BAB IV

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

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

4.1.1 Perangkat Keras Yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam pembangunan modul *E-Commerce* untuk WordPress tercantum pada tabel sebagai berikut :

Tabel 4-1 Perangkat Keras Yang Digunakan

Perangkat	Spesifikasi
Processor	Kecepatan 2.4 GHz
RAM	1 GHz
Harddisk	40 Gb
Monitor	Resolusi 1280 x 800
VGA	256 Mb

4.1.2 Perangkat Lunak Yang Digunakan

Spesifikasi perangkat lunak yang digunakan dalam modul *E-Commerce* untuk WordPress adalah sebagai berikut:

Tabel 4-2 Perangkat Lunak Yang Digunakan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP SP 2
2	Bahasa Pemrograman	PHP
3	*DBMS*	MySQL 5.0.27
4	*Web browser*	Mozilla firefox 3.5.5
5	*Code editor*	Macromedia Dreamweaver 8
6	DFD modeler	Microsoft Visio 2007
7	Tools pendukung	Wordpress 2.9.2

4.2 Implementasi Database

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

1. Tabel wp_commentmeta

```
CREATE TABLE IF NOT EXISTS `wp_commentmeta` (
  `meta_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `comment_id` bigint(20) unsigned NOT NULL DEFAULT '0',
  `meta_key` varchar(255) DEFAULT NULL,
  `meta_value` longtext,
  PRIMARY KEY (`meta_id`),
  KEY `comment_id` (`comment_id`),
  KEY `meta_key` (`meta_key`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `wp_commentmeta`
  ADD CONSTRAINT `wp_commentmeta_ibfk_1` FOREIGN KEY
(`comment_id`) REFERENCES `wp_comments` (`comment_ID`);
```

2. Tabel wp_comments

```
CREATE TABLE IF NOT EXISTS `wp_comments` (
  `comment_ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `comment_post_ID` bigint(20) unsigned NOT NULL DEFAULT '0',
```

```
  `comment_author` tinytext NOT NULL,
  `comment_author_email` varchar(100) NOT NULL DEFAULT '',
  `comment_author_url` varchar(200) NOT NULL DEFAULT '',
  `comment_author_IP` varchar(100) NOT NULL DEFAULT '',
  `comment_date` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `comment_date_gmt` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `comment_content` text NOT NULL,
  `comment_karma` int(11) NOT NULL DEFAULT '0',
  `comment_approved` varchar(20) NOT NULL DEFAULT '1',
  `comment_agent` varchar(255) NOT NULL DEFAULT '',
  `comment_type` varchar(20) NOT NULL DEFAULT '',
  `comment_parent` bigint(20) unsigned NOT NULL DEFAULT '0',
  `user_id` bigint(20) unsigned NOT NULL DEFAULT '0',
  PRIMARY KEY (`comment_ID`),
  KEY `comment_approved` (`comment_approved`),
  KEY `comment_post_ID` (`comment_post_ID`),
  KEY `comment_approved_date_gmt`
(`comment_approved`,`comment_date_gmt`),
  KEY `comment_date_gmt` (`comment_date_gmt`),
  KEY `user_id` (`user_id`),
  KEY `comment_parent` (`comment_parent`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
ALTER TABLE `wp_comments`
  ADD CONSTRAINT `wp_comments_ibfk_1` FOREIGN KEY
(`comment_post_ID`) REFERENCES `wp_posts` (`ID`);
```

3. Tabel wp_kabkota

```
CREATE TABLE IF NOT EXISTS `wp_kabkota` (
  `id_kabkota` int(10) NOT NULL AUTO_INCREMENT,
  `provinsi_id` int(10) DEFAULT NULL,
  `nama_kabkota` varchar(50) DEFAULT NULL,
  `kabkota` varchar(20) DEFAULT NULL,
  `ongkir` int(11) DEFAULT '10000',
  PRIMARY KEY (`id_kabkota`),
  KEY `provinsi_id` (`provinsi_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `wp_kabkota`
  ADD CONSTRAINT `wp_kabkota_ibfk_1` FOREIGN KEY (`provinsi_id`)
REFERENCES `wp_provinsi` (`id_provinsi`) ON UPDATE CASCADE;
```

4. Tabel wp_links

```
    CREATE TABLE IF NOT EXISTS `wp_links` (
      `link_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
      `link_url` varchar(255) NOT NULL DEFAULT '',
      `link_name` varchar(255) NOT NULL DEFAULT '',
      `link_image` varchar(255) NOT NULL DEFAULT '',
      `link_target` varchar(25) NOT NULL DEFAULT '',
      `link_description` varchar(255) NOT NULL DEFAULT '',
      `link_visible` varchar(20) NOT NULL DEFAULT 'Y',
      `link_owner` bigint(20) unsigned NOT NULL DEFAULT '1',
      `link_rating` int(11) NOT NULL DEFAULT '0',
      `link_updated` datetime NOT NULL DEFAULT '0000-00-00
    00:00:00',
      `link_rel` varchar(255) NOT NULL DEFAULT '',
      `link_notes` mediumtext NOT NULL,
```

```
  `link_rss` varchar(255) NOT NULL DEFAULT '',
  PRIMARY KEY (`link_id`),
  KEY `link_visible` (`link_visible`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

5. Tabel wp_detail_users

```
CREATE TABLE IF NOT EXISTS `wp_detail_users` (
  `id_user` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `id_referral` bigint(20) unsigned DEFAULT '0',
  `id_kabkota` int(11) NOT NULL DEFAULT '0',
  `ktp` varchar(25) NOT NULL DEFAULT '',
  `nama` varchar(50) NOT NULL DEFAULT '',
  `alamat` varchar(100) NOT NULL DEFAULT '',
  `kodepos` varchar(6) NOT NULL DEFAULT '',
  `telp` varchar(15) NOT NULL DEFAULT '',
  `tgl_daftar` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `jml_komisi` int(11) NOT NULL DEFAULT '0',
  `sisa_komisi` int(11) NOT NULL DEFAULT '0',
  `ac` varchar(50) NOT NULL DEFAULT '',
  `bank` varchar(50) NOT NULL DEFAULT '',
  `rekening` varchar(50) NOT NULL DEFAULT '',
  `kelamin` tinyint(4) NOT NULL DEFAULT '0',
  `subdomain` varchar(50) NOT NULL DEFAULT '',
  `visit` bigint(20) NOT NULL DEFAULT '0',
  `membership` tinyint(4) NOT NULL,
  PRIMARY KEY (`id_user`),
  KEY `id_user` (`id_user`),
  KEY `id_referral` (`id_referral`),
  KEY `id_kabkota` (`id_kabkota`)
```

```
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `wp_detail_users`
  ADD CONSTRAINT `wp_detail_users_ibfk_1` FOREIGN KEY
(`id_user`) REFERENCES `wp_users` (`ID`) ON DELETE CASCADE ON
UPDATE CASCADE,
  ADD CONSTRAINT `wp_detail_users_ibfk_2` FOREIGN KEY
(`id_referral`) REFERENCES `wp_detail_users` (`id_user`) ON
DELETE SET NULL ON UPDATE CASCADE,
  ADD CONSTRAINT `wp_detail_users_ibfk_3` FOREIGN KEY
(`id_kabkota`) REFERENCES `wp_kabkota` (`id_kabkota`);
```

6. Tabel wp_options

```
CREATE TABLE IF NOT EXISTS `wp_options` (
  `option_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `blog_id` int(11) NOT NULL DEFAULT '0',
  `option_name` varchar(64) NOT NULL DEFAULT '',
  `option_value` longtext NOT NULL,
  `autoload` varchar(20) NOT NULL DEFAULT 'yes',
  PRIMARY KEY (`option_id`),
  UNIQUE KEY `option_name` (`option_name`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

7. Tabel wp_postmeta

```
CREATE TABLE IF NOT EXISTS `wp_postmeta` (
  `meta_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `post_id` bigint(20) unsigned NOT NULL DEFAULT '0',
  `meta_key` varchar(255) DEFAULT NULL,
  `meta_value` longtext,
  PRIMARY KEY (`meta_id`),
  KEY `post_id` (`post_id`),
```

```
  KEY `meta_key` (`meta_key`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
ALTER TABLE `wp_postmeta`
  ADD CONSTRAINT `wp_postmeta_ibfk_1` FOREIGN KEY (`post_id`)
REFERENCES `wp_posts` (`ID`);
```

8. Tabel wp_posts

```
CREATE TABLE IF NOT EXISTS `wp_posts` (
  `ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `post_author` bigint(20) unsigned NOT NULL DEFAULT '0',
  `post_date` datetime NOT NULL DEFAULT '0000-00-00 00:00:00',
  `post_date_gmt` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `post_content` longtext NOT NULL,
  `post_title` text NOT NULL,
  `post_excerpt` text NOT NULL,
  `post_status` varchar(20) NOT NULL DEFAULT 'publish',
  `comment_status` varchar(20) NOT NULL DEFAULT 'open',
  `ping_status` varchar(20) NOT NULL DEFAULT 'open',
  `post_password` varchar(20) NOT NULL DEFAULT '',
  `post_name` varchar(200) NOT NULL DEFAULT '',
  `to_ping` text NOT NULL,
  `pinged` text NOT NULL,
  `post_modified` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `post_modified_gmt` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `post_content_filtered` text NOT NULL,
  `post_parent` bigint(20) unsigned NOT NULL DEFAULT '0',
```

```
  `guid` varchar(255) NOT NULL DEFAULT '',
  `menu_order` int(11) NOT NULL DEFAULT '0',
  `post_type` varchar(20) NOT NULL DEFAULT 'post',
  `post_mime_type` varchar(100) NOT NULL DEFAULT '',
  `comment_count` bigint(20) NOT NULL DEFAULT '0',
  PRIMARY KEY (`ID`),
  KEY `post_name` (`post_name`),
  KEY `type_status_date`
(`post_type`,`post_status`,`post_date`,`ID`),
  KEY `post_parent` (`post_parent`),
  KEY `post_author` (`post_author`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
ALTER TABLE `wp_posts`
  ADD CONSTRAINT `wp_posts_ibfk_1` FOREIGN KEY (`post_author`)
REFERENCES `wp_users` (`ID`);
```

9. Tabel wp_provinsi

```
CREATE TABLE `wp_provinsi` (
  `id_provinsi` int(10) NOT NULL auto_increment,
  `nama_provinsi` varchar(100) default NULL,
  PRIMARY KEY  (`id_provinsi`),
  KEY `id_provinsi` (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

10. Tabel wp_terms

```
CREATE TABLE IF NOT EXISTS `wp_terms` (
  `term_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `name` varchar(200) NOT NULL DEFAULT '',
  `slug` varchar(200) NOT NULL DEFAULT '',
  `term_group` bigint(10) NOT NULL DEFAULT '0',
```

```
    PRIMARY KEY (`term_id`),
    UNIQUE KEY `slug` (`slug`),
    KEY `name` (`name`)
  ) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

11. Tabel wp_term_relationships

```
CREATE TABLE IF NOT EXISTS `wp_term_relationships` (
  `object_id` bigint(20) unsigned NOT NULL DEFAULT '0',
  `term_taxonomy_id` bigint(20) unsigned NOT NULL DEFAULT '0',
  `term_order` int(11) NOT NULL DEFAULT '0',
  PRIMARY KEY (`object_id`,`term_taxonomy_id`),
  KEY `term_taxonomy_id` (`term_taxonomy_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
ALTER TABLE `wp_term_relationships`
  ADD CONSTRAINT `wp_term_relationships_ibfk_2` FOREIGN KEY
(`term_taxonomy_id`) REFERENCES `wp_term_taxonomy` (`term_id`),
  ADD CONSTRAINT `wp_term_relationships_ibfk_1` FOREIGN KEY
(`object_id`) REFERENCES `wp_posts` (`ID`);
```

12. Tabel wp_term_taxonomy

```
CREATE TABLE IF NOT EXISTS `wp_term_taxonomy` (
  `term_taxonomy_id` bigint(20) unsigned NOT NULL
AUTO_INCREMENT,
  `term_id` bigint(20) unsigned NOT NULL DEFAULT '0',
  `taxonomy` varchar(32) NOT NULL DEFAULT '',
  `description` longtext NOT NULL,
  `parent` bigint(20) unsigned NOT NULL DEFAULT '0',
  `count` bigint(20) NOT NULL DEFAULT '0',
  PRIMARY KEY (`term_taxonomy_id`),
  UNIQUE KEY `term_id_taxonomy` (`term_id`,`taxonomy`),
```

```
  KEY `taxonomy` (`taxonomy`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8 ;
ALTER TABLE `wp_term_taxonomy`
  ADD CONSTRAINT `wp_term_taxonomy_ibfk_1` FOREIGN KEY
(`term_id`) REFERENCES `wp_terms` (`term_id`);
```

13. Tabel wp_usermeta

```
CREATE TABLE IF NOT EXISTS `wp_usermeta` (
  `umeta_id` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `user_id` bigint(20) unsigned NOT NULL DEFAULT '0',
  `meta_key` varchar(255) DEFAULT NULL,
  `meta_value` longtext,
  PRIMARY KEY (`umeta_id`),
  KEY `meta_key` (`meta_key`),
  KEY `user_id` (`user_id`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
ALTER TABLE `wp_pemesanan`
  ADD CONSTRAINT `wp_usermeta_ibfk_1` FOREIGN KEY (`user_id`)
REFERENCES `wp_ users` (`ID`) ON DELETE CASCADE ON UPDATE
CASCADE;
```

14. Tabel wp_pemesanan

```
CREATE TABLE IF NOT EXISTS `wp_pemesanan` (
  `id_pemesanan` int(11) unsigned NOT NULL AUTO_INCREMENT,
  `id_pembeli` bigint(20) unsigned NOT NULL DEFAULT '0',
  `id_referral` bigint(20) unsigned NOT NULL DEFAULT '0',
  `id_session` varchar(100) NOT NULL,
  `tgl_transaksi` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `ongkir` int(11) unsigned NOT NULL DEFAULT '0',
```

```
  `status_pembayaran` tinyint(2) NOT NULL DEFAULT '0',
  `status_pengiriman` tinyint(2) NOT NULL DEFAULT '0',
  `nama_penerima` varchar(50) NOT NULL,
  `alamat_penerima` varchar(255) NOT NULL,
  `id_kabkota` int(11) NOT NULL,
  `kodepos` varchar(8) NOT NULL,
  `telepon` varchar(20) NOT NULL,
  `bayar_dari` varchar(255) NOT NULL,
  `tujuan_bayar` varchar(255) NOT NULL,
  `tgl_bayar` date NOT NULL,
  `total_bayar` int(11) NOT NULL,
  `catatan` varchar(255) NOT NULL,
  PRIMARY KEY (`id_pemesanan`),
  KEY `id_pembeli` (`id_pembeli`),
  KEY `id_referral` (`id_referral`),
  KEY `id_kabkota` (`id_kabkota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `wp_pemesanan`
  ADD CONSTRAINT `wp_pemesanan_ibfk_1` FOREIGN KEY
(`id_pembeli`) REFERENCES `wp_detail_users` (`id_user`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `wp_pemesanan_ibfk_2` FOREIGN KEY
(`id_referral`) REFERENCES `wp_detail_users` (`id_user`) ON
UPDATE CASCADE,
  ADD CONSTRAINT `wp_pemesanan_ibfk_3` FOREIGN KEY
(`id_kabkota`) REFERENCES `wp_kabkota` (`id_kabkota`);
```

15. Tabel wp_detail_pemesanan

```
CREATE TABLE IF NOT EXISTS `wp_detail_users` (
```

```
  `id_user` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `id_referral` bigint(20) unsigned DEFAULT '0',
  `id_kabkota` int(11) NOT NULL DEFAULT '0',
  `ktp` varchar(25) NOT NULL DEFAULT '',
  `nama` varchar(50) NOT NULL DEFAULT '',
  `alamat` varchar(100) NOT NULL DEFAULT '',
  `kodepos` varchar(6) NOT NULL DEFAULT '',
  `telp` varchar(15) NOT NULL DEFAULT '',
  `tgl_daftar` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `jml_komisi` int(11) NOT NULL DEFAULT '0',
  `sisa_komisi` int(11) NOT NULL DEFAULT '0',
  `ac` varchar(50) NOT NULL DEFAULT '',
  `bank` varchar(50) NOT NULL DEFAULT '',
  `rekening` varchar(50) NOT NULL DEFAULT '',
  `kelamin` tinyint(4) NOT NULL DEFAULT '0',
  `subdomain` varchar(50) NOT NULL DEFAULT '',
  `visit` bigint(20) NOT NULL DEFAULT '0',
  `membership` tinyint(4) NOT NULL,
  PRIMARY KEY (`id_user`),
  KEY `id_user` (`id_user`),
  KEY `id_referral` (`id_referral`),
  KEY `id_kabkota` (`id_kabkota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
ALTER TABLE `wp_detail_pemesanan`
  ADD CONSTRAINT `wp_detail_pemesanan_ibfk_1` FOREIGN KEY
(`id_pemesanan`) REFERENCES `wp_pemesanan` (`id_pemesanan`)
ON DELETE CASCADE ON UPDATE CASCADE,
```

```
  ADD CONSTRAINT `wp_detail_pemesanan_ibfk_2` FOREIGN KEY
(`id_barang`) REFERENCES `wp_posts` (`ID`) ON DELETE CASCADE
ON UPDATE CASCADE;
```

16. Tabel wp_users

```
CREATE TABLE IF NOT EXISTS `wp_users` (
  `ID` bigint(20) unsigned NOT NULL AUTO_INCREMENT,
  `user_login` varchar(60) NOT NULL DEFAULT '',
  `user_pass` varchar(64) NOT NULL DEFAULT '',
  `user_nicename` varchar(50) NOT NULL DEFAULT '',
  `user_email` varchar(100) NOT NULL DEFAULT '',
  `user_url` varchar(100) NOT NULL DEFAULT '',
  `user_registered` datetime NOT NULL DEFAULT '0000-00-00
00:00:00',
  `user_activation_key` varchar(60) NOT NULL DEFAULT '',
  `user_status` int(11) NOT NULL DEFAULT '0',
  `display_name` varchar(250) NOT NULL DEFAULT '',
  PRIMARY KEY (`ID`),
  KEY `user_login_key` (`user_login`),
  KEY `user_nicename` (`user_nicename`)
) ENGINE=InnoDB  DEFAULT CHARSET=utf8;
```

4.3 Implementasi Antar Muka

Implementasi antar muka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna.

Tabel 4-3 Implementasi Antar Muka

Menu	**Deskripsi**	**Nama File**
Login administrator dan member	Digunakan sebagai halaman login member dan administrator	index.php dan wp-login.php
ganti *password* administrator dan member	Digunakan jika administrator dan member ingin merubah *password*	admin.php?page=profil
Categories (admin)	Digunakan untuk melihat, menambah, dan merubah data kategori	categories.php
Produk (admin)	Digunakan untuk melihat, menambah, dan merubah data produk	edit.php,post-new.php
Artikel Kesehatan (admin)	Digunakan untuk melihat, menambah, dan merubah data artikel	edit.php,post-new.php
Content	Digunakan untuk melihat, dan merubah data content	edit.php,post-new.php
Lihat Pemesanan	Digunakan untuk melihat data pemesanan, data detail pemesanan, dan mengubah status pemesanan.	admin.php?page=pesanan
Registrasi *member*	Digunakan untuk melakukan pendaftaran *member*	registrasi.php
Edit Profil	Digunakan untuk mengubah profil *member*	admin.php?page=profil
Produk	Digunakan untuk melihat seluruh data produk yang ada	home.php dan single1.php
Profil	Digunakan untuk melihat data profil perusahaan	single.php
Artikel Kesehatan	Digunakan untuk melihat data artikel kesehatan	Single.php
Upgrade keanggotaan	Digunakan untuk melihat Cara Upgrade keanggotaan	Single.php
Konfirmasi Pembayaran	Digunakan untuk Konfirmasi pembayaran yang telah dilakukan	Single.php
Cara Pembayaran	Digunakan untuk melihat Cara Pembayaran	Single.php
Apa itu Sistem Referrall ?	Digunakan untuk melihat pengertian referral	Single.php

Syarat & Kondisi	Digunakan untuk melihat Syarat & Kondisi	Single.php
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus detail pemesanan yang akan dilakukan	cart.php
Checkout	Digunakan menyelesaikan pesana	Checkout.php
Logout	Digunakan User (operator, administrator, *member*)Untuk dapat keluar dari aplikasi	wp-login.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 berikut menjelaskan skenario pengujian aplikasi *member*.

Tabel 4-4 Pengujian Aplikasi *Member*

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Registrasi *member*	Menampilkan *form* user baru	*Black Box*
	Isi *form* pendaftaran	*Black Box*
	Simpan data baru	*Black Box*
Login	Isi data login	*Black Box*
	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Logout	Menghapus *session user*	*Black Box*
Ganti profil Dan password	Menampilkan *form* ubah profil dan password	*Black Box*
	Ubah data *member dan atau data password*	*Black Box*
	Simpan perubahan	*Black Box*
	Verifikasi *password* lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verifikasi konfirmasi *password*	*Black Box*
Menu produk	Menampilkan data produk	*Black Box*
	Menampilkan data detail produk	*Black Box*
	Cari produk	*Black Box*
Menu profil	Menampilkan profil perusahaan	*Black Box*
Menu artikel kesehatan	Menampilkan data artikel kesehatan	*Black Box*
Menu keranjang belanja	Menampilkan data detail pemesanan	*Black Box*
	Mengubah data detail pemesanan	*Black Box*
	Menghapus data detail pemesanan	*Black Box*
Form konfirmasi	Menampilkan konfirmasi pemesanan	*Black Box*
Form pengiriman	Menampilkan form pengiriman	*Black Box*
	Isi form pengiriman	*Black Box*
Form cara pembayaran	Menampilkan cara pembayaran	*Black Box*
Menu panel member	Menampilkan data	*Black Box*

	pemesanan yang pernah dilakukan	
	Menampilkan data detail tiap pemesanan	*Black Box*

Tabel berikut menjelaskan skenario pengujian aplikasi administrator.

Tabel 4-5 Skenario Pengujian Aplikasi Administrator

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Login	Isi data login	*Black Box*
	Verifikasi *username*	*Black Box*
	Verivikasi *password*	*Black Box*
Logout	Menghapus *session user*	*Black Box*
ganti *password*	Menampilkan *form* ganti *password* operator	*Black Box*
	Verifikasi *password* lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verfikasi konfirmasi *password*	*Black Box*
pengiriman	Tampilkan data pengiriman	*Black Box*
	Tambah data pengiriman	*Black Box*
	Ubah data pengiriman	*Black Box*
provinsi	Tampilkan data provinsi	*Black Box*
	Tambah data provinsi	*Black Box*
	Ubah data provinsi	*Black Box*
kota	Tampilkan data kota	*Black Box*
	Tambah data kota	*Black Box*
	Ubah data kota	*Black Box*
	Tampilkan data ongkos kirim	*Black Box*
	Tambah data ongkos kirim	*Black Box*
	Ubah data ongkos kirim	*Black Box*
Data pemesanan	Tampilkan data pemesanan	*Black Box*
	Ubah status data pemesanan	*Black Box*

	Lihat data detail pemesanan	*Black Box*
Data content	Tampilkan data content	*Black Box*
	Ubah data content	*Black Box*
Data kategori	Tampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
Data produk	Tampilkan data produk	*Black Box*
	Tambah data produk	*Black Box*
	Ubah data produk	*Black Box*
	hapus data produk	*Black Box*

4.4.2 Kasus dan Hasil Pengujian

4.4.3 Pengujian Registrasi Member

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

Tabel 4-6 Pengujian Registrasi *Member*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik menu registrasi *member*	Tampil form registrasi *member*	Dapat menampilkan form registrasi *member*	[√] Diterima [ ] Ditolak
Memasukan data pendaftaran, contoh masukan : *Nama lengkap: irfan *Alamat : jl. Elang 1 no. 31 Pilih provinsi : jawa barat Pilih kota : bandung No telp/hp : 0856666 Jenis kelamin : pria Nama bank : BNI No. rek : 111111 Atas nama : irfan *Email :	Jika salah satu field yang bertanda bintang 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

ainkdast@gmail.com *Ulangi email : ainkdast@gmail.com *Username : ainkdast *Password : 123456 *Ulangi password : 123456 *url reseller : ainkdast			
Klik tombol simpan	Jika salah satu field yang bertanda bintang tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.Jika data masukan benar muncul pesan "Anda telah berhasil melakukan registrasi"	Tampil pesan "Anda telah berhasil melakukan registrasi" jika data masukan benar dan dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Memasukan data kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*field* ini harus diisi"	[√] Diterima [ ] Ditolak

4.4.4 Pengujian Login *Member*

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

Tabel 4-7 Pengujian login *member*

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang	Pengamatan	Kesimpulan

	diharapkan		
Memasukan username : ainkdast	Mengisikan data username pada *field* username. Jika salah maka akan muncul pesan kesalahan	Dapat mengisi *field* username dan muncul pesan kesalahanjika salah dalam pengisian data	[√] Diterima [ ] Ditolak
Memasukan *Password* : 123456	Mengisikan data *password* pada *field password*. Jika salah maka akan muncul pesan kesalahan	Dapat mengisi *field password* dan muncul pesan kesalahan	[√] Diterima [ ] Ditolak
Klik tombol login	Data *username* dan data *password* dicari	Tombol login dapat berfungsi	[√] Diterima [ ] Ditolak
Memasukan data *username* atau *password* yang tidak terdaftar	Tidak dapat login	Muncul pesan kesalahan	[√] Diterima [ ] Ditolak
mengosongkan data *email* atau *password*	Tidak dapat login	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

4.4.5 Pengujian Logout

Berikut dapat dilihat pengujian dari logout pada tabel dibawah ini:

Tabel 4-8 Pengujian Logout

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

4.4.6 Pengujian Ubah Profil Member

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

Tabel 4-9 Pengujian Ubah Profil Dan Password Member

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik ubah profil member	Menampilkan *form* ubah profil	Dapat menampilkan *form* ubah profil	[√] Diterima [ ] Ditolak
Data profil yang akan diubah	Tercantum pada text box data yang akan diubah	Dapat mengisi *field* data yang akan diubah dengan benar	[√] Diterima [ ] Ditolak
Klik ubah	Data berhasil diubah di *database*	Tampil pesan "Data berhasil diubah"	[√] Diterima [ ] Ditolak
Mengosongkan salah satu fied	Menampilkan pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan bahwa field tersebut harus terisi.	[√] Diterima [ ] Ditolak
Mengosongkan semua field	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*field* ini harus diisi"	[√] Diterima [ ] Ditolak

4.4.7 Pengujian Menu Profil

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

Tabel 4-10 Pengujian Menu Profil

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

4.4.8 Pengujian Menu Kontak

Berikut dapat dilihat pengujian dari menu berita pada tabel dibawah ini:

Tabel 4-11 Pengujian Menu Kontak

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik kontak	Menampilkan menu kontak	Tampil menampilkan menu kontak	[√] Diterima [ ] Ditolak

4.4.9 Pengujian Menu Syarat Dan Ketentuan

Berikut dapat dilihat pengujian dari menu kondisi dan ketentuan pada tabel dibawah ini:

Tabel 4-12 Pengujian Menu Syarat Dan Ketentuan

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik syarat dan ketentuan	Menampilkan menu syarat dan ketentuan	Dapat menampilkan form syarat dan ketentuan	[√] Diterima [ ] Ditolak

4.4.10 Pengujian Menu Kategori

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

Tabel 4-13 Pengujian Menu Kategori

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

4.4.11 Pengujian Menu Keranjang Belanja

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

Tabel 4-14 Pengujian Menu *Shopping Cart*

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

	dihapus terhapus dari keranjang belanja	menghapus data pada keranjang belanja	[ ] Ditolak
Klik *checkout*	Menampilkan form pengiriman	Dapat menampilkan form pengiriman	[√] Diterima [ ] Ditolak

4.4.12 Pengujian Form Checkout

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

Tabel 4-15 Pengujian *Form Checkout*

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

4.4.13 Pengujian *Form* Konfirmasi Pembayaran

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

Tabel 4-16 Pengujian *Form* Konfirmasi Pembayaran

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh Idpesanan : 1 Nama Bank : BNI Atas Nama : irfan Jumlah Transfer :	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	[√] Diterima [ ] Ditolak

100000		field tidak terisi atau salah dalam pengisian.	
Mengosongkan semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.4.14 Pengujian Menu *History* Pemesanan

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

Tabel 4-17 Pengujian menu *history* Pemesanan

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

4.4.15 Pengujian Login Administrator

Berikut dapat dilihat pengujian login administrator pada dibawah ini:

Tabel 4-18 Pengujian Login Administrator

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
username : admin	Tercantum pada text box *username*	Dapat mengisi field *username* dengan benar	[√] Diterima [ ] Ditolak
Password : 123456	Tercantum pada text box *password*	Dapat mengisi field *password*	[√] Diterima [ ] Ditolak

		dengan benar	
Klik tombol login	Data *username* dan data *password* dicari	Tombol login dapat berfungsi	[√] Diterima [ ] Ditolak
Memasukan data *username* atau *password* yang tidak terdaftar	Tidak dapat login	Muncul pesan "Tidak Sama Untuk Username Dan/Atau Password."	[√] Diterima [ ] Ditolak
Tidak memasukan data *username* atau *password*	Tidak dapat login	Dapat menampilkan pesan kesalahan "Tidak Sama Untuk Username Dan/Atau Password."	[√] Diterima [ ] Ditolak

4.4.16 Pengujian Ganti *Password* Administrator

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

Tabel 4-19 Pengujian Ganti *Password* Administrator

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik ubah *password*	Menampilkan *form* ubah *password*	Dapat menampilkan *form* ubah *password*	[√] Diterima [ ] Ditolak
Memasukan data yang benar	*Password* baru berhasil disimpan di *database*	Dapat menyimpan data *password* yang baru dengan muncul pesan " Data berhasil diubah"	[√] Diterima [ ] Ditolak
Mengosongkan salah satu *field*	Muncul pesan kesalahan *field* tersebut harus	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

	diisi	”*field* ini harus diisi” bila ada field yang belum diisi	
Memasukan *password* yang salah	muncul pesan "password salah"	Dapat menampilkan pesan kesalahan ”*Password* salah”	[√] Diterima [ ] Ditolak
Memasukan data *password* baru dan konfirmasi *password* baru yang tidak sama	Muncul pesan "password baru dan konfirmasi password harus sama"	Dapat menampilkan pesan "password baru dan konfirmasi password harus sama"	[√] Diterima [ ] Ditolak
Mengosongkan semua *field*	Muncul pesan kesalahan *field* harus diisi	Dapat menampilkan pesan kesalahan *field* harus diisi	√] Diterima [ ] Ditolak

4.4.17 Pengujian Data Provinsi

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

Tabel 4-20 Pengujian Data Provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik provinsi	Menampilkan data provinsi	Dapat menampilkan data provinsi	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah provinsi	Dapat menampilkan *form* ubah provinsi	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan	[√] Diterima [ ] Ditolak

		kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	
Klik tombol simpan	Data berhasil diubah di *database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Klik batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.18 Pengujian Tambah Provinsi

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

Tabel 4-21 Pengujian Tambah Provinsi

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik tambah provinsi	Menampilkan *form* tambah provinsi	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data provinsi Contoh: Nama Provinsi : Jawa Barat	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam	[√] Diterima [ ] Ditolak

		pengisian data, silahkan cek kembali”	
Klik tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] Diterima [ ] Ditolak
Klik batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.19 Pengujian Data Kota

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

Tabel 4-22 Pengujian Data Kota / Kabupaten

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik kota	Menampilkan data kota	Dapat menampilkan data kota	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah kota	Dapat menampilkan *form* ubah kota	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan	[√] Diterima [ ] Ditolak

		cek kembali”	
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak
Klik batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.20 Pengujian Tambah Kota / Kabupaten

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

Tabel 4-23 Pengujian Tambah Kota

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik baru	Menampilkan *form* tambah kota kabupaten	Dapat menampilkan *form* tambah kota	[√] Diterima [ ] Ditolak
Mengisi data kota Contoh: Provinsi : Jawa Barat Nama Kota : bandung	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Mengisi data ongkos kirim Contoh: Nama Kota : bandung Ongkos kirim : 10000	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	[√] Diterima [ ] Ditolak
Tidak mengisi semua	Tampilkan pesan	Dapat	[√] Diterima

field	kesalahan	menampilkan pesan kesalahan ”ada kesalahan didalam pengisian data, silahkan cek kembali”	[ ] Ditolak
Klik tombol simpan	Data kabupaten kota dan ongkos kirim akan tersimpan didalam *database*	Data kabupaten kota dan ongkos kirim akan tersimpan didalam *database*	[√] Diterima [ ] Ditolak
Klik batal	Kembali kemenu sebelumnya	Tampil menu sebelumnya	[√] Diterima [ ] Ditolak

4.4.21 Pengujian Data Content

Berikut dapat dilihat pengujian dari data content pada tabel dibawah ini:

Tabel 4-24 Pengujian Data Content

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik content	Menampilkan data content	Dapat menampilkan data content	[√] Diterima [ ] Ditolak
Klik ubah	Menampilkan *form* ubah content	Dapat menampilkan *form* content	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Default wordpress, tidak terjadi apa-apa	Default wordpress, tidak terjadi apa-apa	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Default wordpress, tidak terjadi apa-apa	Default wordpress, tidak terjadi apa-apa	[√] Diterima [ ] Ditolak
Klik tombol publish	Data berhasil diubah di*database*	Tampil pesan “Data berhasil	[√] Diterima [ ] Ditolak

		diubah”	

4.4.22 Pengujian Data Kategori

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

Tabel 4-25 Pengujian Lihat Kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik kategori	Menampilkan *form* lihat kategori	Dapat menampilkan *form* lihat kategori	[√] Diterima [ ] Ditolak
Klik edit	Menampilkan *form* ubah kategori	Dapat menampilkan *form* ubah kategori	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Default wordpress, tidak terjadi apa-apa	Default wordpress, tidak terjadi apa-apa	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Default wordpress, tidak terjadi apa-apa	Default wordpress, tidak terjadi apa-apa	[√] Diterima [ ] Ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.23 Pengujian Tambah Kategori

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

Tabel 4-26 Pengujian Tambah Kategori

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik baru	Menampilkan *form* tambah kategori	Dapat menampilkan *form* tambah kategori	[√] Diterima [ ] Ditolak
Mengisi data kategori	Jika field nama	Dapat	[√] Diterima

Contoh: Nama kategori: herbal	kategori tidak terisi atau maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	memunculkan pesan kesalahan bahwa field tersebut harus terisi	[ ] Ditolak
Tidak mengisi semua *field*	Default wordpress, tidak terjadi apa-apa	Default wordpress, tidak terjadi apa-apa	[√] Diterima [ ] Ditolak
Klik tombol add category	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam database	[√] Diterima [ ] Ditolak

4.4.24 Pengujian Data Produk

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

Tabel 4-27 Pengujian Lihat Produk

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik post	Menampilkan data produk	Dapat menampilkan data produk	[√] Diterima [ ] Ditolak
Klik edit	Menampilkan *form* ubah produk	Dapat menampilkan *form* ubah produk	[√] Diterima [ ] Ditolak
Tidak mengisi salah satu *field*	Default wordpress, tidak terjadi apa-apa	Default wordpress, tidak terjadi apa-apa	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Default wordpress, tidak terjadi apa-apa	Default wordpress, tidak terjadi apa-apa	[√] Diterima [ ] Ditolak
Klik tombol update	Data berhasil diubah di*database*	Tampil pesan “Data berhasil diubah”	[√] Diterima [ ] Ditolak

4.4.25 Pengujian Tambah Produk

Berikut dapat dilihat pengujian dari tambah produk pada tabel dibawah ini:

Tabel 4-28 Pengujian Tambah Produk

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik baru	Menampilkan *form* tambah produk	Dapat menampilkan *form* tambah produk	[√] Diterima [ ] Ditolak
Mengisi data produk Contoh: kategori: herbal Nama : NutriBurst Keterangan : lain-lain Berat : 700 Harga : 390000 Gambar :	Jika tidak mengisi salah satu mengacu ke Default wordpress, jika kategori tidak di isi maka akan di isi kategori default	Default wordpress	[√] Diterima [ ] Ditolak
Tidak mengisi semua *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak
Klik tombol publish	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database*	[√] Diterima [ ] Ditolak

4.4.26 Pengujian Data Pemesanan

Berikut dapat dilihat pengujian dari lihat pemesanan pada tabel dibawah ini:

Tabel 4-29 Pengujian Lihat Pemesanan

Kasus dan Hasil Uji Benar (Data Benar)			
Aktifitas yang dilakukan	Yang diharapkan	Pengamatan	Kesimpulan
Klik pemesanan	Menampilkan *form* pemesanan	Dapat menampilkan	[√] Diterima [ ] Ditolak

		form lihat pemesanan	
Klik link kirim pada lihat pemesanan	Ubah status pemesanan pada tabel pemesanan di *database* menjadi telah terkirim	Data status pemesanan telah diubah di *database* menjadi telah terkirim	[√] Diterima [ ] Ditolak
Klik link detail data pemesanan pengiriman	Menampilkan *form* detail pemesanan	Dapat menampilkan *form* detail pemesanan	[√] Diterima [ ] Ditolak

4.5 Kesimpulan Pengujian Alpha

Berdasarkan hasi pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan modul e-commerce untuk wordpress bebas kesalahan penulisan dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.6 Pengujian Beta

4.6.1 Skenario Pengujian Beta

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

1. Lingkungan *owner* atau pemilik

   Penelitian di lingkungan pemilik website ini dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternatif penyelesaian permasalahan yang telah dijelaskan di awal.

2. Lingkungan Masyarakat Umum

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

Adapun beberapa poin yang diteliti adalah :

a. Tampilan antar muka yang *user friendly*
b. *Website* dapat memenuhi kebutuhan informasi pelanggan
c. Memudahkan pelanggan dalam melakukan transaksi pembelian barang
d. Pendokumentasikan data *master* dan data transaksi penjualan akan lebih
e. Sistem pembayaran yang mudah digunakan.
f. *Website* menjadi sarana memperluas pangsa pasar penjualan suplemen makanan synergi.
g. Kelayakan *website* untuk dipublikasikan

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

Keterangan :

P = nilai jawaban responden tiap soal.

Q = Jumlah responden

Y = Nilai persentase

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

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

Tabel 4-30 Hasil pengujian kuesioner soal nomor 1

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
1	1	Sangat Setuju	9	90
	2	Setuju	1	10
	3	Cukup Setuju	0	0
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
	6	Tidak Setuju	0	0
	7	Sangat Tidak Setuju	0	0
Jumlah			**10**	**100**

Berdasarkan hasil prosentase diatas maka dapat disimpulkan sebanyak 9 atau 90% menyatakan sangat setuju, dan 1 atau 10% menyatakan setuju bahwa tampilan *website* cukup menarik dan mudah untuk dipahami.

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

Tabel 4-31 Hasil pengujian kuesioner soal nomor 2

Pertanyaan	**No.**	**Keterangan**	**Responden**	**Prosentase (%)**
2	1	Sangat Setuju	9	90
	2	Setuju	1	10
	3	Cukup Setuju	0	0
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
	6	Tidak Setuju	0	0
	7	Sangat Tidak Setuju	0	0
Jumlah			**10**	**100**

Berdasarkan hasil prosentase diatas maka dapat disimpulkan sebanyak 9 orang atau 90 % menyatakan sangat setuju, 1 orang atau 10% menyatakan setuju bahwa informasi yang diberikan sesuai dengan informasi yang dibutuhkan *costumer*.

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

Tabel 4-32 Hasil pengujian kuesioner soal nomor 3

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

Berdasarkan hasil prosentase diatas maka dapat disimpulkan sebanyak 2 atau 20% menyatakan sangat setuju, 8 orang atau 80 % menyatakan setuju bahwa *website* ini akan lebih memudahkan pelanggan dalam melakukan transaksi pembelian barang.

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

Tabel 4-33 Hasil pengujian kuesioner soal nomor 4

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

Berdasarkan hasil prosentase diatas maka dapat disimpulkan sebanyak 2 orang atau 20 % menyatakan sangat setuju, 7 orang atau 70 % menyatakan setuju dan 1 orang menyatakan cukup setuju bahwa dengan adanya *website* ini pihak pemilik website akan lebih mudah mengelola data master dan data transaksi penjualannya.

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

Tabel 4-34 Hasil pengujian kuesioner soal nomor 5

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
5	1	Sangat Setuju	5	50
	2	Setuju	5	50
	3	Cukup Setuju	0	0
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
	6	Tidak Setuju	0	0
	7	Sangat Tidak Setuju	0	0
Jumlah			**10**	**100**

Berdasarkan hasil prosentase diatas maka dapat disimpulkan sebanyak 5 orang atau 50 % menyatakan sangat setuju dan 5 orang atau 50% menyatakan setuju bahwa sistem pembayaran yang digunakan akan lebih memudahkan.

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

Tabel 4-35 Hasil pengujian kuesioner soal nomor 6

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
6	1	Sangat Setuju	5	50
	2	Setuju	3	30
	3	Cukup Setuju	2	20
	4	Biasa Saja	0	0
	5	Kurang Setuju	0	0
	6	Tidak Setuju	0	0
	7	Sangat Tidak Setuju	0	0
Jumlah			**10**	**100**

Berdasarkan hasil prosentase diatas maka dapat disimpulkan sebanyak 5 orang atau 50% menyatakan sangat setuju, 3 orang atau 30% menyatakan setuju dan 2 orang atau 20% menyatakan cukup setuju bahwa *website* ini akan memperluas pangsa pasar pemilik website.

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

Tabel 4-36 Hasil pengujian kuesioner soal nomor 7

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

	7	Sangat Tidak Setuju	0	0
Jumlah			**10**	**100**

Berdasarkan hasil prosentase diatas maka dapat disimpulkan sebanyak 7 orang atau 70% menyatakan sangat setuju dan 3 orang atau 30% menyatakan setuju bahwa *website* ini sudah layak untuk di-*online*-kan.

Hasil Perhitungan :

$\frac{489}{600}$ x 100 % = 81,5 %

4.6.2 Kesimpulan Pengujian Beta

Berdasarkan pengujian beta diatas, bahwa dapat diambil kesimpulan bahwa:

1. Aplikasi ini memudahkan pelanggan dalam mendapatkan informasi barang yang tersedia.
2. Pelanggan dapat melakukan transaksi pemesanan produk dengan lebih mudah.
3. Proses pembayaran yang digunakan cukup mudah digunakan bagi para pelanggan.
4. Aplikasi ini juga dapat memperluas pangsa pasar agen penjualan suplemen makanan synergi karena menjadi tidak terbatas secara geografis.