR` text,
  `STATUS` enum('T','Y') DEFAULT 'T',
  PRIMARY KEY (`TES_ID`),
  KEY `FK_RELATIONSHIP_12` (`email_member`),
  KEY   `FK_RELATIONSHIP_17`   (`ADMIN_ID`))   ENGINE=InnoDB
DEFAULT CHARSET=latin1;
ALTER TABLE `testimonial`
  ADD    CONSTRAINT    `testimonial_ibfk_1`    FOREIGN    KEY
(`ADMIN_ID`) REFERENCES `admin` (`ADMIN_ID`),
  ADD    CONSTRAINT    `testimonial_ibfk_2`    FOREIGN    KEY
(`email_member`) REFERENCES `member` (`EMAIL_MEMBER`);
```

15. Tabel transaksi

```
CREATE TABLE IF NOT EXISTS `transaksi` (
  `TRANSAKSI_ID` int(11) NOT NULL AUTO_INCREMENT,
  `email_member` varchar(100) NOT NULL,
  `id_ongkir` int(30) NOT NULL,
  `TGL_PESAN` datetime DEFAULT NULL,
  `TGL_BAYAR` datetime DEFAULT NULL,
  `TGL_KIRIM` date DEFAULT NULL,
  `TGL_TERIMA` datetime DEFAULT NULL,
  `TGL_EXP` datetime DEFAULT NULL,
  `TOTALBAYAR` decimal(11,0) DEFAULT NULL,
  `BERAT_TOTAL` float DEFAULT NULL,
  `NAMA_PENERIMA` varchar(30) DEFAULT NULL,
  `ALAMAT_KIRIM` text,
  `STATUS_PENGIRIMAN` varchar(20) DEFAULT NULL,
  `JNIS_PMBYARAN` varchar(20) DEFAULT NULL,
  `NAMA_BANK` varchar(50) DEFAULT NULL,
  `NO_REKENING` varchar(20) DEFAULT NULL,
  `NO_RESI` varchar(20) DEFAULT NULL,
  `jmlhbyar` decimal(11,0) DEFAULT NULL,
  `an` varchar(100) DEFAULT NULL,
  PRIMARY KEY (`TRANSAKSI_ID`),
  KEY `fk_mail_member` (`email_member`),
  KEY `fk_id_ongkir2` (`id_ongkir`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `transaksi`
  ADD    CONSTRAINT    `transaksi_ibfk_1`    FOREIGN    KEY
(`email_member`) REFERENCES `member` (`EMAIL_MEMBER`),
  ADD    CONSTRAINT    `transaksi_ibfk_2`    FOREIGN    KEY
(`id_ongkir`) REFERENCES `shipping` (`ID_ONGKIR`);
```

16. Tabel transaksi_detail

```
CREATE TABLE IF NOT EXISTS `transaksi_detail` (
  `transaksidetail_id` int(20) NOT NULL AUTO_INCREMENT,
  `TRANSAKSI_ID` int(11) NOT NULL,
  `produkdetail_id` int(20) NOT NULL,
  `QTY` int(11) DEFAULT NULL,
  `HARGA` decimal(11,0) DEFAULT NULL,
  `produk_diskon` float NOT NULL,
  PRIMARY KEY (`transaksidetail_id`),
  KEY `FK_RELATIONSHIP_5` (`TRANSAKSI_ID`),
  KEY `fk_produkdetail_id` (`produkdetail_id`)
)ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `transaksi_detail`
  ADD   CONSTRAINT   `transaksi_detail_ibfk_1`   FOREIGN   KEY
(`TRANSAKSI_ID`) REFERENCES `transaksi` (`TRANSAKSI_ID`),
  ADD   CONSTRAINT   `transaksi_detail_ibfk_2`   FOREIGN   KEY
(`produkdetail_id`)        REFERENCES        `produk_detail`
(`produkdetail_id`);
```

17. Tabel ukuran

```
CREATE TABLE IF NOT EXISTS `ukuran` (
  `UKURAN_KODE` varchar(5) NOT NULL,
  `UKURAN_NAMA` varchar(10) DEFAULT NULL,
  `DIHAPUS` enum('T','Y') DEFAULT 'T',
  `admin_id` varchar(50) NOT NULL,
  PRIMARY KEY (`UKURAN_KODE`),
  KEY `admin_idfk2` (`admin_id`)
)ENGINE=InnoDB DEFAULT CHARSET=latin1;
ALTER TABLE `ukuran`
  ADD CONSTRAINT `ukuran_ibfk_1` FOREIGN KEY (`admin_id`)
REFERENCES `admin` (`ADMIN_ID`);
```

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. Adapun untuk tampilan implementasi pembangunan aplikasi *E-commerce* di Moist Clothing berbasis *Web* akan dilampirkan pada halaman lampiran dan Implementasi antarmuka Pengunjung dan Member dijelaskan pada tabel 4.3.

Tabel 4. 3 Implementasi antarmuka Pengunjung dan Member

Menu	Deskripsi	Nama File
Registrasi	Digunakan untuk melakukan pendaftaran *member* baru.	Member_daftar.php
Login *member*	Digunakan Digunakan sebagai halaman login *member*	lib_func.php
Lupa *password*	Digunakan jika *member* lupa dengan *password*	lupapass_form. php
Profile	Digunakan untuk melihat data profil perusahaan	profile.php

Menu	Deskripsi	Nama File
Cara Beli	Digunakan untuk melihat cara pembelian produk	cara.php
Ubah Profil	Digunakan untuk mengubah profil *member*.	ubah_profil.php
Ubah password	Digunakan jika *member* ingin merubah *password*	ubah_pass.php
Kategori	Digunakan untuk melihat barang perkategori	kategori. php
Detail Produk	Digunakan untuk melihat, data produk secara detail	produk_detail.php
Testimonial	Digunakan untuk memberikan komentar	testimonial. php
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus data pembelian yang akan dilakukan.	Keranjang-belanja.php
History	Digunakan untuk melihat *history* pembelian yang pernah dilakukan.	history.php

Tabel 4. 4 Implementasi Antarmuka Admin

Menu	Deskripsi	Nama File
Login admin	Digunakan sebagai halaman login administrator	index.php
Lupa Password	Digunakan jika admin lupa terhadap passwordnya	Lib_func.php
Kota	Digunakan untuk melihat, menambah, dan merubah data kota	Kota_view. php
Provinsi	Digunakan untuk melihat, menambah, dan merubah data provinsi	Provinsi_view.php

Menu	Deskripsi	Nama File
Jasa Pengiriman	Digunakan untuk melihat, menambah, dan merubah data jasa pengiriman	Jasa_view.php
Shipping	Digunakan untuk melihat, menambah, dan merubah data shipping	shipping_view.php
Kategori	Digunakan untuk melihat, menambah, dan merubah data kategori	Kategori_view.php
Produk	Digunakan untuk melihat, menambah, dan merubah data produk	Produk_view.php
Ukuran	Digunakan untuk melihat, menambah, dan merubah data Ukuran	Ukuran_view.php
Gambar	Digunakan untuk melihat, menambah, dan merubah data gambar	Gambar_view.php
Produk detail	Digunakan untuk melihat, menambah, dan merubah data detail produk	Produkdetail_form_tambah.php
Data Transaksi	Digunakan untuk melihat data transaksi yang masuk dan meubah statusnya.	Transaksi_view. php
Ongkos Kirim	Digunakan untuk melihat data onkos kirim, merubah, menghapus onkos kirim	Shipping_tambah. Php
Laporan	Digunakan untuk melihat data laporan harian,bulanan kemudian	laporan.php

Menu	Deskripsi	Nama File
	dicetak ke pdf	
Testimonial	Digunakan untuk melihat data testimonial yang akan dibaca.	testimonial.php
Logout	Digunakan administrator 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.

4.4.1 Pengujian Alpa

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

4.4.1.1 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*

Kelas Uji	Butir Uji	Jenis Pengujian
	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 barang	*Black Box*
	Menampilkan data barang	*Black Box*
	Cari barang	*Black Box*
Menu Profil	Menampilkan profil perusahaan	*Black Box*
Menu testimonial	Menampilkan form	*Black Box*
	Isi form Testimonial	*Black Box*
Menu keranjang belanja	Menampilkan data pembelian	*Black Box*
	Mengupdate data pembelian	*Black Box*
	Menghapus data pembelian	*Black Box*
Form konfirmasi	Menampilkan konfirmasi transaksi	*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 administrator.

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 Transaksi	Tampilkan data transaksi	*Black Box*
	Ubah status transaksi	*Black Box*
Kategori	Tampilkan data kategori	*Black Box*
	Tambah Data katgori	*Black Box*
	Ubah data 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*
Ukuran	Tampilkan data Ukuran	*Black Box*
	Tambah data Ukuran	*Black Box*

Kelas Uji	Butir Uji	Jenis Pengujian
	Ubah data ukuran	*Black Box*
Data Laporan	Tampilkan data laporan	*Black Box*
	Cari data laporan	*Black Box*
	Cetak data laporan	*Black Box*
Data Transaksi	Tampilkan data Transaksi	*Black Box*
	Ubah data transaksi	*Black Box*
Data kategori	Tampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
Detail Produk	Tampilkan detail barang	*Black Box*
	Tambah detail barang	*Black Box*
	Ubah detail barang	*Black Box*
Data Produk	Tampilkan data barang	*Black Box*
	Tambah data barang	*Black Box*
	Ubah data barang	*Black Box*
Logout	Menghapus *session user*	*Black Box*

4.4.2 Kasus dan Hasil Pengujian

4.4.2.1 Contoh Uji Registrasi Member

Berikut dapat dilihat pengujian dari registrasi *member* pada tabel 4.7.

Tabel 4. 7 Pengujian Registrasi *member*

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data, contoh masukan: Email : yemima_cutez@yahoo.co.id Nama depan : Yemima Nama belakang : Tri Heryani Alamat : Jl. Cikajang 14 no.71 Provinsi : Jawa Barat Kota : Bandung Kodepos : 42314 Telepon : 79050953 Password : 123456 Ulangi password : 123456	Jika data masukan benar muncul pesan “Terima Kasih Anda Telah Berhasil Registrasi. Silahkan cek email untuk melakukan Aktivasi”	Semua diisi dengan benar, maka sesuai yang di harapkan “Terima Kasih Anda Telah Berhasil Registrasi. Silahkan cek email untuk melakukan Aktivasi”	[√] Diterima [ ] Ditolak
Masukan data, contoh masukan: Email : yemima_cutez@yahoo.co.id Nama depan : Nama belakang : Tri Heryani Alamat : Jl. Cikajang 14 no.71 Provinsi : Jawa Barat Kota : Bandung Kodepos : 42314 Telepon : 79050953 Password : 123456	Jika nama depan tidak diisi maka muncul pesan “Nama depan harus di isi”	Jika nama depan tidak diisi maka sesuai yang di harapkan muncul pesan “Nama depan harus di isi”	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Ulangi password : 123456			
Masukan data, contoh masukan: Email : Nama depan : Yemima Nama belakang : Tri Heryani Alamat : Jl. Cikajang 14 no.71 Provinsi : Jawa Barat Kota : Bandung Kodepos : 42314 Telepon : 79050953 Password : 123456 Ulangi password : 123456	Jika email tidak diisi maka muncul pesan “Email harus diisi”	Jika email tidak diisi maka muncul pesan “Email harus diisi”	[√] Diterima [ ] Ditolak
Masukan data, contoh masukan: Email : yemima_cutez@yahoo.co.id Nama depan : Yemima Nama belakang : Alamat : Jl. Cikajang 14 no.71 Provinsi : Jawa Barat Kota : Bandung Kodepos : 42314 Telepon : 79050953 Password : 123456 Ulangi password : 123456	Jika nama belakang tidak diisi, maka tampilkan pesan kesalahan”Nama belakang harus diisi”	Jika nama belakang tidak diisi, maka tampilkan pesan kesalahan”Nam a belakang harus diisi”	[√] Diterima [ ] Ditolak
Masukan data, contoh masukan:	Jika alamat tidak diisi, maka	Jika alamat tidak diisi, maka	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Email : yemima_cutez@yahoo.co.id Nama depan : Yemima Nama belakang : Tri Heryani Alamat : Provinsi : Jawa Barat Kota : Bandung Kodepos : 42314 Telepon : 79050953 Password : 123456 Ulangi password : 123456	tampilkan pesan kesalahan”Alamat harus diisi”	tampilkan pesan kesalahan”Alamat harus diisi”	
Masukan data, contoh masukan: Email : yemima_cutez@yahoo.co.id Nama depan : Yemima Nama belakang : Tri Heryani Alamat : Jl. Cikajang 14 no.71 Provinsi : Kota : Bandung Kodepos : 42314 Telepon : 79050953 Password : 123456 Ulangi password : 123456	Jika data provinsi tidak diisi, maka tampilkan pesan kesalahan ”Provinsi harus diisi”	Jika data provinsi tidak diisi, maka tampilkan pesan kesalahan ”Provinsi harus diisi”	[√] Diterima [ ] Ditolak
Masukan data, contoh masukan: Email : yemima_cutez@yahoo.co.id	Jika data kota tidak diisi maka tampilkan pesan kesalahan ”kota	Jika data kota tidak diisi maka tampilkan pesan kesalahan ”kota	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama depan : Yemima Nama belakang : Tri Heryani Alamat : Jl. Cikajang 14 no.71 Provinsi : Jawa Barat Kota : Kodepos : 42314 Telepon : 79050953 Password : 123456 Ulangi password : 123456	harus diisi”	harus diisi”	
Masukan data, contoh masukan: Email : yemima_cutez@yahoo.co.id Nama depan : Yemima Nama belakang : Tri Heryani Alamat : Jl. Cikajang 14 no.71 Provinsi : Jawa Barat Kota : Bandung Kodepos : Telepon : 79050953 Password : 123456 Ulangi password : 123456	Jika data kodepos tidak diisi maka tampilkan pesan kesalahan ”kode pos harus diisi”	Jika data kodepos tidak diisi maka tampilkan pesan kesalahan ”kode pos harus diisi”	[√] Diterima [ ] Ditolak
Masukan data, contoh masukan: Email : yemima_cutez@yahoo.co.id Nama depan : Yemima Nama belakang : Tri Heryani	Jika data password tidak diisi maka tampilkan pesan kesalahan ”password harus diisi”	Jika data password tidak diisi maka tampilkan pesan kesalahan ”password harus	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Alamat : Jl. Cikajang 14 no.71 Provinsi : Jawa Barat Kota : Bandung Kodepos : 42314 Telepon : 79050953 Password : Ulangi password : 123456		diisi”	

4.4.2.2 Contoh Uji Login Member

Berikut dapat dilihat pengujian login *member* pada tabel 4.8.

Tabel 4. 8 Pengujian login *member*

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan alamat *email* : yemima_cutez1@yahoo.co.id	Jika *email* salah maka muncul pesan “Email dan password yang anda masukkan salah, silahkan ulangi”	Jika *email* salah maka muncul pesan “Email dan password yang anda masukkan salah, silahkan ulangi”	[√] Diterima [ ] Ditolak
Memasukan *email* : yemima_cutez@yahoo.co.id	Jika *email* salah maka muncul pesan “Email dan password yang anda masukkan salah, silahkan ulangi”	Jika *email* salah maka muncul pesan “Email dan password yang anda masukkan salah, silahkan ulangi”	[√] Diterima [ ] Ditolak
Memasukan *email* : yemima_cutez@yahoo.co.id *Password*: 12390292	Jika *email* salah maka muncul pesan “Email dan password yang anda masukkan salah, silahkan	Jika *email* salah maka muncul pesan “Email dan password yang anda masukkan salah,	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
	ulangi"	silahkan ulangi"	
Memasukan *email* : yemima_cutez@yahoo.co.id *Password*: 123456	Jika *email* dan *password* benar maka masuk ke halaman member	Jika *email* dan *password* benar maka masuk ke halaman member	[√] Diterima [ ] Ditolak

4.4.2.3 Contoh Uji Ganti *Password* Member

Berikut adalah pengujian dari ganti *password* pada member pada tabel 4.9:

Tabel 4. 9 Pengujian ganti password member

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Password Lama: Password Baru : Ulangi password baru :	Muncul pesan "Anda harus mengisikan semua data pada form Ganti Password."	Muncul pesan "Anda harus mengisikan semua data pada form Ganti Password."	[√] Diterima [ ] Ditolak
Password Lama: 1331412 Password Baru : Ulangi password baru :	Muncul pesan "Anda harus mengisikan semua data pada form Ganti Password."	Muncul pesan "Anda harus mengisikan semua data pada form Ganti Password."	[√] Diterima [ ] Ditolak
Password Lama: 123456 Password Baru : 12345	Jika semua data diisi dengan benar maka muncul pesan "Password baru telah disimpan"	Jika semua data diisi dengan benar maka muncul pesan "Password baru telah	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Ulangi password baru :12345		disimpan”	

4.4.2.4 Contoh Uji Keranjang Belanja

Berikut dapat dilihat pengujian dari menu keranjang belanja pada tabel 4.10

Tabel 4. 10 Pengujian Menu Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Info data, Nama produk : Celana olah raga Berat : 0 kg Ukuran : S Diskon : 10% Stok : 18	Menampilkan *form* keranjang belanja	menampilkan *form* keranjang belanja	[√] Diterima [ ] Ditolak
Data masukkan, Qty : 19	Setelah menekan update maka akan menampilkan pesan “Jumlah yang dibeli melebihi stok yang ada”	Setelah menekan update maka akan menampilkan pesan “Jumlah yang dibeli melebihi stok yang ada”	[√] Diterima [ ] Ditolak
Data masukkan, Qty : 1	Setelah menekan selesai maka akan menampilkan form data tujuan pengiriman	Setelah menekan selesai maka akan menampilkan form data tujuan pengiriman	[√] Diterima [ ] Ditolak
Data masukkan, Qty : -3	Setelah menekan update maka akan menampilkan pesan “Jumlah yang dibeli harus lebih dari 0”	Setelah menekan update maka akan menampilkan pesan “Jumlah yang dibeli harus lebih dari 0”	[√] Diterima [ ] Ditolak

4.4.2.5 Contoh Uji Form Pengiriman

Berikut dapat dilihat pengujian dari pengiriman pada tabel 4.11.

Tabel 4. 11 Pengujian form Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : Alamat lengkap: jl.sedayu 152 Telpon/Hp: 089920219 Email : yemima_cutez@yahoo.co.id Kodepos : 43892 Provinsi: jawa barat Kota : Kota Bandug Paket pengiriman : JNE REG Biaya : 4000 Lama pengiriman : 3	Jika data nama belum diisi maka muncul pesan "Nama tidak boleh kosong"	Jika data nama belum diisi maka muncul pesan "Nama tidak boleh kosong"	[√] Diterima [ ] Ditolak
Nama : yemima Alamat lengkap: jl.sedayu152 Telpon/Hp: Email : yemima_cutez@yahoo.co.id Kodepos : 43892 Provinsi: jawa barat Kota : Kota Bandug Paket pengiriman : JNE REG Biaya : 4000 Lama pengiriman : 3	Jika data Telpon/Hp belum diisi maka muncul pesan "No Telepon tidak boleh kosong"	Jika data nama belum diisi maka muncul pesan "No Telepon tidak boleh kosong"	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : yemima Alamat lengkap: jl.sedayu152 Telpon/Hp: 089920219 Email : yemima_cutez@yahoo.co.id Kodepos : 43892 Provinsi: Kota : Paket pengiriman : Biaya : Lama pengiriman :	Jika data provinsi belum diisi maka muncul pesan “data provinsi tidak boleh kosong”	Jika data provinsi belum diisi maka muncul pesan “data provinsi tidak boleh kosong”	[√] Diterima [ ] Ditolak

4.4.2.6 Pengujian Form Konfirmasi Pembayaran

Berikut dapat dilihat pengujian dari *Form* konfirmasi pembayaran pada tabel 4.12.

Tabel 4. 12 Pengujian *Form* konfirmasi pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Info data konfirmasi pembayaran, contoh No.transaksi : 97 Tgl pesan : 14-06-2011 01:31:31 Bank asal : BCA Jumlah yg dibayar : 88000	Jika data konfirmasi pembayaran valid maka tampilkan pesan kesalahan ”konfirmasi pembayaran telah berhasil dikirim, tunggu konfirmasi”	Jika data konfirmasi pembayaran valid maka tampilkan pesan kesalahan ”konfirmasi pembayaran telah berhasil dikirim, tunggu konfirmasi”	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Masukan data konfirmasi pembayaran, contoh No.transaksi : 97 Tgl bayar : 12-06-2011 Atas Nama : Yemima Jumlah Transfer : 88000	Jika tgl bayar kurang dari tgl pesan maka tampilkan pesan kesalahan ”tanggal tidak valid”	Jika tgl bayar kuran dari tgl pesan maka tampilkan pesan kesalahan ”tanggal tidak valid”	[√] Diterima [ ] Ditolak
Masukan data konfirmasi pembayaran, contoh No.transaksi : 97 Tgl bayar : 15-06-2011 Atas Nama : Yemima Jumlah Transfer : 88000	Jika tgl bayar melebihi dari tgl hari ini maka tampilkan pesan kesalahan ”tanggal tidak valid”	Jika tgl bayar melebihi dari tgl hari ini maka tampilkan pesan kesalahan ”tanggal tidak valid”	[√] Diterima [ ] Ditolak
Masukan data konfirmasi pembayaran, contoh No.transaksi : 97 Tgl bayar : 14-06-2011 Atas Nama : Jumlah Transfer : 88000	Jika atas nama tidak diisi,maka tampilkan pesan kesalahan ”nama harus diisi”	Jika atas nama tidak diisi,maka tampilkan pesan kesalahan ”nama harus diisi”	[√] Diterima [ ] Ditolak
Masukan data konfirmasi pembayaran, contoh No.transaksi : 97 Tgl bayar : 14-06-2011 Atas Nama : Yemima Jumlah Transfer : 790	Jika jumlah transfer kurang dari jumlah yang dibayar, maka tampilan pesan “jumlah kurang dari jumlah yang harus dibayar”	Jika jumlah transfer kurang dari jumlah yang dibayar, maka tampilan pesan “jumlah kurang dari jumlah yang harus dibayar”	[√] Diterima [ ] Ditolak

4.4.2.7 Contoh Uji Login Administrator

Berikut dapat dilihat pengujian dari login administrator pada tabel 4.13

Tabel 4. 13 Pengujian login Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Username : mimi *Password* : 310888	Data login valid maka akan masuk ke menu admin	Data login valid maka akan masuk ke menu admin	[√] Diterima [ ] Ditolak
Username : *Password* :	Jika Username dan password kosong, maka muncul pesan "gagal login. Username atau password yang anda masukan salah,silahkan login kembali"	Jika Username dan password kosong, maka muncul pesan "gagal login. Username atau password yang anda masukan salah,silahkan login kembali"	[√] Diterima [ ] Ditolak
Username : mimi *Password* :	Jika Username dan password kosong, maka muncul pesan "gagal login. Username atau password yang anda masukan salah,silahkan login kembali"	Jika Username dan password kosong, maka muncul pesan "gagal login. Username atau password yang anda masukan salah,silahkan login kembali"	[√] Diterima [ ] Ditolak

4.4.2.8 Pengujian Tambah Kategori

Berikut dapat dilihat pengujian dari tambah kategori pada tabel 4.14

Tabel 4. 14 Pengujian Tambah kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data kategori Contoh: Nama kategori: T-shirt	Data kategori akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	Data kategori akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Mengisi data kategori Contoh: Nama kategori:	Jika nama Kategori kosong, maka akan muncul pesan”field tidak boleh kosong”	Jika nama Kategori kosong, maka akan muncul pesan”field tidak boleh kosong”	[√] Diterima [ ] Ditolak

4.4.2.9 Pengujian Tambah Data produk

Berikut adalah pengujian dari pengolahan data produk pada tabel 4.15

Tabel 4. 15 Pengujian Tambah Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data barang Contoh: kategori: Baju Nama : Baju Cewek Harga : 25000 Berat : 0.2 Deskripsi : Bagus loh	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

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data barang Contoh: kategori: Nama : Baju Cewek Harga : 25000 Berat : 0.2 Deskripsi : Bagus loh	Jika Nama Kategori kosong, maka akan muncul pesan"field tidak boleh kosong"	Jika Nama Kategori kosong, maka akan muncul pesan"field tidak boleh kosong"	[√] Diterima [ ] Ditolak
Mengisi data barang Contoh: kategori: Nama : Baju Cewek Harga : a Berat : 0.2 Deskripsi : Bagus loh	Jika harga berisi bukan angka, maka akan muncul pesan"field hanya diisi oleh angka"	Jika harga berisi bukan angka, maka akan muncul pesan"field hanya diisi oleh angka"	[√] Diterima [ ] Ditolak
Mengisi data barang Contoh: kategori: Nama : Baju Cewek Harga : Berat : 0.2 Deskripsi : Bagus loh	Jika data harga tidak diisi, maka akan muncul pesan"field tidak boleh kosong"	Jika data harga tidak diisi, maka akan muncul pesan"field tidak boleh kosong"	[√] Diterima [ ] Ditolak

4.4.2.10 Pengujian Tambah Provinsi

Berikut dapat dilihat pengujian dari tambah provinsi pada tabel 4.16

Tabel 4. 16 Pengujian Tambah Provinsi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data provinsi Contoh: Nama Provinsi :	Jika nama provinsi tidak diisi maka akan muncul pesan "field tidak boleh kosong"	Jika nama provinsi tidak diisi maka akan muncul pesan "field tidak boleh kosong"	[√] 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

4.4.2.11 Pengujian Tambah Kota

Berikut dapat dilihat pengujian tambah kota pada tabel 4.17

Tabel 4. 17 Pengujian Tambah Kota

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota : Bandung	Data kota telah tersimpan didalam *database*	Data kota telah tersimpan didalam *database*	[√] Diterima [ ] Ditolak
Mengisi data kota Contoh: Provinsi : Nama Kota :	Jika nama provinsi tidak terisi maka akan muncul pesan "field tidak boleh kosong."	Jika nama provinsi tidak terisi maka akan muncul pesan "field tidak boleh kosong."	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota :	Jika nama kota tidak terisi maka akan muncul pesan "field tidak boleh kosong."	Jika nama kota tidak terisi maka akan muncul pesan "field tidak boleh kosong."	[√] Diterima [ ] Ditolak

4.4.2.12 Pengujian Tambah Ukuran

Berikut dapat dilihat pengujian dari tambah ukuran pada tabel 4.18

Tabel 4. 18 Pengujian Tambah Ukuran

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data ukuran Contoh: Nama ukuran : S Keterangan : Small	Data ukuran akan tersimpan didalam *database*	Data ukuran akan tersimpan didalam *database*	[√] Diterima [ ] Ditolak
Mengisi data ukuran Contoh: Nama ukuran : Keterangan : Small	Jika nama ukuran field tidak terisi, maka akan muncul pesan "field tidak boleh kosong"	Jika salah satu field tidak terisi, maka akan muncul pesan "field tidak boleh kosong"	[√] Diterima [ ] Ditolak
Mengisi data ukuran Contoh: Nama ukuran : S Keterangan :	Jika nama ukuran field tidak terisi, maka akan muncul pesan "field tidak boleh kosong"	Jika salah satu field tidak terisi, maka akan muncul pesan "field tidak boleh kosong"	[√] Diterima [ ] Ditolak

4.4.2.13 Pengujian Tambah Ongkos Kirim

Berikut dapat dilihat pengujian dari lihat transaksi pada tabel 4.19.

Tabel 4. 19 Pengujian Tambah Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data ongkos kirim Contoh: Provinsi : Jawa Barat Kota : Bandung Jenis pengiriman : JNE harga: Rp.2500	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
Mengisi data ongkos kirim Contoh: Provinsi : Kota : Jenis pengiriman : harga:	Jika nama provinsi tidak terisi, maka akan muncul pesan “field tidak boleh kosong”	Jika nama provinsi tidak terisi, maka akan muncul pesan “field tidak boleh kosong”	[√] Diterima [ ] Ditolak
Mengisi data ongkos kirim Contoh: Provinsi : Jawa Barat Kota : Jenis pengiriman : harga:	Jika nama kota tidak terisi, maka akan muncul pesan “field tidak boleh kosong”	Jika nama kota tidak terisi, maka akan muncul pesan “field tidak boleh kosong”	[√] Diterima [ ] Ditolak

4.4.3 Kesimpulan Hasil Pengujian Alpa

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

4.5 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan dengan membuat kuesioner mengenai kepuasan user dengan kandungan poin syarat *user friendly* untuk selanjutnya dibagikan kepada sebagian *user* dengan mengambil *sample* sebanyak 10 orang dan 1 orang dari pihak Moist Clothing.

4.5.1 Skenario Pengujian Beta

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

1. Lingkungan Moist Clothing

   Penelitian di lingkungan petugas Moist Clothing 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.

4.5.1.1 Kuisioner Pengujian Betha Untuk Pengguna

1. Apakah anda setuju bahwa informasi produk yang diberikan atau ditawarkan oleh website ini sesuai dengan informasi yang anda butuhkan?

   A. Sangat Setuju
   B. Setuju
   C. Cukup setuju
   D. Biasa saja
   E. Tidak setuju

2. Apakah anda setuju bahwa tampilan website cukup menarik dan mudah dipahami atau user friendly?

   A. Sangat Setuju
   B. Setuju
   C. Cukup setuju
   D. Biasa saja
   E. Tidak setuju

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

   A. Sangat Setuju
   B. Setuju
   C. Cukup setuju
   D. Biasa saja
   E. Tidak setuju

4. Apakah anda setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan anda dan terjamin?

   A. Sangat Setuju
   B. Setuju
   D. Biasa saja
   E. Tidak setuju

C. Cukup setuju

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

A. Sangat Setuju
B. Setuju
C. Cukup setuju
D. Biasa saja
E. Tidak setuju

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 . Kuisioner ini diujikan kepada 10 orang.

1. Apakah anda setuju bahwa informasi produk yang diberikan atau ditawarkan oleh *website* ini sesuai dengan informasi yang anda butuhkan?

Tabel 4. 20 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

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
Jumlah			**10**	**100**

Berdasarkan hasil prosentase maka dapat disimpulkan sebanyak 8 atau 80% menyatakan setuju, dan 2 atau 20% menyatakan setuju bahwa informasi yang diberikan sesuai dengan informasi yang dibutuhkan *costumer*.

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

Tabel 4. 21 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**

Berdasarkan hasil prosentase maka dapat disimpulkan sebanyak 4 orang atau 40 % menyatakan sangat setuju, 6 orang atau 60% menyatakan cukup setuju bahwa tampilan *website* cukup menarik dan mudah untuk dipahami.

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

Tabel 4. 22 Hasil pengujian kuesioner soal nomor 3

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
3	1	Sangat Setuju	1	60
	2	Setuju	2	40
	3	Cukup Setuju	7	0
	4	Biasa Saja	0	0
	5	Tidak Setuju	0	0
Jumlah			**10**	**100**

Berdasarkan hasil prosentase maka dapat disimpulkan sebanyak 1 atau 10% menyatakan sangat setuju, 2 orang atau 20 % menyatakan setuju dan, 7 atau 70 % menyatakan cukup setuju bahwa *website* ini akan lebih memudahkan pelanggan dalam melakukan transaksi pembelian barang.

4. Apakah anda setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan anda dan terjamin?

Tabel 4. 23 Hasil pengujian kuesioner soal nomor 4

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

Berdasarkan hasil prosentase maka dapat disimpulkan sebanyak 6 orang atau 60 % menyatakan setuju, 2 orang atau 20% menyatakan cukup setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan,

dan 2 orang atau 20% menyatakan biasa saja bahwa system pembayaran yang digunakan lebih mudah.

5. Apakah anda setuju dengan adanya website ini Moist Clothing telah layak untuk di-*online*-kan?

Tabel 4. 24 Hasil pengujian kuesioner soal nomor 5

Pertanyaan	No.	Keterangan	Responden	Prosentase(%)
5	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**

Berdasarkan hasil prosentase maka dapat disimpulkan sebanyak 8 orang atau 80% menyatakan setuju dan 2 orang atau 20% menyatakan setuju bahwa *website* ini akan memperluas pangsa pasar Moist Clothing Apakah anda setuju bahwa *website* ini telah layak untuk di-*online*-kan

4.5.1.2 Kuisioner Pengujian Betha Untuk Admin

1. Menurut anda, aplikasi E-commerce ini cukup membantu anda dalam pengolahan data?

   Menurut saya, aplikasi ini cukup membantu dalam pengolahan data, karena tidak dilakukan secara manual lagi.

2. Apakah instruksi yang terdapat dalam website tersebut mudah dimengerti?

Instruksi-instruksi yang terdapat di dalam website ini mudah dimengerti, mungkin untuk kedepannya dapat ditambahkan lagi instruksi pada bagian admin.

3. Apakah website ini dapat membantu dalam pengolahan data barang?

   Untuk saat ini cukup membantu, karena selama ini pengolahan data barangnya masih menggunakan cara yang manual.

4. Apakah anda setuju bahwa dengan adanya website ini pihak Moist Clothing akan lebih mudah mengelola data master dan data transaksi penjualan?

   Saya setuju, karena sebelumnya transaksi penjualan dan pengelolaan data belum secara online, maka dengan adanya aplikasi ini sangat membatu pihak moist-clothing.

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

   Ya, sangat membantu karena pada perusahaan kami, belum ada pembuatan laporan, karena perusahaan ini belum fokus.

4.5.2 Kesimpulan Pengujian Beta

Berdasarkan pengujian beta , bahwa dapat diambil kesimpulan bahwa aplikasi ini memudahkan pelanggan dalam mendapatkan informasi barang yang tersedia pada Moist Clothing selain itu pelanggan juga dapat melakukan transaksi pembelian barang dengan lebih mudah.

Dalam Proses pembayaran yang digunakan cukup mudah digunakan bagi pelanggan dan dapat dipercaya. Aplikasi ini juga memudahkan pihak toko dalam mengolah data master, data barang dan data transaksi dari keluar masuknya

barang. Aplikasi ini juga dapat memperluas usaha pada Moist Clothing tidak terbatas pemasarannya secara geografis karena dapat digunakan dimanapun.