BAB 4
IMPLEMENTASI DAN PENGUJIAN

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

4.1 Implementasi Sistem

Implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis.

4.1.1 Implementasi Perangkat Lunak

Perangkat lunak (*Software*) yang dipasang pada sistem *e-commerce* yang digunakan untuk membangun *website* ini adalah sebagai berikut:

Tabel 4.1 Implementasi Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows 7 Professional
2	PHP	Bahasa Pemrograman
3	MySQL	Penyimpanan *Database*
4	PhpMyAdmin	*Interface Database*
5	Wamp*Server* 2.1	*Web Server*
6	Adobe Dreamweaver CS5	Kode dan Perancangan Tampilan
7	Mozilla Firefox	*Web Browser*

4.1.2 Perangkat Keras Yang Digunakan

Selain perangkat lunak (*software*) juga dibutuhkan keberadaan perangkat keras (*hardware*), kebutuhan minimum yang diperlukan agar dapat membangun sistem ini dengan spesifikasi sebagai berikut:

Tabel 4.2 Analisis Perangkat Keras

No	Perangkat Keras	Spesifikasi
1	*Processor*	Intel core i3 3.8 GHz
2	*RAM*	2048 MB

3	*Harddisk*	500 GB
4	*Mouse*	-
5	*Keyboard*	-
6	*Monitor*	14 *inch*

4.2 Implementasi Basis Data

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

Tabel 4.3 Tabel Bank

Table Bank
CREATE TABLE IF NOT EXISTS `bank` (`*id*_bank` int(11) *NOT NULL* AUTO_INCREMENT, `nama_bank` varchar(45) *NOT NULL*, *PRIMARY KEY* (`*id*_bank`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;

Tabel 4.4 Tabel Jasa Pengiriman

Table Jasa Pengiriman
CREATE TABLE IF NOT EXISTS `jasa_pengiriman` (`*id*_jasa_pengiriman` int(11) *NOT NULL* AUTO_INCREMENT, `nama_jasa_pengiriman` varchar(50) *NOT NULL*, *PRIMARY KEY* (`*id*_jasa_pengiriman`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=2 ; ---- Dumping data for table `jasa_pengiriman`---- -- Constraints for dumped tables---- -- Constraints for table `jasa_pengiriman`-- ALTER TABLE `jasa_pengiriman` ADD CONSTRAINT `jasa_pengiriman_ibfk_1` *FOREIGN KEY* (`*id*_jasa_pengiriman`) REFERENCES `kategori_pengiriman` (`*id*_kategori_pengiriman`) ON DELETE NO ACTION ON MEMPERBARUI CASCADE;

Tabel 4.5 Tabel Kategori

Table Kategori
CREATE TABLE IF NOT EXISTS `kategori` (`*id*_kategori` int(11) *NOT NULL* AUTO_INCREMENT, `nama_kategori` varchar(45) *NOT NULL*,

```
  `seokategori` varchar(100) NOT NULL,
  PRIMARY KEY (`id_kategori`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

Tabel 4.6 Tabel Kategori Pengiriman

Table Kategori Pengiriman

```
CREATE TABLE IF NOT EXISTS `kategori_pengiriman` (
  `id_kategori_pengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `nama_kategori_pengiriman` varchar(50) NOT NULL,
  `id_jasa_pengiriman` int(11) NOT NULL,
  PRIMARY KEY (`id_kategori_pengiriman`),
  KEY `id_jasa_pengiriman` (`id_jasa_pengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2
;
---- Dumping data for table `kategori_pengiriman`----
-- Constraints for dumped tables----
-- Constraints for table `kategori_pengiriman`--
ALTER TABLE `kategori_pengiriman`
  ADD CONSTRAINT `kategori_pengiriman_ibfk_1` FOREIGN KEY
(`id_jasa_pengiriman`) REFERENCES `jasa_pengiriman`
(`id_jasa_pengiriman`) ON DELETE CASCADE ON MEMPERBARUI
CASCADE;
```

Tabel 4.7 Tabel Kota

Table Kota

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(11) NOT NULL AUTO_INCREMENT,
  `id_provinsi` int(11) NOT NULL,
  `nama_kota` varchar(45) NOT NULL,
  PRIMARY KEY (`id_kota`),
  KEY `kdprovinsi` (`id_provinsi`),
  KEY `nama_kota` (`nama_kota`),
  KEY `id_provinsi` (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=99
;
---- Dumping data for table `kota`----
-- Constraints for dumped tables----
-- Constraints for table `kota`--
ALTER TABLE `kota`
  ADD CONSTRAINT `kota_ibfk_1` FOREIGN KEY (`id_provinsi`)
REFERENCES `provinsi` (`id_provinsi`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE; (60, 15, 'Blitar'),
ALTER TABLE `kota`
  ADD CONSTRAINT `kota_ibfk_1` FOREIGN KEY (`id_provinsi`)
```

```
REFERENCES `provinsi` (`id_provinsi`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE;
```

Tabel 4.8 Tabel *Member*

Table Member

```
CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(3) NOT NULL AUTO_INCREMENT,
  `nama_member` varchar(50) NOT NULL,
  `nama pengguna` varchar(50) NOT NULL,
  `kata sandi` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `alamat` varchar(100) NOT NULL,
  `telepon` varchar(13) NOT NULL,
  `id_kota` int(3) NOT NULL,
  `kode_pos` int(5) NOT NULL,
  `tgl_daftar` date NOT NULL,
  `block` enum('ya','tidak') NOT NULL DEFAULT 'ya',
  `kode_verifikasi` varchar(50) NOT NULL,
  `status_masuk` enum('offline','online') NOT NULL DEFAULT 'offline',
  PRIMARY KEY (`id_member`),
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=12
;
---- Dumping data for table `member`----
-- Constraints for dumped tables----
-- Constraints for table `member`--
ALTER TABLE `member`
  ADD CONSTRAINT `member_ibfk_1` FOREIGN KEY (`id_kota`)
REFERENCES `kota` (`id_kota`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE;
```

Tabel 4.9 Tabel Ongkos Kirim

Table Ongkos Kirim

```
CREATE TABLE IF NOT EXISTS `ongkos_kirim` (
  `id_ongkir` int(11) NOT NULL AUTO_INCREMENT,
  `id_kota` int(11) NOT NULL,
  `id_jasa_pengiriman` int(11) NOT NULL,
  `id_kategori_pengiriman` int(11) NOT NULL,
  `ongkos` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_ongkir`),
  KEY `kdkota` (`id_kota`),
  KEY `id_ongkir` (`id_ongkir`),
  KEY `id_kategori_pengiriman` (`id_kategori_pengiriman`),
  KEY `id_jasa_pengiriman` (`id_jasa_pengiriman`),
```

```
  KEY `id_kota` (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=212 ;
---- Dumping data for table `ongkos_kirim`--
---- Constraints for dumped tables----
-- Constraints for table `ongkos_kirim`--
ALTER TABLE `ongkos_kirim`
  ADD CONSTRAINT `ongkos_kirim_ibfk_3` FOREIGN KEY
(`id_kategori_pengiriman`) REFERENCES `kategori_pengiriman`
(`id_kategori_pengiriman`) ON DELETE CASCADE ON MEMPERBARUI
CASCADE,
  ADD CONSTRAINT `ongkos_kirim_ibfk_1` FOREIGN KEY (`id_kota`)
REFERENCES `kota` (`id_kota`) ON DELETE CASCADE ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `ongkos_kirim_ibfk_2` FOREIGN KEY
(`id_jasa_pengiriman`) REFERENCES `jasa_pengiriman`
(`id_jasa_pengiriman`) ON DELETE CASCADE ON MEMPERBARUI
CASCADE;
```

Tabel 4.10 Tabel Pemesanan

Table Pemesanans

```
CREATE TABLE IF NOT EXISTS `pemesanans` (
  `id_pemesanan` int(11) NOT NULL AUTO_INCREMENT,
  `id_member` int(11) NOT NULL,
  `id_pengiriman` int(11) NOT NULL,
  `total_bayar` double NOT NULL,
  `status` enum('Belum Bayar','Telah
Bayar','Lunas','Dikirim','Diterima','Expired','Dibatalkan') NOT NULL
DEFAULT 'Belum Bayar',
  `noresi` varchar(15) NOT NULL DEFAULT '',
  `tgl_masuk` datetime NOT NULL,
  `tgl_memperbarui` timestamp NOT NULL DEFAULT
CURRENT_TIMESTAMP,
  PRIMARY KEY (`id_pemesanan`),
  UNIQUE KEY `id_pengiriman` (`id_pengiriman`),
  KEY `userID` (`id_member`),
  KEY `id_member` (`id_member`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=101 ;
---- Dumping data for table `pemesanans`----
-- Constraints for dumped tables—
---- Constraints for table `pemesanans`--
ALTER TABLE `pemesanans`
  ADD CONSTRAINT `pemesanans_ibfk_2` FOREIGN KEY
(`id_pengiriman`) REFERENCES `pengiriman` (`id_pengiriman`) ON
```

```
DELETE CASCADE ON MEMPERBARUI CASCADE,
  ADD CONSTRAINT `pemesanans_ibfk_1` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`) ON DELETE CASCADE ON
MEMPERBARUI CASCADE;
```

Tabel 4.11 Tabel Pemesanan Detil

Table Pemesanan Detil

```
CREATE TABLE IF NOT EXISTS `pemesanandetil` (
  `id_pemesanandetil` int(11) NOT NULL AUTO_INCREMENT,
  `id_pemesanan` int(11) NOT NULL,
  `id` int(11) NOT NULL,
  `jumlah` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id_pemesanandetil`),
  KEY `produkId` (`id`),
  KEY `pemesananId` (`id_pemesanan`),
  KEY `id_pemesanan` (`id_pemesanan`),
  KEY `id` (`id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=92
;
---- Dumping data for table `pemesanandetil`----
-- Constraints for dumped tables----
-- Constraints for table `pemesanandetil`--
ALTER TABLE `pemesanandetil`
  ADD CONSTRAINT `pemesanandetil_ibfk_2` FOREIGN KEY
(`id_pemesanan`) REFERENCES `pemesanans` (`id_pemesanan`) ON
DELETE CASCADE ON MEMPERBARUI CASCADE,
  ADD CONSTRAINT `pemesanandetil_ibfk_1` FOREIGN KEY (`id`)
REFERENCES `produk` (`id`) ON DELETE CASCADE ON
MEMPERBARUI CASCADE;
```

Tabel 4.12 Tabel *User*

Table User

```
CREATE TABLE IF NOT EXISTS `user` (
  `id_user` int(11) NOT NULL AUTO_INCREMENT,
  `nama_jabatan` enum('pemilik','pegawai') NOT NULL DEFAULT
'pegawai',
  `nama_user` varchar(50) NOT NULL,
  `nama_pengguna` varchar(20) NOT NULL,
  `kata_sandi` varchar(50) NOT NULL,
  `alamat_user` varchar(50) NOT NULL,
  `telepon_user` varchar(13) NOT NULL,
  `e-mail` varchar(50) NOT NULL,
  `no_rek` varchar(45) DEFAULT NULL,
  PRIMARY KEY (`id_user`)
```

```
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

Tabel 4.13 Tabel Pembayaran

Table Pembayaran

```
CREATE TABLE IF NOT EXISTS `pembayaran` (
  `id_pembayaran` int(11) NOT NULL AUTO_INCREMENT,
  `id_pemesanan` int(11) NOT NULL,
  `id_member` int(11) NOT NULL,
  `cara_bayar` enum('Cash On Delivery','Transfer Bank','Paypal') DEFAULT
NULL,
  `id_bank` int(11) NOT NULL,
  `an` varchar(50) DEFAULT NULL,
  `no_rek` int(11) DEFAULT NULL,
  `email_paypal` varchar(50) DEFAULT NULL,
  `kode_matauang` char(45) DEFAULT NULL,
  `tgl_bayar` timestamp NOT NULL DEFAULT CURRENT_TIMESTAMP,
  `jumlah_bayar` double NOT NULL,
  PRIMARY KEY (`id_pembayaran`),
  KEY `id_pemesanan` (`id_pemesanan`),
  KEY `id_member` (`id_member`),
  KEY `id_bank` (`id_bank`),
  KEY `id_pemesanan_2` (`id_pemesanan`),
  KEY `id_bank_2` (`id_bank`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=26
;
---- Dumping data for table `pembayaran`----
-- Constraints for dumped tables----
-- Constraints for table `pembayaran`--
ALTER TABLE `pembayaran`
  ADD CONSTRAINT `pembayaran_ibfk_3` FOREIGN KEY (`id_bank`)
REFERENCES `bank` (`id_bank`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `pembayaran_ibfk_1` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `pembayaran_ibfk_2` FOREIGN KEY
(`id_pemesanan`) REFERENCES `pemesanans` (`id_pemesanan`) ON
DELETE CASCADE ON MEMPERBARUI CASCADE;
```

Tabel 4.14 Tabel Pengiriman

Table Pengiriman

```
CREATE TABLE IF NOT EXISTS `pengiriman` (
  `id_pengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_member` int(11) NOT NULL,
```

```
  `nama_kirim` varchar(255) NOT NULL,
  `id_kota` int(11) NOT NULL,
  `kode_pos_kirim` int(5) NOT NULL,
  `no_telepon_kirim` varchar(255) NOT NULL,
  `id_jasa_pengiriman` int(11) NOT NULL,
  `id_kategori_pengiriman` int(11) DEFAULT NULL,
  `alamat_kirim` text NOT NULL,
  UNIQUE KEY `id_pengiriman` (`id_pengiriman`),
  KEY `id_kota` (`id_kota`),
  KEY `id_kategori_pengiriman` (`id_kategori_pengiriman`),
  KEY `id_member` (`id_member`),
  KEY `id_kota_2` (`id_kota`),
  KEY `id_jasa_pengiriman` (`id_jasa_pengiriman`),
  KEY `id_kategori_pengiriman_2` (`id_kategori_pengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=96
;
---- Dumping data for table `pengiriman`----
-- Constraints for dumped tables----
-- Constraints for table `pengiriman`--
ALTER TABLE `pengiriman`
  ADD CONSTRAINT `pengiriman_ibfk_4` FOREIGN KEY
(`id_kategori_pengiriman`) REFERENCES `kategori_pengiriman`
(`id_kategori_pengiriman`) ON DELETE CASCADE ON MEMPERBARUI
CASCADE,
  ADD CONSTRAINT `pengiriman_ibfk_1` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`) ON DELETE CASCADE ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `pengiriman_ibfk_2` FOREIGN KEY (`id_kota`)
REFERENCES `kota` (`id_kota`) ON DELETE CASCADE ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `pengiriman_ibfk_3` FOREIGN KEY
(`id_jasa_pengiriman`) REFERENCES `jasa_pengiriman`
(`id_jasa_pengiriman`) ON DELETE CASCADE ON MEMPERBARUI
CASCADE;
```

Tabel 4.15 Tabel Produk

Table Produk

```
CREATE TABLE IF NOT EXISTS `produk` (
  `id` int(11) NOT NULL AUTO_INCREMENT,
  `nama_produk` varchar(255) NOT NULL,
  `gambar` varchar(45) DEFAULT NULL,
  `harga` varchar(16) NOT NULL,
  `detil` text NOT NULL,
  `tgl_masuk` date NOT NULL,
  `id_kategori` int(11) NOT NULL,
```

```
  `berat` float NOT NULL,
  `ukuran` varchar(3) NOT NULL,
  `warna` varchar(25) NOT NULL,
  `gender` varchar(20) NOT NULL,
  `diskon` float NOT NULL DEFAULT '0',
  `stok` int(11) NOT NULL,
  `terjual` int(4) DEFAULT '0',
  `hits` int(11) NOT NULL DEFAULT '0',
  `restok` int(11) NOT NULL DEFAULT '0',
  `keaktifan` varchar(20) DEFAULT NULL,
  `id_warna` int(11) NOT NULL,
  `id_ukuran` int(11) NOT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `nama_produk` (`nama_produk`),
  KEY `id_kategori` (`id_kategori`),
  KEY `id_warna` (`id_warna`),
  KEY `id_ukuran` (`id_ukuran`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=18
;
---- Dumping data for table `produk`----
-- Constraints for dumped tables----
-- Constraints for table `produk`--
ALTER TABLE `produk`
  ADD CONSTRAINT `produk_ibfk_3` FOREIGN KEY (`id_ukuran`)
REFERENCES `ukuran` (`id_ukuran`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `produk_ibfk_1` FOREIGN KEY (`id_kategori`)
REFERENCES `kategori` (`id_kategori`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `produk_ibfk_2` FOREIGN KEY (`id_warna`)
REFERENCES `warna` (`id_warna`) ON DELETE NO ACTION ON
MEMPERBARUI CASCADE;
```

Tabel 4.16 Tabel Provinsi

Table Provinsi

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(11) NOT NULL AUTO_INCREMENT,
  `nama_provinsi` varchar(45) NOT NULL,
  PRIMARY KEY (`id_provinsi`),
  KEY `nama_provinsi` (`nama_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=35
;
```

Tabel 4.17 Tabel Pengembalian

Table Pengembalian
CREATE TABLE IF NOT EXISTS `pengembalian` (`*id*_pengembalian` int(11) *NOT NULL* AUTO_INCREMENT, `*id*_pemesanan` int(11) *NOT NULL*, `*id_member*` int(11) *NOT NULL*, `status` enum('Menunggu Konfirmasi Distro','Dikonfirmasi','Dikirim','Diterima','Dikembalikan','Ditolak') *NOT NULL*, `noresi` varchar(100) *NOT NULL*, `tgl_masuk` datetime *NOT NULL*, `tgl_memperbarui` timestamp *NOT NULL* DEFAULT CURRENT_TIMESTAMP, *PRIMARY KEY* (`*id*_pengembalian`), KEY `userID` (`*id_member*`), KEY `*id*_pemesanan` (`*id*_pemesanan`), KEY `*id*_pemesanan_2` (`*id*_pemesanan`), KEY `*id_member*` (`*id_member*`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=274 ; ---- Dumping data for table `pengembalian`-- ---- Constraints for dumped tables---- -- Constraints for table `pengembalian`-- ALTER TABLE `pengembalian` ADD CONSTRAINT `pengembalian_ibfk_2` *FOREIGN KEY* (`*id_member*`) REFERENCES `*member*` (`*id_member*`) ON DELETE CASCADE ON MEMPERBARUI CASCADE, ADD CONSTRAINT `pengembalian_ibfk_1` *FOREIGN KEY* (`*id*_pemesanan`) REFERENCES `pemesanans` (`*id*_pemesanan`) ON DELETE CASCADE ON MEMPERBARUI CASCADE;

Tabel 4.18 Tabel Pengembalian Detil

Table Pengembalian Detil
CREATE TABLE IF NOT EXISTS `pengembalian_detil` (`*id*_pengembaliandetil` int(11) *NOT NULL* AUTO_INCREMENT, `*id*_pengembalian` int(11) *NOT NULL*, `*id*` int(11) *NOT NULL*, `jumlah` int(11) *NOT NULL*, `alasan` varchar(50) *NOT NULL*, *PRIMARY KEY* (`*id*_pengembaliandetil`), KEY `pengembalianID` (`*id*_pengembalian`), KEY `produkID` (`*id*`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=48 ; ---- Dumping data for table `pengembalian_detil`----

```
-- Constraints for dumped tables----
-- Constraints for table `pengembalian_detil`--
ALTER TABLE `pengembalian_detil`
  ADD CONSTRAINT `pengembalian_detil_ibfk_2` FOREIGN KEY (`id`)
REFERENCES `produk` (`id`) ON DELETE CASCADE ON
MEMPERBARUI CASCADE,
  ADD CONSTRAINT `pengembalian_detil_ibfk_1` FOREIGN KEY
(`id_pengembalian`) REFERENCES `pengembalian` (`id_pengembalian`) ON
DELETE CASCADE ON MEMPERBARUI CASCADE;
```

Tabel 4.19 Ukuran

Table Ukuran
CREATE TABLE IF NOT EXISTS `ukuran` (`id_ukuran` int(11) *NOT NULL* AUTO_INCREMENT, `ukuran` varchar(10) *NOT NULL*, *PRIMARY KEY* (`id_ukuran`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=18 ;

Tabel 4.20 Tabel Warna

Table Warna
CREATE TABLE IF NOT EXISTS `warna` (`id_warna` int(11) *NOT NULL* AUTO_INCREMENT, `warna` varchar(20) *NOT NULL*, *PRIMARY KEY* (`id_warna`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;

4.3 Implementasi Antar Muka

Setelah melakukan implementasi data dilakukanlah implementasi antarmuka yang berguna sebagai tampilan untuk pengguna sistem. Implementasi dilakukan pada setiap hasil perancangan antarmuka yang telah dibuat kedalam bentuk aplikasi pada Adobe Dreamweaver CS5.

Untuk mengetahui lebih jelas tentang proses yang ada didalam sistem beserta hasil implementasinya dapat dilihat dalam tabel berikut:

Tabel 4.21 Implementasi Antarmuka Pegawai

Menu	Deskripsi	Nama File

Login	Digunakan sebagai halaman masuk Pemilik	*Staff_login*.php
Home Pegawai	Digunakan sebagai halaman beranda Pemilik dan terdapat beberapa menu diantaranya menu pengolahan data master, data transaksi, dan laporan.	*Index*.php
Manajemen Kategori	Halaman yang berisi informasi Kategori	Kategori_*list*.php
Tambah Kategori	Halaman yang brguna untuk menambah data Kategori	Kategori _tambah.php
Ubah Kategori	Halaman yang digunakan untuk mengubah data Kategori.	Kategori_edit.php
Manajemen Produk	Halaman yang berisi informasi produk	produk_*list*.php
Tambah produk	Halaman yang brguna untuk menambah data produk	produk_tambah.php
Ubah produk	Halaman yang digunakan untuk mengubah data produk	produk_edit.php
Warna	Halaman yang berisi informasi warna	Warna_*list*.php
Tambah warna	Halaman yang brguna untuk menambah data warna	Warna_tambah.php
Ubah warna	Halaman yang digunakan untuk mengubah data warna	Warna_edit.php
Manajemen bank	Halaman yang berisi informasi bank	bank _*list*.php
Tambah bank	Halaman yang brguna untuk menambah data bank	bank _tambah.php
Ubah bank	Halaman yang digunakan untuk mengubah data bank	bank _edit.php
Manajemen jasa pengiriman	Halaman yang berisi informasi jasa pengiriman	Jasa_pengiriman _*list*.php
Tambah jasa pengiriman	Halaman yang brguna untuk menambah data jasa pengiriman	Jasa_pengiriman _tambah.php
Ubah jasa pengiriman	Halaman yang digunakan untuk mengubah data jasa pengiriman	Jasa_pengiriman _edit.php
Manajemen kategori pengiriman	Halaman yang berisi informasi kategori pengiriman	kategori _pengiriman _*list*.php
Tambah kategori pengiriman	Halaman yang brguna untuk menambah data kategori pengiriman	kategori _pengiriman _tambah.php
Ubah kategori pengiriman	Halaman yang digunakan untuk mengubah data kategori pengiriman	kategori _pengiriman _edit.php

Manajemen *member*	Halaman yang berisi informasi *member*	*member* _*list*.php
Tambah *member*	Halaman yang brguna untuk menambah data *member*	*member* _tambah.php
Ubah *member*	Halaman yang digunakan untuk mengubah data *member*	*member* _edit.php
Manajemen kota	Halaman yang berisi informasi kota	kota_*list*.php
Tambah kota	Halaman yang brguna untuk menambah data kota	kota_tambah.php
Ubah kota	Halaman yang digunakan untuk mengubah data kota	kota_edit.php
Manajemen Provinsi	Halaman yang berisi informasi provinsi	provinsi_*list*.php
Tambah provinsi	Halaman yang brguna untuk menambah data provinsi	provinsi_tambah.php
Ubah provinsi	Halaman yang digunakan untuk mengubah data provinsi	provinsi_edit.php
Manajemen Ongkir	Halaman yang berisi informasi Ongkir	Ongkir _*list*.php
Tambah Ongkir	Halaman yang brguna untuk menambah data Ongkir	Ongkir _tambah.php
Ubah Ongkir	Halaman yang digunakan untuk mengubah data Ongkir	Ongkir _edit.php
Manajemen Hubungi	Halaman yang berisi informasi komentar	hubungi _*list*.php
backup	Halaman yang brguna untuk mem-*backup database*	*Backup*.php
restore	Halaman yang brguna untuk me-*restore database*	*Restore*.php
Manajemen Pesanan	Halaman yang berisi informasi pesanan	pemesanan _*list*.php
Ubah status	Halaman digunakan untuk mengubah status pesanan	pesanan_edit.php
Manajemen Pengembalian	Halaman yang berisi informasi pengembalian	pengembalian _*list*.php
Ubah status	Halaman digunakan untuk mengubah status pengembalian	pengembalian_edit.php
Logout	Digunakan pegawai untuk keluar dari aplikasi.	*Logout*.php

Tabel 4.22 Implementasi Antarmuka Pemilik

Menu	Deskripsi	Nama File

Login	Digunakan sebagai halaman masuk Pemilik	*Admin_login*.php
Home Pemilik	Digunakan sebagai halaman beranda Pemilik dan terdapat beberapa menu diantaranya menu pengolahan data master dan laporan.	*Index*.php
Pegawai	Halaman yang berisi informasi pegawai	Pegawai_*list*.php
Tambah pegawai	Halaman yang brguna untuk menambah data *user*	pegawai_tambah.php
Ubah pegawai	Halaman yang digunakan untuk mengubah data *user*	pegawai _ubah.php
Laporan penjualan	Halaman yang berisi informasi mengenai laporan pemesanan	laporan_jual.php
Laporan stok	Halaman yang berisi informasi mengenai laporan stok yang tersedia	laporan_stok.php
Logout	Digunakan Pemilik untuk keluar dari aplikasi.	*Logout*.php

Tabel 4.23 Implementasi Antarmuka Pengunjung

Menu	Deskripsi	Nama File
Pendaftaran	Digunakan sebagai halaman pendaftaran sebagai *member*	registrasi.php
Beranda	Digunakan sebagai halaman beranda	*index*.php
Semua Produk	Halaman yang berisi informasi mengenai data produk	Produk_*list*.php
Cara pembelian	Halaman yang berisi informasi mengenai tutorial pemesanan,pembayaran dan pengembalian	Cara_beli.php
Profil Perusahaan	Halaman yang berisi informasi tentang perusahaan	Profil_perusahaan.php
Hubungi Kami	Halaman ini berguna untuk mengirim pesan ke pada Pemilik disaat Pemilik tidak *online*	Hubungi_kami.php

Tabel 4.24 Implementasi Antarmuka *Member*

Menu	Deskripsi	Nama File
Login	Digunakan sebagai halaman masuk untuk masuk kedalam aplikasi	*login*.php

Beranda	Digunakan sebagai halaman beranda	*index*.php
Semua Produk	Halaman yang berisi informasi mengenai data produk	semua_produk.php
Cara pembelian	Halaman yang berisi informasi mengenai tutorial pemesanan,pembayaran dan pengembalian	Cara_pemesanan.php
Profil	Halaman yang berisi informasi tentang perusahaan	Profile_*member*.php
Hubungi Kami	Halaman ini berguna untuk mengirim pesan ke pada Pemilik disaat Pemilik tidak *online*	Hubungi_kami.php
Ganti kata sandi	Halaman ini berguna untuk *member* mengganti kata sandi nya	lupa.php
Ubah akun	Halaman ini berguna untuk *member* mengganti data pribadinya	Edit_profile.php
History Pemesanan	Halaman ini berguna untuk melihat *history* pemesanan dan status pemesanan	*History*_pemesanan.php
Keranjang belanja	Digunakan untuk melihat, mengubah, menghapus data pembelian yang akan dilakukan	*cart*.php
Pengisian Alamat kirim	Digunakan untuk memasukan alamat kirim dan ongkos kirim	Alamat_kirim.php
Checkout	Halaman *review* pemesanan dan digunakan untuk *checkout*	*Checkout*.php
Pilih pembayaran	Digunakan untuk memilih cara pembayaran	Pilih_pembayaran.php
History pemesanan	Digunakan untuk melihat *history* pemesanan	*History*_pemesanan.php
History pemesanan detil	Digunakan untuk melihat detil pemesanan	*History*_pemesanan detil.php
Konfirmasi pembayran	Digunakan untuk melakukan konfirmasi pembayran	Konfirmasi_pembayaran.php
Pengembalian	Digunakan untuk melakukan pengembalian	Pengembalian.php
Pengembalian detil	Digunakan untuk melihat detil pengembalalian	*Retur*_detail.php
History pengembalian	Digunakan untuk melihat riwayat pengembalalian	*History*_pengembalian.php
Srs	Menyuguhkan informasi srs content based	*Content_based*.php
Logout	Digunakan *member* untuk keluar dari aplikasi.	*Logout*.php

4.4 Pengujian Alpha

4.4.1 Metode Pengujian

Sebuah aplikasi yang akan diterapkan haruslah bebas dari kesalahan-kesalahan atau *error*. Oleh karena itu, aplikasi harus diuji coba terlebih dahulu untuk menemukan kesalahan-kesalahan yang mungkin terjadi.

4.4.2 Rencana Pengujian

Pengujian program ini menggunakan metode *black box.* Pengujian *black box* merupakan pengujian program berdasarkan fungsi dari program. Tujuan dari metode *Black Box* ini adalah untuk menemukan kesalahan fungsi pada program. Pengujian dengan metode *Black box* dilakukan dengan cara memberikan sejumlah *input* pada program aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah aplikasi yang kemudian diproses sesuai dengan kebutuhan fungsionalnya untuk melihat apakah aplikasi menghasilkan *output* yang diinginkan dan sesuai dengan fungsi dari program tersebut. Apabila dari *input* yang diberikan proses menghasilkan *output* yang sesuai dengan kebutuhan fungsionalnya, maka aplikasi yang bersangkutan telah benar, tetapi jika *output* yang dihasilkan tidak sesuai dengan kebutuhan fungsionalnya, maka masih terdapat kesalahan pada program aplikasi.

Pengujian dilakukan dengan mencoba semua kemungkinan yang terjadi dan pengujian dilakukan berulang-ulang. Jika dalam pengujian ditemukan kesalahan, maka akan dilakukan penelusuran dan perbaikan untuk memperbaiki kesalahan yang terjadi. Jika telah selesai melakukan perbaikan, maka akan dilakukan secara terus menerus hingga diperoleh hasil yang terbaik.

4.4.1 Skenario Pengujian Alpa

Skenario pengujian alpha di Distro Branson For Life terdiri dari empat pelaku sistem diantaranya *member*, pengunjung, pegawai dan pemilik.

Tabel 4.25 Skenario Pengujian Aplikasi Pegawai

Kelas Uji	Butir Uji	Jenis

		Pengujian
Login	Isi data *login*	*Black Box*
	Validasi nama pengguna *dan* kata sandi	*Black Box*
Pengolahan Data Produk	Pengolahan Data Kategori	*Black Box*
	Pengolahan Data Produk	*Black Box*
	Pengolahan Data Warna	*Black Box*
	Pengeolahan Data Ukuran	*Black Box*
Pengolahan Data Kategori	Tampil Data Kategori	*Black Box*
	Tambah Data Kategori	*Black Box*
	Ubah Data Kategori	*Black Box*
Pengolahan Data Produk	Tampil Data Produk	*Black Box*
	Tambah Data Produk	*Black Box*
	Ubah Data Produk	*Black Box*
Pengolahan Data Warna	Tampil Data Warna	*Black Box*
	Tambah Data Warna	*Black Box*
	Ubah Data Warna	*Black Box*
Pengolahan Data Ukuran	Tampil Data Ukuran	*Black Box*
	Tambah Data Ukuran	*Black Box*
	Ubah Data Ukuran	*Black Box*
Pengolahan Data Pemesanan	Pengolahan Data Pemesanan	*Black Box*
	Pengolahan Data Pembayaran	*Black Box*
	Pengolahan Data Pengembalian	*Black Box*
	Pengolahan Data Restok	*Black Box*
Pengolahan Data Pemesanan	Tampil Data Pemesanan	*Black Box*
	Ubah Status Pemesanan	*Black Box*
	Tampil Detil Pemesanan	*Black Box*
	Input Nomor Resi	*Black Box*
	Cetak Label	*Black Box*
Pengolahan Data Pembayaran	Tampil Data Pembayaran	*Black Box*
	Tampil Detil Pembayaran	*Black Box*
Pengolahan Data Pengembalian	Tampil Data Pengembalian	*Black Box*
	Tampil Detil Pengembalian	*Black Box*
	Ubah Status Pengembalian	*Black Box*
Pengolahan Data Restok	Tampil Data Restok	*Black Box*
Pengolahan Data Pengiriman	Pengolahan Data Provinsi	*Black Box*
	Pengolahan Data Kota	*Black Box*
	Pengolahan Data Jasa Pengiriman	*Black Box*
	Pengolahan Data Kategori Pengiriman	*Black Box*
	Pengolahan Data Ongkos Kirim	*Black Box*
Pengolahan Data Provinsi	Tampil Data Provinsi	*Black Box*
	Tambah Data Provinsi	*Black Box*
	Ubah Data Provinsi	*Black Box*
Pengolahan Data Kota	Tampil Data Kota	*Black Box*

	Tambah Data Kota	*Black Box*
	Ubah Data Kota	*Black Box*
Pengolahan Data Jasa Pengiriman	Tampil Data Jasa Pengiriman	*Black Box*
	Tambah Data Jasa Pengiriman	*Black Box*
	Ubah Data Jasa Pengiriman	*Black Box*
Pengolahan Data Kategori Pengiriman	Tampil Data Kategori Pengiriman	*Black Box*
	Tambah Data Kategori Pengiriman	*Black Box*
	Ubah Data Kategori Pengiriman	*Black Box*
Pengolahan Data Ongkos Kirim	Tampil Data Ongkos Kirim	*Black Box*
	Tambah Data Ongkos Kirim	*Black Box*
	Ubah Data Ongkos Kirim	*Black Box*
Pengolahan Data Distro	Pengolahan Data *Member*	*Black Box*
	Pengolahan Data Bank	*Black Box*
	Pengolahan Data Hubungi Kami	*Black Box*
	Pengeolahan Data *Backup Restore*	*Black Box*
Pengolahan Data *Member*	Tampil Data *Member*	*Black Box*
	Tambah Data *Member*	*Black Box*
	Ubah Data *Member*	*Black Box*
Pengolahan Data Bank	Tampil Data Bank	*Black Box*
	Tambah Data Bank	*Black Box*
	Ubah Data Bank	*Black Box*
Pengolahan Data Hubungi Kami	Tampil Data Hubungi Kami	*Black Box*
Backup Restore	Tampil Halaman *Backup Restore*	*Black Box*
Logout	Menghapus *session* pegawai	*Black Box*

Tabel 4.26 Skenario Pengujian Aplikasi Pengunjung

Kelas Uji	Butir Uji	Jenis Pengujian
Home	Menampilkan halaman utama.	*Black Box*
Daftar *member*	Menampilkan *form member* baru	*Black Box*
	Isi *form* pendaftaran	*Black Box*
	Simpan data baru	*Black Box*
Lupa kata sandi	Verifikasi *Email*	*Black Box*
Produk	Menampilkan barang	*Black Box*
Cara pembelian	Menampilkan cara pembelian	*Black Box*
Tentang kami	Menampilkan tentang kami	*Black Box*
Hubungi	Menampilkan *form*	*Black Box*
	Isi *form* hubungi	*Black Box*
Keranjang belanja	Menampilkan data pembelian	*Black Box*
	Mengmemperbarui data	*Black Box*

	pembelian	
	Menghapus data pembelian	*Black Box*

Tabel 4.27 Skenario Pengujian Aplikasi *Member*

Kelas Uji	Butir Uji	Jenis Pengujian
Home	Menampilkan halaman utama.	*Black Box*
Daftar *member*	Menampilkan *form user* baru	*Black Box*
	Isi *form* pendaftaran	*Black Box*
Login	Isi data *masuk*	*Black Box*
	Verifikasi Nama pengguna	*Black Box*
	Verifikasi Kata sandi	*Black Box*
Lupa kata sandi	Verifikasi *Email*	*Black Box*
Produk	Menampilkan barang	*Black Box*
Cara pembelian	Menampilkan cara pembelian	*Black Box*
Tentang kami	Menampilkan tentang kami	*Black Box*
Hubungi	Menampilkan *form*	*Black Box*
	Isi *form* hubungi	*Black Box*
Ubah akun	Menampilkan *form* ubah akun	*Black Box*
	Ubah data akun	*Black Box*
	Simpan perubahan	*Black Box*
Ubah kata sandi	Menampilkan *form* ubah kata sandi	*Black Box*
Keranjang belanja	Menampilkan data pembelian	*Black Box*
	Mengmemperbarui 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*
Lihat *history*	Menampilkan *history* pemesanan	*Black Box*
	Melakukan konfirmasi pembayaran	*Black Box*
Pengembalian	Menampilkan pengembalian pembelian	*Black Box*
	Memilih barang yang di pengembalian	*Black Box*
Rekomendasi	Menampilkan produk rekomendasi	*Black Box*

Logout	Menghapus session *user*	*Black Box*

Tabel 4.28 Skenario Pengujian Aplikasi Pemilik

Kelas Uji	Butir Uji	Jenis Pengujian
Login Pemilik	Isi data *masuk*	*Black Box*
	Verifikasi nama pengguna	*Black Box*
	Verifikasi kata sandi	*Black Box*
Manajemen pegawai	Tampil Data pegawai	*Black Box*
	Tambah Data pegawai	*Black Box*
	Ubah Data pegawai	*Black Box*
	Hapus Data pegawai	*Black Box*
Laporan stok	Tampilkan data stok barang	*Black Box*
Laporan penjualan	Tampilkan data laporan penjualan	*Black Box*
Logout	Menghapus session *user*	*Black Box*

4.4.2 Kasus dan Hasil Pengujian

4.4.2.1 Pengujian Daftar *Member*

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

Tabel 4.29 Pengujian Daftar *Member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data pendaftaran, contoh masukan : Nama pengguna : gankro Kata sandi: gankro Nama : naufal syafiq rani Alamat :jl teratai mekar 29 *Email* : naufal.syafiq.lee@gmail.com Telepon : 08562295xxx Provinsi : Jawa Barat Kota : Bandung Kodepos : 40613	Jika semua *field* terisi atau benar dalam pengisiann, maka akan dapat melajutkan daftar.	*Form* daftar terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses daftar.	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data kosong	Tampilkan	Tampilkan	[✓] Diterima

	pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* tersebut harus terisi.	pesan kesalahan dalam pengisian, maka muncul pesan kesalahan bahwa *field* tersebut harus terisi	[ ] Ditolak
Nama Lengkap tidak diisi	Tampilkan pesan kesalahan "silakan isikan nama"	Dapat menampilkan pesan kesalahan "silakan isikan nama"	[✓] Diterima [ ] Ditolak
Nama pengguna tidak diisi	Tampilkan pesan kesalahan "silakan isikan nama pengguna"	Dapat menampilkan pesan kesalahan "silakan isikan nama pengguna"	[✓] Diterima [ ] Ditolak
Email tidak diisi	Tampilkan pesan kesalahan "silakan isikan *email* anda"	Dapat menampilkan pesan kesalahan "silakan isikan *email* anda"	[✓] Diterima [ ] Ditolak
Format *Email* Salah	Tampilkan pesan kesalahan "format *email* salah"	Dapat menampilkan pesan kesalahan "format *email* salah"	[✓] Diterima [ ] Ditolak
Alamat tidak diisi	Tampilkan pesan kesalahan "silakan isikan alamat"	Dapat menampilkan pesan kesalahan "silakan isikan alamat"	[✓] Diterima [ ] Ditolak
Kodepos tidak diisi	Tampilkan pesan kesalahan "Kodepos harus	Dapat menampilkan pesan	[✓] Diterima [ ] Ditolak

	diisi”	kesalahan “kode harus diisi”	
Kodepos diisi dengan huruf	Tampilkan pesan kesalahan “kodepos harus angka”	Dapat menampilkan pesan kesalahan “kodepos harus angka”	[✓] Diterima [ ] Ditolak
Kata sandi	Tampilkan pesan kesalahan “silakan isi kata sandi”	Dapat menampilkan pesan kesalahan “silakan isi kata sandi”	[✓] Diterima [ ] Ditolak
Ulangi Kata sandi	Tampilkan pesan kesalahan “kata sandi harus sama”	Dapat menampilkan pesan kesalahan “kata sandi harus sama”	[✓] Diterima [ ] Ditolak

4.4.2.2 Pengujian *Login Member*

Berikut dapat dilihat pengujian *masuk member* pada Tabel dibawah ini :

Tabel 4.30 Pengujian *Login Member*

Kasus dan Hasil Uji (DataBenar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan nama pengguna : gankro	Mengisikan data nama pengguna pada *field* nama pengguna	Dapat mengisi *field* nama pengguna dan mengisi data *login* dengan benar	[✓] Diterima [ ] Ditolak
Memasukan kata sandi : gankro	Mengisikan data kata sandi pada *field* kata sandi	Mengisikan data kata sandi pada *field* kata sandi	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan nama pengguna atau kata sandi yang salah	Tampilkan pesan kesalahan “nama pengguna atau kata sandi yang anda masukan salah”	Muncul pesan “Nama pengguna atau kata sandi yang anda masukan salah”	[✓] Diterima [ ] Ditolak
Mengosongkan nama	Tampilkan pesan	Muncul pesan	[✓] Diterima

pengguna	kesalahan "silakan isikan nama pengguna"	kesalahan "silakan isikan nama pengguna"	[ ] Ditolak
Mengosongkan kata sandi	Tampilkan pesan kesalahan "silakan isikan kata sandi"	Muncul pesan kesalahan "silakan isikan kata sandi	[✓] Diterima [ ] Ditolak

4.4.2.3 Pengujian Keranjang Belanja *Member*

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

Tabel 4.31 Pengujian Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukan data jumlah	Tercantum pada text box data yang akan diubah	Dapat mengisi data jumlah dengan benar	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan kuantitas melebihi stok yang tersedia	Tampilkan pesan kesalahan "Maaf, jumlah pesanan Anda melebihi stok yang tersedia. Stok tersedia sejumlah x potong"	Muncul pesan "Maaf, jumlah pesanan Anda melebihi stok yang tersedia. Stok tersedia sejumlah x potong"	[✓] Diterima [ ] Ditolak

4.4.2.4 Pengujian Pengiriman

Berikut dapat dilihat pengujian dari *Form* Pengiriman pada tabel di bawah ini:

Tabel 4.32 Pengujian *Form* Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama penerima	Mengisikan data nama penerima pada *field* nama penerima	Dapat mengisi *field* nama penerima dan mengisi data pengiriman dengan benar	[✓] Diterima [ ] Ditolak
Alamat penerima	Mengisikan data alamat penerima pada *field* alamat	Dapat mengisi *field* alamat penerima dan	[✓] Diterima [ ] Ditolak

	penerima	mengisi data pengiriman dengan benar	
Provinsi tujuan	Mengisikan data nama provinsi pada *field* provinsi tujuan	Dapat mengisi *field* provinsi tujuan dan mengisi data pengiriman dengan benar	[✓] Diterima [ ] Ditolak
Kota tujuan	Mengisikan data kota tujuan pada *field* nama penerima	Dapat mengisi *field* kota tujuan dan mengisi data pengiriman dengan benar	[✓] Diterima [ ] Ditolak
Kode pos	Mengisikan data kode pos pada *field* kode pos	Dapat mengisi *field* kode pos dan mengisi data pengiriman dengan benar	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama penerima	Menampilkan pesan kesalahan "Nama harus diisi"	Dapat menampilkan pesan kesalahan "Nama harus diisi"	[✓] Diterima [ ] Ditolak
Alamat penerima	Menampilkan pesan kesalahan "Alamat harus diisi dengan jelas"	Dapat menampilkan pesan kesalahan "Alamat harus diisi dengan jelas"	[✓] Diterima [ ] Ditolak
Provinsi tujuan	Menampilkan pesan kesalahan "Silahkan pilih"	Dapat menampilkan pesan kesalahan "Silahkan pilih"	[✓] Diterima [ ] Ditolak
Kota tujuan	Menampilkan pesan kesalahan "Silahkan pilih"	Dapat menampilkan pesan kesalahan "Silahkan pilih"	[✓] Diterima [ ] Ditolak
Kode pos	Menampilkan pesan kesalahan "Kode pos harus diisi"	Dapat menampilkan pesan kesalahan "Kode pos harus diisi"	[✓] Diterima [ ] Ditolak

4.4.2.5 Pengujian Konfirmasi Pembayaran

Berikut dapat dilihat pengujian dari *Form* Pembayaran pada tabel di bawah ini:

Tabel 4.33 Pengujian Konfirmasi Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh *Id* pemesanan : 2 Atas Nama : Ferry Dari Bank : Mandiri No Rek. ; 0784832423789 Tanggal transfer : 2011/07/23 Jumlah Transfer : 125000	Menngisikan data konfirmasi pembayaran dan dapat mengisi *field* konfrmasi pembayaran dengan benar	Dapat memunculkan pesan konfirmasi sudah tersimpan.	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisikan atasnama	Tampilkan pesan kesalahan ”silakan isikan an pengirim”	Dapat menampilkan pesan kesalahan ” silakan isikan an pengirim”	[✓] Diterima [ ] Ditolak
Tidak mengisikan bank asal	Tampilkan pesan kesalahan ”silakan pilih bank asal pengiriman”	Dapat menampilkan pesan kesalahan ” silakan isikan an pengirim”	[✓] Diterima [ ] Ditolak
Tidak mengisikan no rekening	Tampilkan pesan kesalahan ”silakan isikan no rekening pengirim”	Dapat menampilkan pesan kesalahan ” silakan isikan no rekening pengirim”	[✓] Diterima [ ] Ditolak
Tidak mengisikan data tanggal transfer	Tampilkan pesan kesalahan ”silakan pilih tanggal transfer”	Dapat menampilkan pesan kesalahan ” silakan pilih tanggal transfer”	[✓] Diterima [ ] Ditolak
Tidak mengisikan data jumlah transfer	Tampilkan pesan kesalahan ”silakan isikan jumlah	Dapat menampilkan pesan kesalahan ”	[✓] Diterima [ ] Ditolak

	transfer”	silakan isikan jumlah transfer”	

4.4.2.6 Pengujian *Login* Pegawai

Berikut dapat dilihat pengujian dari *login* pegawai pada tabel di bawah ini:

Tabel 4.34 Pengujian *Login* Pegawai

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama pengguna *:* abokBrnsn	Tercantum pada text box nama pengguna	Dapat mengisi *field* nama pengguna dengan benar	[✓] Diterima [ ] Ditolak
Kata sandi : abokBrnsn	Tercantum pada text box kata sandi	Dapat mengisi *field* kata sandi dengan benar	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama pengguna atau kata sandi tidak terdaftar	Tidak dapat *login*	Muncul pesan “nama pengguna atau kata sandi salah, silahkan coba lagi.”	[✓] Diterima [ ] Ditolak
Data nama pengguna atau kata sandi kosong	Tidak dapat *login*	Muncul pesan kesalahan “nama pengguna atau kata sandi salah, silahkan coba lagi.”	[✓] Diterima [ ] Ditolak

4.4.2.7Pengujian Pengolahan Tambah Data Kategori

Berikut dapat dilihat pengujian dari tambah data kategori pada tabel di bawah ini:

Tabel 4.35 Pengujian Tambah Data Kategori

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data Kategori Contoh: Nama Kategori: Tees	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah Kategori	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses tambah Kategori	[✓] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama Kategori masih kosong	Tampilkan pesan kesalahan "Kategori harus diisi"	Dapat menampilkan pesan kesalahan "Kategori harus diisi"	[✓] Diterima [ ] Ditolak
Data nama kategori sudah ada	Tampilkan pesan kesalahan "kategori sudah ada"	Dapat menampilkan pesan kesalahan "kategori sudah ada"	[✓] Diterima [ ] Ditolak

4.4.2.8 Pengujian Pengolahan Ubah Data Kategori

Berikut dapat dilihat pengujian dari ubah data Kategori pada tabel di bawah ini:

Tabel 4.36 Pengujian Ubah Data Kategori

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data Kategori Contoh: Nama Kategori: Tees	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses ubah data Kategori	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses ubah data Kategori	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama Kategori masih kosong	Tampilkan pesan kesalahan "Kategori harus diisi"	Muncul pesan kesalahan "Kategori harus diisi"	[✓] Diterima [ ] Ditolak
Data nama kategori sudah ada	Tampilkan pesan kesalahan "kategori sudah ada"	Dapat menampilkan pesan kesalahan "kategori sudah ada"	[✓] Diterima [ ] Ditolak

4.4.2.9 Pengujian Pengolahan Cari Data Kategori

Berikut dapat dilihat pengujian dari cari data Kategori pada tabel di bawah ini :

Tabel 4.37 Pengujian Cari Data Kategori

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi cari	Jika semua *field*	Jika semua *field*	[✓] Diterima

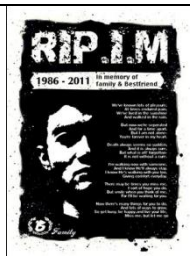
Kategori Contoh: cari Kategori: Tees	terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari Kategori	terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari Kategori	[ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama Kategori masih kosong	Tampilkan pesan kesalahan "kategori tidak ada"	Dapat menampilkan pesan kesalahan "Kategori tidak ada"	[✓] Diterima [ ] Ditolak
Data nama kategori tidak ada	Tampilkan pesan kesalahan "Kategori tidak ada"	Dapat menampilkan pesan kesalahan "Kategori tidak ada"	[✓] Diterima [ ] Ditolak

4.4.2.10 Pengujian Pengolahan Tambah Data Produk

Berikut dapat dilihat pengujian dari tambah produk pada tabel di bawah ini:

Tabel 4.38 Pengujian Tambah Data Produk

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data produk Contoh: Nama Produk: RIP IM Pth M Harga : 75000 Detil: blablabla Kategori: Tees Berat : 0.4 kg Ukuran: M Warna : Putih For (Gender): Male Stok : 20 Diskon: …. Gambar:	Jika semua *field* terisi semua maka akan muncul pesan " data berhasil ditambah"	Jika semua *field* terisi semua maka akan muncul pesan " data berhasil ditambah"	[✓] Diterima [ ] Ditolak

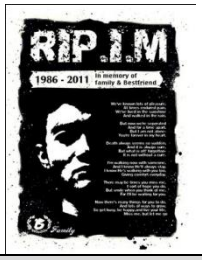
			
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama produk kosong	Muncul pesan kesalahan“nama produk harus diisi”	Dapat menampilkan pesan kesalahan “nama produk harus diisi”	[✓] Diterima [ ] Ditolak
Harga tidak diisi	Muncul pesan kesalahan“harga harus diisi”	Dapat menampilkan pesan kesalahan “harga harus diisi”	[✓] Diterima [ ] Ditolak
Detil tidak diisi	Muncul pesan kesalahan“detil harus diisi”	Dapat menampilkan pesan kesalahan “detil harus diisi”	[✓] Diterima [ ] Ditolak
Kategori tidak dipilih	Muncul pesan kesalahan“kategori harus dipilih”	Dapat menampilkan pesan kesalahan “kategori harus diilih”	[✓] Diterima [ ] Ditolak
Berat tidak diisi	Muncul pesan kesalahan“berat harus diisi”	Dapat menampilkan pesan kesalahan “berat harus diisi”	[✓] Diterima [ ] Ditolak
Ukuran tidak diisi	Muncul pesan kesalahan“ukuran harus diisi”	Dapat menampilkan pesan kesalahan “ukuran harus diisi”	[✓] Diterima [ ] Ditolak
Warna tidak diisi	Muncul pesan kesalahan“warna harus diisi”	Dapat menampilkan pesan kesalahan “warna harus diisi”	[✓] Diterima [ ] Ditolak
Stok tidak diisi	Muncul pesan kesalahan“stok harus diisi”	Dapat menampilkan pesan kesalahan	[✓] Diterima [ ] Ditolak

		"stok harus diisi"	
Diskon tidak diisi	Muncul pesan kesalahan"diskon harus diisi"	Muncul pesan kesalahan "diskon harus diisi"	[✓] Diterima [ ] Ditolak

4.4.2.11 Pengujian Pengolahan Ubah Data Produk

Berikut dapat dilihat pengujian dari ubah data Kategori pada tabel di bawah ini:

Tabel 4.39 Pengujian Ubah Data Produk

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data produk Contoh: Nama Produk: RIP IM Pth M Harga : 75000 Detil: blablabla Kategori: Tees Berat : 0.4 kg Ukuran: M Warna : Putih For (Gender): Male Stok : 20 Diskon: …. Gambar:  	Jika semua *field* terisi semua maka akan muncul pesan " data berhasil disimpan"	Jika semua *field* terisi semua maka akan muncul pesan " data berhasil disimpan"	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama produk kosong	Muncul pesan kesalahan"nama produk harus diisi"	Dapat menampilkan pesan kesalahan "nama produk	[✓] Diterima [ ] Ditolak

		harus diisi”	
Harga kosong	Muncul pesan kesalahan“harga harus diisi”	Dapat menampilkan pesan kesalahan “harga harus diisi”	[✓] Diterima [ ] Ditolak
Detil kosong	Muncul pesan kesalahan“detil harus diisi”	Dapat menampilkan pesan kesalahan “detil harus diisi”	[✓] Diterima [ ] Ditolak
Berat kosong	Muncul pesan kesalahan“berat harus diisi”	Dapat menampilkan pesan kesalahan “berat harus diisi”	[✓] Diterima [ ] Ditolak
Ukuran kosong	Muncul pesan kesalahan“ukuran harus diisi”	Dapat menampilkan pesan kesalahan “ukuran harus diisi”	[✓] Diterima [ ] Ditolak
Warna kosong	Muncul pesan kesalahan“warna harus diisi”	Dapat menampilkan pesan kesalahan “warna harus diisi”	[✓] Diterima [ ] Ditolak
Stok kosong	Muncul pesan kesalahan“stok harus diisi”	Dapat menampilkan pesan kesalahan “stok harus diisi”	[✓] Diterima [ ] Ditolak

4.4.2.12 Pengujian Pengolahan Cari Data Produk

Berikut dapat dilihat pengujian dari cari data Kategori pada tabel di bawah ini :

Tabel 4.40 Pengujian Cari Data Produk

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi cari produk Contoh: cari produk: Pth	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari produk	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari produk	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama	Tampilkan pesan	Dapat	[✓] Diterima

produk masih kosong	kesalahan "produk tidak ada"	menampilkan pesan kesalahan "produk tidak ada"	[ ] Ditolak
Data nama produk tidak ada	Tampilkan pesan kesalahan "produk tidak ada"	Dapat menampilkan pesan kesalahan "produk tidak ada"	[✓] Diterima [ ] Ditolak

4.4.2.13 Pengujian Pengolahan Tambah Data Provinsi

Berikut dapat dilihat pengujian dari tambah provinsi pada tabel di bawah ini:

Tabel 4.41 Pengujian Tambah Data Provinsi

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

4.4.2.14 Pengujian Pengolahan Ubah Data Provinsi

Berikut dapat dilihat pengujian dari ubah data provinsi pada tabel di bawah ini:

Tabel 4.42 Pengujian Ubah Data Provinsi

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Contoh data masukan provinsi Contoh: Nama Provinsi : Jawa Barat	Jika semua *field* terisi maka akan muncul pesan "data berhasil disimpan"	Jika semua *field* terisi maka akan muncul pesan "data berhasil disimpan".	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama provinsi masih kosong	Tampilkan pesan kesalahan "nama provinsi harus diisi"	Dapat menampilkan pesan kesalahan " nama provinsi harus diisi"	[✓] Diterima [ ] Ditolak
Data nama provinsi sudah ada	Tampilkan pesan kesalahan "provinsi sudah ada"	Dapat menampilkan pesan kesalahan "provinsi sudah ada"	[✓] Diterima [ ] Ditolak

4.4.2.15 Pengujian Pengolahan Cari Data Provinsi

Berikut dapat dilihat pengujian dari cari data provinsi pada tabel di bawah ini:

Tabel 4.43 Pengujian Cari Data Provinsi

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi cari provinsi Contoh: cari barang: jawa barat	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari provinsi	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari provinsi	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama provinsi masih kosong	Tampilkan pesan kesalahan "provinsi tidak ada"	Dapat menampilkan pesan kesalahan "provinsi tidak ada"	[✓] Diterima [ ] Ditolak
Data nama provinsi tidak ada	Tampilkan pesan kesalahan "provinsi tidak	Dapat menampilkan pesan kesalahan	[✓] Diterima [ ] Ditolak

	ada”	“provinsi tidak ada”	

4.4.2.16 Pengujian Pengolahan Tambah Kota

Berikut dapat dilihat pengujian dari tambah kota pada tabel di bawah ini:

Tabel 4.44 Pengujian Tambah Kota

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota : Cianjur	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
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama kota masih kosong	Tampilkan pesan kesalahan “nama kota harus diisi”	Dapat menampilkan pesan kesalahan ” nama kota harus diisi”	[✓] Diterima [ ] Ditolak
Data nama kota sudah ada	Tampilkan pesan kesalahan “kota sudah ada”	Dapat menampilkan pesan kesalahan “kota sudah ada”	[✓] Diterima [ ] Ditolak

4.4.2.17 Pengujian Pengolahan Ubah Data Kota

Berikut dapat dilihat pengujian dari ubah data kota pada tabel di bawah ini

Tabel 4.45 Pengujian Ubah Data Kota

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data kota Contoh: Nama kota: Bandung	Jika semua *field* terisi maka akan muncul pesan “data berhasil disimpan”	Jika semua *field* terisi maka akan muncul pesan “data berhasil disimpan”.	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama kota masih kosong	Tampilkan pesan kesalahan “nama kota harus diisi”	Dapat menampilkan pesan kesalahan ” nama kota harus diisi”	[✓] Diterima [ ] Ditolak
Data nama kota sudah ada	Tampilkan pesan kesalahan “kota sudah ada”	Dapat menampilkan pesan kesalahan “kota sudah ada”	[✓] Diterima [ ] Ditolak

4.4.2.18 Pengujian Pengolahan Cari Data Kota

Berikut dapat dilihat pengujian dari cari data kota pada tabel di bawah ini:

Tabel 4.46 Pengujian Cari Data Kota

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi cari kota Contoh: cari barang: jawa barat	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari kota	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses cari kota	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama kota masih kosong	Tampilkan pesan kesalahan “kota tidak ada”	Dapat menampilkan pesan kesalahan “kota tidak ada”	[✓] Diterima [ ] Ditolak
Data nama kota tidak ada	Tampilkan pesan kesalahan “kota tidak ada”	Dapat menampilkan pesan kesalahan “kota tidak ada”	[✓] Diterima [ ] Ditolak

4.4.2.19 Pengujian Pengolahan Tambah Ongkos Kirim

Berikut dapat dilihat pengujian dari tambah ongkos kirim pada tabel di bawah ini:

Tabel 4.47 Pengujian Tambah Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data ongkos kirim Contoh: Kategori	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat	[✓] Diterima [ ] Ditolak

pengiriman : regular Provinsi : Jawa Barat Kota : Cianjur Tarif: Rp.5000	melanjutkan proses tambah ongkos kirim	melanjutkan proses tambah ongkos kirim	
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Kategori pengiriman tidak dipilih	Tampilkan pesan kesalahan "silahkan pilih kategori pengiriman"	Dapat menampilkan pesan kesalahan " silahkan pilih kategori pengiriman"	[✓] Diterima [ ] Ditolak
Provinsi tidak dipilih	Tampilkan pesan kesalahan "silahkan pilih provinsi"	Dapat menampilkan pesan kesalahan " silahkan pilih provinsi"	[✓] Diterima [ ] Ditolak
Kota tidak dipilih	Tampilkan pesan kesalahan "silahkan pilih kota"	Dapat menampilkan pesan kesalahan " silahkan pilih kota"	[✓] Diterima [ ] Ditolak
Ongkos tidak diisi	Tampilkan pesan kesalahan "ongkos kirim harus diisi"	Dapat menampilkan pesan kesalahan " ongkos harus diisi"	[✓] Diterima [ ] Ditolak

4.4.2.20 Pengujian *Login* Pemilik

Berikut dapat dilihat pengujian *login* pemilik pada tabel dibawah ini:

Tabel 4.48 Pengujian *Login* Pemilik

Kasus dan Hasil Uji (DataBenar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan nama pengguna : mametBrnsn	Mengisikan data nama pengguna pada *field* nama pengguna	Dapat mengisi *field* nama pengguna dan mengisi data *login* dengan benar	[✓] Diterima [ ] Ditolak
Memasukan kata sandi : mametBrnsn	Mengisikan data kata sandi pada *field* kata sandi	Mengisikan data kata sandi pada *field* kata sandi	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan nama pengguna atau kata sandi yang tidak terdaftar	Tampilkan pesan kesalahan "Nama pengguna atau kata sandi salah, silahkan coba lagi"	Muncul pesan "Nama pengguna atau kata sandi salah, silahkan coba lagi"	[✓] Diterima [ ] Ditolak
Data nama pengguna atau kata sandi kosong	Tidak dapat *login*	Muncul pesan kesalahan "nama pengguna atau kata sandi salah, silahkan coba lagi."	[✓] Diterima [ ] Ditolak

4.4.2.21 Pengujian Tambah Pegawai

Berikut dapat dilihat pengujian tambah pegawai pada Tabel dibawah ini:

Tabel 4.49 Pengujian Tambah Pegawai

Kasus dan Hasil Uji (DataBenar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan Nama pegawai : Donny wicaksono	Mengisikan data nama pegawai pada *field* nama pegawai	Dapat mengisikan data nama pegawai pada *field* nama pegawai	[✓] Diterima [ ] Ditolak
Memasukan Nama pengguna: abokBrnsn	Mengisikan data nama pengguna pada *field* Nama pengguna	Dapat mengisikan data nama pengguna pada *field* nama pengguna	[✓] Diterima [ ] Ditolak
Memasukan kata sandi : abokBrnsn	Mengisikan data kata sandi pada *field* kata sandi	Mengisikan data kata sandi pada *field* kata sandi	[✓] Diterima [ ] Ditolak
Memasukan e-mail : abokBrnsn@gmai.com	Mengisikan data e-mail pada *field* e-mail	Mengisikan data e-mail pada *field* e-mail	[✓] Diterima [ ] Ditolak
Memasukan Alamat Pegawai: Pasirjati	Mengisikan data alamat pegawai pada *field* kata sandi	Mengisikan data alamat pegawai pada *field* alamat pegawai	[✓] Diterima [ ] Ditolak
Memasukan No Telepon: 08578492780	Mengisikan data no telepon pada *field* no telepon	Mengisikan data no telepon pada *field* no telepon pegawai	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			

Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tidak Memasukan Nama pegawai	Menampilkan pesan “nama pegawai harus diisi”	Dapat menampilkan pesan “nama pegawai harus diisi	[✓] Diterima [ ] Ditolak
Tidak memasukan nama pengguna	Menampilkan pesan kesalahan “ nama pengguna harus diisi“	Dapat menampilkan pesan kesalahan “ nama pengguna harus diisi	[✓] Diterima [ ] Ditolak
Tidak memasukan kata sandi	Menampilkan pesan kesalahan “ kata sandi harus diisi“	Dapat menampilkan pesan kesalahan kata sandi harus diisi”	[✓] Diterima [ ] Ditolak
Tidak memasukan e-mail	Menampilkan pesan kesalahan “ e-mail harus diisi“	Dapat menampilkan pesan kesalahan “ e-mail harus diisi“	[✓] Diterima [ ] Ditolak
Tidak memasukan alamat pegawai	Menampilkan pesan kesalahan “alamat pegawai harus diisi“	Dapat menampilkan pesan kesalahan “alamat pegawai harus diisi“	[✓] Diterima [ ] Ditolak
Tidak Memasukan No Telepon	Menampilkan pesan kesalahan “no telepon harus diisi“	Menampilkan pesan kesalahan “no telepon harus diisi“	[✓] Diterima [ ] Ditolak

4.4.2.22 Pengujian Ubah Pegawai

Berikut dapat dilihat pengujian ubah pegawai pada tabel dibawah ini:

Tabel 4.50 Pengujian Ubah Pegawai

Kasus dan Hasil Uji (DataBenar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan Nama pegawai : Donny wicaksono	Mengisikan data nama pegawai pada *field* nama pegawai	Dapat mengisikan data nama pegawai pada *field* nama pegawai	[✓] Diterima [ ] Ditolak
Memasukan Nama pengguna: abokBrnsn	Mengisikan data nama pengguna pada *field* Nama	Dapat mengisikan data nama pengguna	[✓] Diterima [ ] Ditolak

	pengguna	pada *field* nama pengguna	
Memasukan kata sandi : abokBrnsn	Mengisikan data kata sandi pada *field* kata sandi	Mengisikan data kata sandi pada *field* kata sandi	[✓] Diterima [ ] Ditolak
Memasukan e-mail : abokBrnsn@gmai.com	Mengisikan data e-mail pada *field* e-mail	Mengisikan data e-mail pada *field* e-mail	[✓] Diterima [ ] Ditolak
Memasukan Alamat Pegawai: Pasirjati	Mengisikan data alamat pegawai pada *field* kata sandi	Mengisikan data alamat pegawai pada *field* alamat pegawai	[✓] Diterima [ ] Ditolak
Memasukan No Telepon: 08578492780	Mengisikan data no telepon pada *field* no telepon	Mengisikan data no telepon pada *field* no telepon pegawai	[✓] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Tidak Memasukan Nama pegawai	Menampilkan pesan “nama pegawai harus diisi”	Dapat menampilkan pesan “nama pegawai harus diisi	[✓] Diterima [ ] Ditolak
Tidak memasukan nama pengguna	Menampilkan pesan kesalahan “ nama pengguna harus diisi“	Dapat menampilkan pesan kesalahan “ nama pengguna harus diisi	[✓] Diterima [ ] Ditolak
Tidak memasukan kata sandi	Menampilkan pesan kesalahan “ kata sandi harus diisi“	Dapat menampilkan pesan kesalahan kata sandi harus diisi”	[✓] Diterima [ ] Ditolak
Tidak memasukan e-mail	Menampilkan pesan kesalahan “ e-mail harus diisi“	Dapat menampilkan pesan kesalahan “ e-mail harus diisi“	[✓] Diterima [ ] Ditolak
Tidak memasukan alamat pegawai	Menampilkan pesan kesalahan “alamat pegawai harus diisi“	Dapat menampilkan pesan kesalahan “alamat pegawai harus diisi“	[✓] Diterima [ ] Ditolak
Tidak Memasukan No Telepon	Menampilkan pesan kesalahan “no telepon	Menampilkan pesan kesalahan “no telepon harus	[✓] Diterima [ ] Ditolak

	harus diisi“	diisi“	

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

4.5.1 Skenario Pengujian Beta

Pengujian beta ini dilakukan dengan cara melakukan pengujian langsung di tempat penelitian yaitu pada lingkungan Distro Branson For Life.

Penelitian di lingkungan Distro Branson For Life dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

4.5.1.1 Lingkungan Masyarakat Umum

Penelitian dilakukan di lingkungan masyarakat umum khususnya yang sudah terbiasa dengan teknologi internet, dilakukan untuk mengetahui sejauh mana sistem yang dibangun dapat memberikan kemudahan masyarakat umum untuk melakukan transaksi pemesanan produk.

Adapun beberapa poin yang diteliti adalah:

1. Kemudahan dalam bertransaksi secara *online* melalui *website e-commerce.*
2. Kemudahan dalam pembayaran transaksi pemesanan.
3. Kelengkapan informasi produk yang ditampilkan

Kuesioner ini terdiri dari empat pertanyaan (contoh kuesioner dapat dilihat pada lampiran) dengan menggunkan skala *likert* dengan skala satu sampai empat, dengan ketentuan pertanyaan dan skala sebagai berikut :

1. Apakah anda setuju *website e-commerce* di Distro Branson For Life mempermudah anda bertransaksi daripada harus datang ke Distro?

Tabel 4.51 Kuesioner Pertanyaan 1

Pertanyaan	Pilihan Jawaban	Keterangan	Responden
1	A	Sangat Setuju	6
	B	Setuju	18
	C	Cukup Setuju	6
	D	Kurang Setuju	0
	E	Tidak Setuju	0

2. Apakah anda setuju pembayaran secara *online* memudahkan dalam melakukan transaksi?

Tabel 4.52 Kuesioner Pertanyaan 3

Pertanyaan	Pilihan Jawaban	Keterangan	Responden
3	A	Sangat Setuju	6
	B	Setuju	12
	C	Cukup Setuju	6
	D	Kurang Setuju	6
	E	Tidak Setuju	0

Kuesioner tersebut dianalisis dengan membuat skor S=5, SS=4, CS=3, KS=2 dan S=1, untuk masing-masing jawaban dari pertanyaan nomor 1 dan nomor 3. Kuesioner tersebut diisi oleh 30 orang responden dari masyarakat umum. Berikut adalah perhitungan skor untuk kuesioner.

Tabel 4.53 Perhitungan Jumlah Skor

Untuk Seluruh Responden Masyarakat Umum

Jawaban	Skor	Responden	Jumlah Pertanyaan	(Skor x Pertanyaan)	Responden x (Skor x Pertanyaan)
A	5	30	2	10	300
B	4	30	2	8	240
C	3	30	2	6	180
D	2	30	2	4	120
E	1	30	2	2	60

Berdasarkan perhitungan dari jumlah skor untuk seluruh responden dari rumus Responden x (skor x pertanyaan) maka didapat jumlah skor ideal

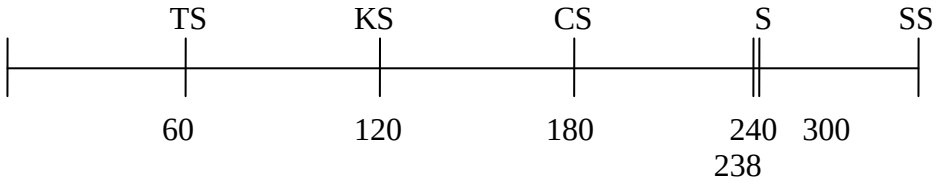
(maksimum) adalah 600. Berikut ini adalah hasil pengujian kuesioner masyarakat umum.

Tabel 4.54 Hasil Pengujian Kuesioner Beta Masyarakat Umum

Responden	Pertanyaan		Total
	No. 1	No. 3	
1	5	2	10
2	3	3	9
3	2	4	8
4	4	2	8
5	3	2	6
6	2	2	4
7	4	2	8
8	3	2	6
9	5	2	10
10	3	3	9
11	3	3	9
12	4	2	8
13	2	4	8
14	3	2	6
15	5	2	10
16	3	3	9
17	5	2	10
18	2	4	8
19	3	3	9
20	4	2	8
21	2	2	4
22	3	2	6
23	3	3	9
24	3	3	9
25	3	3	9
26	2	2	4
27	3	3	9
28	4	2	8
29	4	2	8
30	3	3	9
Total Skor			238

Setelah menghitung total skor dari masing-masing pertanyaan dari masing-masing responden, maka didapat total skor dari penjumlahan total skor masing-masing skor responden yaitu 238.

Secara kontinum dapat digambarkan sebagai berikut.



Gambar 4.1 Hasil Kuesioner

Berdasarkan data yang diperoleh skor yang didapat terletak pada posisi mendekati jawaban setuju (S), maka skor penilaian *website* ini berada diatas rata-rata skor.

3. Apakah tampilan antarmuka *website e-commerce* di Distro Branson For Life ini sudah *user friendly?*

Tabel 4.35 Kuesioner Pertanyaan 2

Pertanyaan	Pilihan Jawaban	Keterangan	Responden	Skor	Responden x Skor
2	A	Sangat Setuju	8	5	40
	B	Setuju	12	4	48
	C	Cukup Setuju	6	3	18
	D	Kurang Setuju	4	2	8
	E	Tidak Setuju	0	1	0
Total					118

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 30 = 150 (seandainya semua menjawab SB). Jumlah skor yang diperoleh dari penelitian 118. Jadi berdasarkan data tu maka tingkat penilaian setuju terhadap tampilan *website e-commerce* di distro Branson For Life itu = (118 : 150) x 100% = 79% dari yang diharapkan (100%)

4. Apakah anda setuju dengan adanya *website e-commerce* di distro Branson For Life, anda dapat mendapatkan informasi produk yang dibutuhkan?

Tabel 4.56 Kuesioner Pertanyaan 4

Pertanyaan	Pilihan Jawaban	Keterangan	Responden	Skor	Skor x Pertanyaan
4	A	Sangat Setuju	24	5	120
	B	Setuju	6	4	24
	C	Cukup Setuju	0	3	0
	D	Kurang Setuju	0	2	0
	E	Tidak Setuju	0	1	0
Total					144

Jumlah skor ideal (kriterium) untuk seluruh item = 5 x 30 = 150 (seandainya semua menjawab SS). Jumlah skor yang diperoleh dari penelitian 144. Jadi berdasarkan data tu maka tingkat penilaian baik terhadap tersampaikannya informasi produk yang mereka butuhkan itu = (144 : 150) x 100% = 96% dari yang diharapkan (100%)

4.5.1.2 Kesimpulan Pengujian *Beta* di Lingkungan Masyarakat Umum

Dari hasil pengujian *beta* yang telah dilakukan yaitu dengan pengujian perhitungan jawaban dari kuesioner yang telah dibagikan di lapangan didapat kesimpulan yaitu :

1. Website *e-commerce* di Distro Branson For Life dapat mempermudah transaksi daripada harus datang langsung ke lokasi penjualan.
2. Website *e-commerce* di Distro Branson For Life dirasa telah mampu menjadi sarana promosi dalam menginformasikan produk yang tersedia di Distro Branson For Life.

4.5.1.3 Lingkungan Pegawai

Penelitian dilakukan di lingkungan pegawai yang sudah terbiasa dengan teknologi internet, dilakukan untuk mengetahui sejauh mana sistem yang dibangun dapat memberikan kemudahan kepada pegawai untuk melakukan pengelolaan transaksi dan pengelolaan data produk dan stok.

Adapun pertanyaan-pertanyaan yang diajukan adalah sbb:

1. Pertanyaan:

Apakah *website e-commerce* yang dibangun dapat membantu pengelolaan transaksi di Distro Branson For Life?

Jawaban:

Iya, membantu, saya tidak lagi harus mencatat lagi pesanan-pesanan yang masuk, begitupun dengan pengiriman, saya tidak perlu mengubah-ubah nama dan alamat penerima karena telah langsung jadi dengan sekali klik.

2. Pertanyaan:

   Apakah *website e-commerce* di Distro Branson For Life dapat membantu promosi produk yang tersedia di Distro Branson For Life?

   Jawaban:

   Membantu-membantu, enak tinggal masukin-masukin barangnya, langsung tampil di web, tidak seperti web yang sebelumnya yang harus mengetikan koding html pada setiap produk yang akan dipromosikan.

3. Pertanyaan:

   Apakah *website e-commerce* di Distro Branson For Life dapat membantu anda dalam mengelola data dan stok produk di Distro Branson For Life?

   Jawaban:

   Membantu, data produk bisa langsung dilihat, tidak perlu lagi menghitung dulu ke laporan. Lihat laporan juga bisa langsung request dari Mamet (pemilik). Mempermudah pokoknya.

4. Pertanyaan:

   Apakah *website e-commerce* di Distro Branson For Life dapat membantu anda dalam mengelola data lainnya disamping dari data produk?

   Jawaban:

   Iya iya, membantu, gampang tinggal klik-klik, menambah atau mengubah data pun mudah karena tinggal ubah/*input* langsung di dalam web.

4.5.1.4 Kesimpulan Pengujian *Beta* di Lingkungan Pegawai

Dari hasil pengujian *beta* yang telah dilakukan yaitu dengan cara melakukan wawancara dengan pegawai di Branson For Life didapat kesimpulan yaitu :

1. Website *e-commerce* di Branson For Life dapat membantu pengelolaan transaksi.
2. Website *e-commerce* di Branson For Life dapat mambantu promosi produk yang tersedia di Distro Branson For Life.
3. Website *e-commerce* di Branson For Life dapat membantu mempermudah pengelolaan data dan stok produk, serta data-data lain di Distro Branson For Life.

4.5.1.5 Lingkungan Pemilik

Penelitian dilakukan di lingkungan pemilik yang sudah terbiasa dengan teknologi internet, dilakukan untuk mengetahui sejauh mana sistem yang dibangun dapat memberikan kemudahan kepada pemilik untuk melakukan pengelolaan laporan transaksi di Distro Branson For Life.

Adapun pertanyaan-pertanyaan yang diajukan adalah sbb:

1. Pertanyaan:

   Apakah *website e-commerce* di Distro Branson For Life dapat membantu anda dalam rekapitulasi data transaksi dan data produk di Distro Branson For Life?

   Jawaban:

   Ia, tinggal pilih bulan dan tahunnya, terus klik jadi, gampang jadinya, mudah juga liat laporannya. Tinggal *download* langsung bisa dicetak.
2. Pertanyaan:

   Apakah *website e-commerce* di Distro Branson For Life dapat membantu anda dalam mengelola pegawai di Distro Branson For Life?

   Jawaban:

   Membantu, karena bisa mengubah data pegawai dengan mudah, begitu juga jika ada penambahan pegawai di distro ini. Ya semoga dengan adanya aplikasi ini dapat membantu mengembangkan bisnis ini.

4.5.1.6 Kesimpulan Pengujian *Beta* di Lingkungan Pemilik

Dari hasil pengujian *beta* yang telah dilakukan yaitu dengan cara melakukan wawancara dengan pemilik di Branson For Life didapat kesimpulan yaitu :

1. Website *e-commerce* di Distro Branson For Life mempermudah dalam rekapitulasi data transaksi dan stok di Distro Branson For Life
2. Website *e-commerce* di Branson For Life dapat membantu dalam pengelolaan pegawai di Distro Branson For Life.