BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Bab ini merupakan tahap penerjemahan kebutuhan pembuatan aplikasi ke dalam representasi perangkat lunak sebelum penulisan kode program dimulai sesuai dengan hasil analisis yang telah dilakukan (implementasi). Implementasi yang dilakukan mencakup data-data yang digambarkan dengan tampilan. Sedangkan untuk pengujian meliputi pengujian terhadap perangkat lunak hasil implementasi.

4.1 Implementasi Sistem

Implementasi merupakan tahap menerjemahkan perancangan berdasarkan hasil analisis. Tujuan implementasi adalah untuk mengkonfirmasikan modul program perancangan pada para pelaku sistem sehingga user dapat memberi masukan kepada pengembang sistem.

4.1.1 Implementasi Perangkat Keras (*Hardware*)

Kebutuhan minimal perangkat keras *(Hardware)* yang diperlukan untuk mengimplementasikan sistem *website e-commerce* ini adalah perangkat keras komputer yang diusulkan dengan spesifikasi sebagai berikut :

1. Processor : Intel 1.8 Ghz
2. Memory : 1 GB
3. VGA : VGA Card On-Board 64 MB
4. Kapasitas *free Harddisk space* 80 GB

5. Monitor
6. Mouse
7. Keyboard

4.1.2 Implementasi Perangkat Lunak (*software*)

Adapun perangkat lunak yang digunakan untuk mengimplementasikan perangkat lunak ini adalah sebagai berikut :

a. *Microsoft Windows* XP sebagai sistem operasi
b. *Adobe Dreamweaver* CS3 sebagai media *coder*
c. Bahasa pemrograman yang digunakan yaitu PHP
d. *Software* aplikasi *Web Server* XAMPP
e. *Software* aplikasi *database MySQL*
f. *Internet Explorer, Mozilla Firefox. Google Chrome, Opera* Sebagai *Web Browser*

4.1.3 Implementasi *Database*

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

4.1.3.1 Tabel Member

Implementasi dari tabel *member* dapat dilihat pada tabel 4.1

Tabel 4.1 Implementasi Tabel Member

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_member	int(15)			No		auto_increment	
	nama_awal	char(20)	latin1_swedish_ci		No			
	nama_akhir	char(20)	latin1_swedish_ci		No			
	jenis_kelamin	char(20)	latin1_swedish_ci		No			
	email	varchar(50)	latin1_swedish_ci		No			
	password	varchar(50)	latin1_swedish_ci		No			
	id_propinsi	int(10)			No			
	id_kabupaten_kota	int(10)			No			
	alamat	varchar(100)	latin1_swedish_ci		No			
	kode_pos	int(10)			No			
	telephone	varchar(20)	latin1_swedish_ci		No			
	status	varchar(50)	latin1_swedish_ci		No			

```
CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(15) NOT NULL AUTO_INCREMENT,
  `nama_awal` char(20) NOT NULL,
  `nama_akhir` char(20) NOT NULL,
  `jenis_kelamin` char(20) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(50) NOT NULL,
  `id_propinsi` int(15) NOT NULL,
  `id_kabupaten_kota` int(15) NOT NULL,
  `alamat` varchar(100) NOT NULL,
  `kode_pos` int(10) NOT NULL,
  `telephone` int(50) NOT NULL,
  PRIMARY KEY (`id_member`)
  CONSTRAINT `member_fk1' FOREIGN KEY (`id_propinsi`)
REFERENCES `propinsi` (`id_propinsi`),
 CONSTRAINT `member_fk1' FOREIGN KEY
(`id_kabupaten_kota`) REFERENCES `kabupaten_kota`
(`id_kabupaten_kota`)
) ENGINE=InnoDB ;
```

4.1.3.2 Tabel *User*

Implementasi dari tabel *user* dapat dilihat pada tabel 4.2

Tabel 4.2 Implementasi Tabel User

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_pengguna	int(15)			No			
	username	varchar(35)	latin1_swedish_ci		No			
	password	varchar(35)	latin1_swedish_ci		No			
	status	varchar(20)	latin1_swedish_ci		No			
	level	int(2)			No			

```
CREATE TABLE IF NOT EXISTS `pengguna` (
  `id_pengguna` int(15) NOT NULL,
  `username` varchar(35) NOT NULL,
  `password` varchar(35) NOT NULL,
  `status` varchar(20) NOT NULL,
  `level` int(2) NOT NULL,
  PRIMARY KEY (`id_pengguna`)
) ENGINE=InnoDB ;
```

4.1.3.3 Tabel *Page Content*

Implementasi dari tabel *page content* dapat dilihat pada tabel 4.3

Tabel 4.3 Implementasi Tabel *Page Content*

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_page_content	int(15)			No		auto_increment	
	id_menu_content	int(15)			No			
	judul	varchar(30)	latin1_swedish_ci		No			
	content	text	latin1_swedish_ci		No			
	id_pengguna	int(15)			No			

```
   CREATE TABLE IF NOT EXISTS `page_content` (
  `id_page_content` int(15) NOT NULL AUTO_INCREMENT,
  `id_menu_content` int(15) NOT NULL,
  `judul` varchar(30) NOT NULL,
  `content` text NOT NULL,
  `id_pengguna` int(15) NOT NULL,
  PRIMARY KEY (`id_page_content`),
  CONSTRAINT `pacecont_fk1' FOREIGN KEY
(`id_menu_content`)REFERENCES `menu_content`
(`id_menu_content`),CONSTRAINT `pagecont_fk1' FOREIGN
KEY (`id_pengguna`) REFERENCES `pengguna`
(`id_pengguna`)
) ENGINE=InnoDB;
```

4.1.3.4 Tabel *Menu Content*

Implementasi dari tabel *mnu content* dapat dilihat pada tabel 4.4

Tabel 4.4 Implementasi Tabel *Menu Content*

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_menu_content	int(15)			No		auto_increment	
	nama_menu	varchar(25)	latin1_swedish_ci		No			
	meta_keyword	varchar(50)	latin1_swedish_ci		No			
	meta_description	varchar(100)	latin1_swedish_ci		No			
	id_pengguna	int(15)			No			

```
CREATE TABLE IF NOT EXISTS `menu_content` (
  `id_menu_content` int(15) NOT NULL AUTO_INCREMENT,
  `nama_menu` varchar(25) NOT NULL,
  `meta_keyword` varchar(50) NOT NULL,
  `meta_description` varchar(100) NOT NULL,
  `id_pengguna` int(15) NOT NULL,
  PRIMARY KEY (`id_menu_content`),
  CONSTRAINT `menu_fk1' FOREIGN KEY (`id_pengguna`)
REFERENCES `pengguna` (`id_pengguna`),
) ENGINE=InnoDB;
```

4.1.3.5 Tabel Produk

Implementasi dari tabel produk dapat dilihat pada tabel 4.5

Tabel 4.5 Implementasi Tabel Produk

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_produk	int(15)			No		auto_increment	
	id_category	int(15)			No			
	kode_produk	varchar(15)	latin1_swedish_ci		No			
	nama_produk	varchar(35)	latin1_swedish_ci		No			
	dimensi_produk	varchar(35)	latin1_swedish_ci		No			
	berat_produk	int(20)			No			
	deskripsi	varchar(100)	latin1_swedish_ci		No			
	stok	int(8)			No			
	harga	int(11)			No			
	gambar	text	latin1_swedish_ci		No			
	rating_produk	int(11)			No			
	tanggal	date			No			
	id_pengguna	int(15)			No			

```
CREATE TABLE IF NOT EXISTS `produk` (
  `id_produk` int(15) NOT NULL auto_increment,
  `id_category` int(15) NOT NULL,
  `nama_produk` char(35) NOT NULL,
  `deskripsi` varchar(100) NOT NULL,
  `stok` int(8) NOT NULL,
  `harga` decimal(11,2) NOT NULL,
  `gambar` text NOT NULL,
  PRIMARY KEY  (`id_produk`),
  CONSTRAINT `produk_fk' FOREIGN KEY (`id_category`)
REFERENCES `category` (`id_category`),
) ENGINE=InnoDB;
```

4.1.3.6 Tabel *Category*

Implementasi dari tabel *category* dapat dilihat pada tabel 4.6

Tabel 4.6 Implementasi Tabel Category

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_category	int(15)			No		auto_increment	
	nama_category	char(35)	latin1_swedish_ci		No			
	deskripsi	varchar(100)	latin1_swedish_ci		No			

```
CREATE TABLE IF NOT EXISTS `category` (

  `id_category` int(15) NOT NULL auto_increment,
  `nama_category` char(35) NOT NULL,
  `deskripsi` varchar(100) NOT NULL,
  PRIMARY KEY  (`id_category`)
) ENGINE=InnoDB;
```

4.1.3.7 Tabel Order

Implementasi dari tabel order dapat dilihat pada tabel 4.7

Tabel 4.7 Implementasi Tabel Order

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_order	varchar(50)	latin1_swedish_ci		No			
	tanggal_order	date			No			
	id_member	int(15)			No			
	total_produk	int(10)			No			
	diskon	varchar(200)	latin1_swedish_ci		No			
	total_harga	decimal(11,2)			No			
	id_pengiriman	varchar(50)	latin1_swedish_ci		No			
	tanggal_pesan	date			No			
	tanggal_berakhir	date			No			
	tanggal_kirim	date			No			
	tanggal_sampai	date			No			
	jenis_pembayaran	varchar(50)	latin1_swedish_ci		No			
	acak_konfi	varchar(4)	latin1_swedish_ci		No			
	status_order	varchar(35)	latin1_swedish_ci		No			
	id_pembayaran	varchar(30)	latin1_swedish_ci		No			
	keterangan	varchar(50)	latin1_swedish_ci		No			
	id_pengguna	int(15)			No			

```
CREATE TABLE IF NOT EXISTS `data_order` (
  `id_order` varchar(50) NOT NULL,
  `tanggal_order` date NOT NULL,
  `id_member` int(15) NOT NULL,
  `total_produk` int(10) NOT NULL,
  `diskon` varchar(200) NOT NULL,
  `total_harga` decimal(11,2) NOT NULL,
  `id_pengiriman` varchar(50) NOT NULL,
  `tanggal_pesan` date NOT NULL,
  `tanggal_berakhir` date NOT NULL,
  `tanggal_kirim` date NOT NULL,
  `tanggal_sampai` date NOT NULL,
  `jenis_pembayaran` varchar(50) NOT NULL,
  `status_order` varchar(35) NOT NULL,
  `id_pembayaran` int(15) NOT NULL,
  `keterangan` varchar(50) NOT NULL,
  `id_pengguna` int(15) NOT NULL,
  PRIMARY KEY (`id_order`),
  CONSTRAINT `dataorder_fk1' FOREIGN KEY
(`id_member`)REFERENCES `member` (`id_member`),
 CONSTRAINT `dataorder_fk2' FOREIGN KEY
(`id_order_alamat`) REFERENCES `order_alamat`
(`id_order_alamat`),
 CONSTRAINT `dataorder_fk3' FOREIGN KEY
(`id_pembayaran`)REFERENCES `order_pembayaran`
(`id_pembayaran`),
 CONSTRAINT `dataorder_fk4' FOREIGN KEY
(`id_pengguna`) REFERENCES `pengguna` (`id_pengguna`)
) ENGINE=InnoDB;
```

4.1.3.8 Tabel *Order* Alamat

Implementasi dari tabel order alamat dapat dilihat pada tabel 4.8

Tabel 4.8 Implementasi Tabel Order Alamat

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
☐	id_pengiriman	varchar(50)	latin1_swedish_ci		No			
☐	delive	varchar(50)	latin1_swedish_ci		No			
☐	no_resi	varchar(50)	latin1_swedish_ci		No			
☐	kepada	varchar(50)	latin1_swedish_ci		No			
☐	alamat	varchar(100)	latin1_swedish_ci		No			
☐	city	varchar(100)	latin1_swedish_ci		No			
☐	kode_pos	int(10)			No			
☐	waktu_pengiriman	varchar(10)	latin1_swedish_ci		No			
☐	biaya_pengiriman	decimal(11,2)			No			

```
CREATE TABLE IF NOT EXISTS `data_order_alamat` (
  `id_pengiriman` varchar(50) NOT NULL,
  `alamat` varchar(100) NOT NULL,
  `city` varchar(100) NOT NULL,
  `kode_pos` int(10) NOT NULL,
  `waktu_pengiriman` varchar(10) NOT NULL,
  `biaya_pengiriman` decimal(11,2) NOT NULL,
  PRIMARY KEY (`id_pengiriman`)
) ENGINE=InnoDB;
```

4.1.3.9 Tabel *Order* Produk

Implementasi dari tabel order produk dapat dilihat pada tabel 4.9

Tabel 4.9 Implementasi Tabel Order Produk

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_order_produk	int(15)			No		auto_increment	
	id_order	varchar(50)	latin1_swedish_ci		No			
	id_produk	int(15)			No			
	jumlah_produk	int(10)			No			
	total_harga_produk	decimal(11,2)			No			

```
  CREATE TABLE IF NOT EXISTS `data_order_produk` (
  `id_order_produk` int(15) NOT NULL AUTO_INCREMENT,
  `id_order` varchar(50) NOT NULL,
  `id_produk` int(15) NOT NULL,
  `jumlah_produk` int(10) NOT NULL,
  `total_harga_produk` decimal(11,2) NOT NULL,
  PRIMARY KEY (`id_order_produk`),
  CONSTRAINT `order_produk_fk1` FOREIGN KEY
(`id_order`) REFERENCES `data_order` ('id_order`),
   ADD CONSTRAINT `order_produk_fk2` FOREIGN KEY
(`id_produk`) REFERENCES `produk` ('id_produk`);
) ENGINE=InnoDB;
```

4.1.3.10 Tabel *Head Retur Produk*

Implementasi dari tabel head retur produk dapat dilihat pada tabel 4.10

Tabel 4.10 Implementasi Tabel Head Retur Produk

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_retur	varchar(50)	latin1_swedish_ci		No			
	tanggal	date			No			
	id_member	int(15)			No			
	id_order	varchar(50)	latin1_swedish_ci		No			
	konfirmasi	varchar(20)	latin1_swedish_ci		No			

```
CREATE TABLE IF NOT EXISTS `head_retur_produk` (
   `id_retur` varchar(50) NOT NULL,
   `tanggal` date NOT NULL,
   `id_member` int(15) NOT NULL,
   `id_order` varchar(50) NOT NULL,
   `konfirmasi` varchar(20) NOT NULL,
   PRIMARY KEY (`id_retur`),
   ADD CONSTRAINT `headretur_fk1` FOREIGN KEY
   (`id_member`) REFERENCES `member` (`id_member`),
   ADD CONSTRAINT `headretur_fk1` FOREIGN KEY
   (`id_order`) REFERENCES `data_order` (`id_order`)
   ) ENGINE=InnoDB;
```

4.1.3.11 Tabel Detail Retur

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.11

Tabel 4.11 Implementasi Tabel Detail Retur

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_detail_retur	int(15)			No		auto_increment	
	id_retur	varchar(50)	latin1_swedish_ci		No			
	id_produk	int(15)			No			
	id_category	int(15)			No			
	total_produk	int(20)			No			

```
  CREATE TABLE IF NOT EXISTS `detail_retur_produk` (
  `id_detail_retur` int(15) NOT NULL AUTO_INCREMENT,
  `id_retur` varchar(50) NOT NULL,
  `id_produk` int(15) NOT NULL,
  `id_category` int(15) NOT NULL,
  `total_produk` int(20) NOT NULL,
  PRIMARY KEY (`id_detail_retur`),
  CONSTRAINT `detretur_fk1` FOREIGN KEY
  (`id_retur`) REFERENCES `head_retur_produk`
(‘id_retur`),
  ADD CONSTRAINT `detretur_fk2` FOREIGN KEY
  (`id_category`) REFERENCES `category`
(‘id_category`)
) ENGINE=InnoDB;
```

4.1.3.12 Tabel Jasa pengiriman

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.12

Tabel 4.12 Implementasi Tabel Jasa Pengiriman

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_jasa_pengiriman	int(11)			No			
	nama_jasa_pengiriman	varchar(21)	latin1_swedish_ci		No			
	url_pengiriman	varchar(250)	latin1_swedish_ci		No			

```
   CREATE TABLE IF NOT EXISTS `jasa_pengiriman` (
`id_jasa_pengiriman` int(11) NOT NULL,
`nama_jasa_pengiriman` varchar(21) NOT NULL,
`url_pengiriman` varchar(250) NOT NULL,
PRIMARY KEY (`id_jasa_pengiriman`)
) ENGINE=InnoDB;
```

4.1.3.13 Tabel paket pengiriman

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.13

Tabel 4.13 Implementasi Tabel Jasa Pengiriman

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_paket_pengiriman	int(11)			No			
	id_jasa_pengiriman	int(11)			No			
	nama_paket_pengiriman	varchar(21)	latin1_swedish_ci		No			

```
CREATE TABLE IF NOT EXISTS `paket_pengiriman` (
`id_paket_pengiriman` int(11) NOT NULL,
`id_jasa_pengiriman` int(11) NOT NULL,
`nama_paket_pengiriman` varchar(21) NOT NULL,
PRIMARY KEY (`id_paket_pengiriman`),
CONSTRAINT `paket_pengiriman_ibfk_1` FOREIGN KEY
(`id_jasa_pengiriman`) REFERENCES `jasa_pengiriman`
(`id_jasa_pengiriman`)
) ENGINE=InnoDB;
```

4.1.3.14 Tabel ongkos kirim

Implementasi dari tabel detail retur dapat dilihat pada tabel Tabel 4.14 Tabel

Tabel 4.14 Implementasi Tabel Ongkos Kirim

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_ongkos_kirin	int(11)			No			
	id_paket_pengiriman	int(11)			No			
	id_kota	int(11)			No			
	ongkos	double			No			

```
CREATE TABLE IF NOT EXISTS `ongkos_kirim` (
`id_ongkos_kirim` int(11) NOT NULL,
`id_paket_pengiriman` int(11) NOT NULL,
`id_kota` int(11) NOT NULL,
`ongkos` double NOT NULL,
 PRIMARY KEY (`id_ongkos_kirim`),
 CONSTRAINT `ongkos_kirim_ibfk_2` FOREIGN KEY (`id_kota`)
REFERENCES `kota` (`id_kota`),
 ADD CONSTRAINT `ongkos_kirim_ibfk_3` FOREIGN KEY
(`id_paket_pengiriman`) REFERENCES `paket_pengiriman`
(`id_paket_pengiriman`)
) ENGINE=InnoDB;
```

4.1.3.15 Tabel komentar

Implementasi dari tabel detail retur dapat dilihat pada tabel Tabel 4.15

Tabel 4.15 Implementasi Tabel Komentar

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_komentar	int(11)			No		auto_increment	
	tanggal	varchar(21)	latin1_swedish_ci		No			
	isi	text	latin1_swedish_ci		No			
	ditampilkan	varchar(6)	latin1_swedish_ci		No			
	id_member	int(15)			No			
	jenis	int(30)			No			
	id_produk	int(15)			No			

```
CREATE TABLE `komentar` (
`id_komentar` int(11) NOT NULL AUTO_INCREMENT,
`tanggal` varchar(21) NOT NULL,
`isi` text NOT NULL,
`ditampilkan` varchar(6) NOT NULL, `
id_member` int(15) NOT NULL,
`jenis` varchar(30) NOT NULL,
`id_produk` varchar(15) NOT NULL,
PRIMARY KEY (`id_komentar`),
CONSTRAINT `komentar_ibfk_3` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`),
CONSTRAINT `komentar_ibfk_2` FOREIGN KEY (`id_produk`)
REFERENCES `produk` (`id_produk`);
) ENGINE=InnoDB;
```

4.1.3.16 Tabel rating

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.16

Tabel 4.16 Implementasi Tabel rating

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_rating	int(11)			No		auto_increment	
	total_nilai	int(11)			No			
	total_penilai	int(11)			No			
	id_produk	int(15)			No			
	id_member	int(15)			No			

```
CREATE TABLE `rating`
( `id_rating` int(11) NOT NULL AUTO_INCREMENT,
`total_nilai` int(11) NOT NULL,
`total_penilai` int(11) NOT NULL,
`id_produk` int(15) NOT NULL,
`id_member` int(15) NOT NULL,
PRIMARY KEY (`id_rating`),
CONSTRAINT `rating_ibfk_1` FOREIGN KEY (`id_produk`)
REFERENCES `produk` (`id_produk`),
CONSTRAINT `rating_ibfk_2` FOREIGN KEY (`id_member`)
REFERENCES `member` (`id_member`);
) ENGINE=InnoDB;
```

4.1.3.17 Tabel kota

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.17

Tabel 4.17 Implementasi Tabel Kabupaten Kota

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_kota	int(11)			No		auto_increment	
	nama_kota	varchar(21)	latin1_swedish_ci		No			
	ongkoskirim	double			No			
	ditampilkan	varchar(6)	latin1_swedish_ci		No			
	id_propinsi	int(11)			No			

```
CREATE TABLE `kota` (
`id_kota` int(11) NOT NULL AUTO_INCREMENT,
`nama_kota` varchar(21) NOT NULL,
`ongkoskirim` double NOT NULL,
`ditampilkan` varchar(6) NOT NULL, `
id_propinsi` int(11) NOT NULL,
PRIMARY KEY (`id_kota`),
CONSTRAINT `kota_ibfk_1` FOREIGN KEY (`id_propinsi`)
REFERENCES `propinsi` (`id_propinsi`);
) ENGINE=InnoDB;
```

4.1.3.18 Tabel Propinsi

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.18

Tabel 4.18 Implementasi Tabel Propinsi

	Field	Type	Collation	Attributes	Null	Default	Extra	Action
	id_propinsi	int(11)			No		auto_increment	
	nama_propinsi	varchar(21)	latin1_swedish_ci		No			
	ditampilkan	int(6)			No			

```
CREATE TABLE `propinsi` (
`id_propinsi` int(11) NOT NULL AUTO_INCREMENT,
`nama_propinsi` varchar(21) NOT NULL,
`ditampilkan` varchar(6) NOT NULL, PRIMARY KEY (`id_propinsi`)
) ENGINE=InnoDB ;
```

4.1.4 Implementasi Antar Muka

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

Tabel 4.19 Implementasi Antar Muka Pengunjung Dan Member

No	Menu	Deskripsi	Nama *File*
1	Daftar member baru	Digunakan oleh pengunjung untuk melakukan pendaftaran menjadi *customer/member* baru	registrasi.php
2	Login member	Digunakan sebagai halaman login member	login.php

3	Lupa password	Digunakan apabila member lupa password yang dimiliki	index.php
4	Beranda	Digunakan sebagai tampilan awal member	index2.php
5	Ubah data pribadi	Digunakan untuk melihat dan mengubah profil member	pribadi.php
6	Ubah password	Digunakan untuk merubah password bagi member	password.php
7	Pembelian	Digunakan untuk melihat history pembelian pada member	order_data.php
9	Konfirmasi Pembayaran	Digunakan untuk melihat mengisi konfirmasi data pembayaran produk yang dipesan	confirm_payment.php
10	Pelacakan pengiriman	Digunakan untuk pelacakan pengiriman barang pada jasa pengiriman yang telah ditentukan	tracking.php
11	Retur produk	Digunakan oleh member untuk melakukan retur barang	retur.php
12	Keranjang Belanja	Digunakan untuk melihat, merubah, menghapus daftar pesanan yang akan dilakukan	cart.php

13	Logout	Digunakan member untuk keluar dari akunnya	logout.php
14	Cara pembelian	Digunakan untuk penjelasan tata cara pembelian	menu.php
15	Profil perusahaan	Digunakan untuk menampilkan profil dan alamat perusahaan	menu2.php

Tabel 4.20 Implementasi Antar Muka Administrator

No	Menu	Deskripsi	Nama *File*
1	*Login*	Digunakan sebagai halaman login admin	login.php
2	*Home* Admin	Digunakan sebagai halaman beranda admin dan terdapat beberapa menu diantaranya menu data *member* dan data *user*.	index2.php
3	*member*	Halaman yang digunakan untuk menampilkan dan menghapus data *member*	view_member.php
4	Pengguna	Digunakan untuk melihat, menambah, dan aktivasi data *operator*	view_user.php
5	Backup dan	Digunakan untuk membuat cadangan	backup.php

	Restore	*database* dan	
6	Logout	Digunakan admin untuk keluar dari halaman *administrator*	logout.php

Tabel 4.21 Implementasi Antar Muka *Operator*

No	Menu	Deskripsi	Nama *File*
1	Login	Digunakan sebagai halaman login operator	logout.php
2	Kategori Produk	Digunakan untuk mengubah, menambah, atau menghapus kategori produk	view_category.php
3	Produk	Digunakan untuk mengubah, menambah, menghapus, membuat laporan daftar produk, atau mengirim update barang baru ke member	view_produk.php
4	Data Pemesanan	Digunakan admin untuk melihat dan menghapus history penjualan	detail_order.php
5	Data Pengiriman	Digunakan untuk melihat, mengubah atau menghapus daftar pengiriman barang dan membuat laporan penjualan	pengiriman_order.php
6	Data Retur Produk	Digunakan untuk melihat daftar barang yang diretur oleh konsumen	retur.php

7	Data Peringatan Stok	Digunakan untuk melihat stok barang habis yang dipesan oleh *member*	edit_alarm.php
	Data Propinsi	Digunakan untuk melihat atau menambah data propinsi	propinsi.php
	Data Kabupaten Kota	Digunakan untuk melihat atau menambah data kota atau kabupaten	kota.php
	Data Jasa Pengiriman	Digunakan untuk melihat, merubah atau menambah data jasa pengiriman	jaskir.php
	Data Paket Pengiriman	Digunakan untuk melihat, merubah atau menambah data paket pengiriman	paket.php
	Data Ongkos Kirim	Digunakan untuk melihat, merubah atau menambah data ongkos kirim	ongkir.php
7	Konten Menu	Digunakan untuk mengubah, menambah atau menghapus daftar menu pada halaman utama member dan pengunjung	Menu.php
8	Konten Halaman	Digunakan untuk mengubah, menambah atau menghapus isi menu pada halaman utama member dan pengunjung	Page_menu.php
9	Komentar	Digunakan untuk melihat dan mengelola data komentar pada barang	Komentar.php
10	Logout	Digunakan *operator* untuk keluar dari	Logout.php

		halaman *operator*	

4.2 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan-kesalahan dan kekurangan-kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kreteria 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.

4.2.1 Pengujian *Black Box*

Untuk menentukan pengujian *black box* ini dibagi menjadi tiga, yaitu skenario pengujian aplikasi *member*, *administrator*, dan *operator*.

4.2.1.1 Skenario

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi *member* maka dapat dilihat pada tabel 4.22 sebagai berikut :

Tabel 4.22 Skenario Pengujian Aplikasi *Member*

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Isi data *login*	*Black Box*
	Verifikasi *e-mail*	*Black Box*
	Verifikasi *password*	*Black Box*
Lupa *Password*	Menampilkan *form* lupa password	*Black Box*
Daftar *member* baru	Menampilkan *form* daftar *member*	*Black Box*
	Isi *form* daftar *member*	*Black Box*
	Simpan data *member* baru	*Black Box*
Ubah Data Pribadi	Menampilkan *form* ganti data profil	*Black Box*
	Ubah data profil	*Black Box*
	Simpan perubahan	*Black Box*
Ubah *Password*	Menampilkan form ganti *password*	*Black Box*
	Isi *form* ganti *password*	*Black Box*
	Simpan data baru	*Black Box*
Pembelian	Menampilkan daftar pemesanan yang telah dilakukan	*Black Box*
Konfirmasi Pembayaran	Menampilkan form konfirmasi pembayaran	*Black Box*
	Isi form konfirmasi pembayaran	*Black Box*
	Simpan data pembayaran	*Black Box*
Keranjang belanja	Menampilkan daftar belanja	*Black Box*

	Mengubah daftar belanja	*Black Box*
	Menghapus daftar belanja	*Black Box*
	Menampilkan konfirmasi transaksi	*Black Box*
Retur produk	Menampilkan pesanan yang akan diretur	*Black Box*
Pelacakan pengiriman	Menampilkan *form* pelacakan pengiriman	*Black Box*
	isi *form* pelacakan pengiriman	*Black Box*
Keluar	Menghapus *session member*	*Black Box*

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi *administrator* maka dapat dilihat pada tabel 4.23 sebagai berikut :

Tabel 4.23 Skenario Pengujian Aplikasi *Administrator*

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Login	Isi data *login*	*Black Box*
Beranda	Menampilkan data beranda *administrator*	*Black Box*
Member	Menampilkan data *member*	*Black Box*
	Menghapus data *member*	*Black Box*
Pengguna	Menampilkan data operator	*Black Box*
	Ubah status operator	*Black Box*
	Menambah operator	*Black Box*
Logout	Menghapus *session administrator*	*Black Box*

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi *operator* maka dapat dilihat pada tabel 4.24 sebagai berikut :

Tabel 4.24 Skenario Pengujian Aplikasi *Operator*

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Login	Isi data *login*	*Black Box*
Beranda	Menampilkan data beranda *administrator*	*Black Box*
Kategori Produk	Menampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
	Hapus data kategori	*Black Box*
Produk	Menampilkan data barang	*Black Box*
	Tambah data barang	*Black Box*
	Cetak data barang	*Black Box*
	Ubah data barang	*Black Box*
	Hapus data barang	*Black Box*
	Kirim data barang	*Black Box*
Data Pemesanan	Menampilkan data pesanan	*Black Box*
	Hapus data pesanan	*Black Box*
Data Pengiriman	Menampilkan data riwayat pengiriman	*Black Box*
	Menghapus data pengiriman	*Black Box*

	Ubah data pengiriman	*Black Box*
	Cetak data pengiriman	*Black Box*
Data retur produk	Menampilkan data retur produk	*Black Box*
	Konfirmasi permintaan retur	*Black Box*
Data peringatan stok	Menampilkan data barang stok habis yang dipesan	*Black Box*
	Isi status stok barang	*Black Box*
Konten Menu	Menampilkan menu halaman utama pengunjung dan *member*	*Black Box*
	Mengubah menu halaman utama pengunjung dan *member*	*Black Box*
	Menghapus menu halaman utama pengunjung dan *member*	*Black Box*
	menambah menu halaman utama pengunjung dan *member*	*Black Box*
Konten Halaman	Menampilkan isi menu halaman utama pengunjung dan *member*	*Black Box*
	Menghapus isi menu halaman utama pengunjung dan *member*	*Black Box*
	Mengubah isi menu halaman utama pengunjung dan *member*	*Black Box*

	Menambah isi menu halaman utama pengunjung dan *member*	*Black Box*
Komentar	Menampilkan komentar barang	*Black Box*
Backup Retore	Memilih daftar database yang akan di backup	*Black Box*
	Mencari database yang telah dibackup untuk direstore	*Black Box*
Logout	Menghapus *session administrator*	*Black Box*

4.2.1.2 Kasus Dan Hasil Pengujian

4.2.1.2.1 Pengujian *Login Member*

Untuk penjelasan lebih lanjut tentang pengujian data benar *login member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.25 Pengujian Data Benar *Login Member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Email : ksatryo_perdana@yahoo.com	Field data *email* terisi	Field data *email* terisi	[√] Diterima [ ] Ditolak
Memasukan password :12345	Field data password terisi	Field data password terisi	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah *login member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.26 Pengujian Data Salah *Login Member*

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
memasukan data *email* dan *password* yang tidak sesuai	Menampilkan pesan kesalahan "ERROR: Email atau password salah"	Dapat menampilkan pesan kesalahan "ERROR : Email atau password salah"	[√] Diterima [ ] Ditolak
Memasukan data *email* atau *password* yang tidak terdaftar	Menampilkan pesan kesalahan "Email atau password salah"	Dapat menampilkan pesan kesalahan "Email atau password salah"	[√] Diterima [ ] Ditolak
Memasukan data email yang tidak valid, contoh : *Email* : Unyu_keren.com	Menampilkan pesan kesalahan "Email tidak valid"	Dapat menampilkan pesan kesalahan "Email tidak valid"	[√] Diterima [ ] Ditolak
Mengosongkan	Menampilkan	Dapat	[√] Diterima

data *email* dan *password*	pesan kesalahan "Email atau password tidak lengkap"	menampilkan pesan kesalahan "Email dan password tidak lengkap"	[ ] Ditolak

4.2.1.2.2 Pengujian Lupa *Password Member*

Untuk penjelasan lebih lanjut tentang pengujian data benar lupa *password member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.27 Pengujian Data Benar Lupa *Password Member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Email : ksatryo_perdana@yahoo.com	Data *email* dan anti spam yang diinputkan dikirim dan tampil pesan konfirmasi *password* baru telah dikirim ke *email* anda.	Setelah menginputkan *email*, anti spam, dan menekan tombol *reset password* saya, maka tampil pesan konfirmasi *password* telah	[√] Diterima [ ] Ditolak

		dikirim ke *email* anda	

Untuk penjelasan lebih lanjut tentang pengujian data salah lupa *password member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.28 Pengujian Data Salah Lupa *Password Member*

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Format *email* tidak valid, contoh : *E-mail* : Unyu_keren.com	Menampilkan pesan kesalahan "email tidak valid"	Menampilkan pesan kesalahan "email tidak valid"	[√] Diterima [ ] Ditolak
Email tidak ditemukan dalam *database*	Menampilkan pesan kesalahan "ERROR:Email belum terdaftar"	Dapat menampilkan pesan kesalahan "ERROR:Email belum terdaftar"	[√] Diterima [ ] Ditolak

4.2.1.2.3 Pengujian Daftar *Member*

Untuk penjelasan lebih lanjut tentang pengujian data benar daftar *member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.29 Pengujian Data Benar Daftar *Member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data daftar *member* baru, contoh : Nama Awal : Dimaz Nama Akhir : Faily Jenis Kelamin : Laki-Laki *Email* : dimaz.perdana@gmail.com *Password* : dimazdua Confirm *Password* : dimazdua City / Kota : Bandung Alamat : Jl. Tuisda 39 Kode Pos : 40134 No. Telp/Hp : 08562175289	Jika data masukan benar maka akan muncul pesan "SELAMAT! anda telah menjadi member, silahkan cek email untuk aktifasi akun anda"	Data yang dimasukan benar dan muncul pesan "SELAMAT! anda telah menjadi member, silahkan cek email untuk aktifasi akun anda"	[√] Diterima [ ] Ditolak

Verifikasi Code : 978GL			

Untuk penjelasan lebih lanjut tentang pengujian data salah daftar *member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.30 Pengujian Data Salah Daftar *Member*

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data kosong	Menampilkan pesan kesalahan "Email harus diisi", "password harus diisi", "konfirmasi password harus diisi", "nama harus diisi", "alamat diisi", "propinsi dipilih", "kota dipilih", "kode pos diisi"	Dapat menampilkan pesan kesalahan "Email harus diisi", "password harus diisi", "konfirmasi password harus diisi", "nama harus diisi", "alamat diisi", "propinsi dipilih", "kota dipilih", "kode pos diisi"	[√] Diterima [ ] Ditolak

Mengisi email yang sudah terdaftar	Menampilkan pesan kesalahan "Registrasi Gagal. Email yang anda masukan sudah terdaftar"	Dapat menampilkan pesan kesalahan "Registrasi Gagal. Email yang anda masukan sudah terdaftar"	[√] Diterima [ ] Ditolak
Mengisi email dengan format tidak valid, contoh : Unyu_abies.com	Menampilkan pesan kesalahan "Email tidak valid"	Dapat menampilkan pesan kesalahan "Email tidak valid"	[√] Diterima [ ] Ditolak
Mengisi konfirmasi password yang berbeda dengan password	Menampilkan pesan kesalahan "Password tidak sesuai"	Dapat menampilkan pesan kesalahan "Password tidak sesuai"	[√] Diterima [ ] Ditolak
Format kode pos tidak sesuai, contoh : Kode Pos : abcdef	Menampilkan pesan kesalahan "Password salah"	Dapat menampilkan pesan kesalahan "Kode pos salah"	[√] Diterima [ ] Ditolak

4.2.1.2.4 Pengujian Ubah *Password Member*

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah *password member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.31 Pengujian Data Benar Ubah *Password Member* asd

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data ganti password dengan benar, contoh : Password Lama : dimazdua Password Baru : dimaztiga Confirm Password Baru : dimaztiga	*Member mengisi data dengan benar data yang diisi*	Dapat mengisi data dengan benar pada data yang akan diubah oleh *member*	[√] Diterima [ ] Ditolak
Tekan tombol ubah dengan memasukan data dengan benar pada form ubah *password*	*Password* baru berhasil disimpan di *database* dan dapat	Dapat menyimpan data *password* yang baru ke *database* dan menampilkan	[√] Diterima [ ] Ditolak

	menampilkan pesan "*Password* telah diubah"	pesan "*Password* telah diubah	

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah *password member* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.32 Pengujian Data Salah Ubah *Password Member*

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Password Lama : dimazdua Password Baru : dimaztiga Confirm Password Baru : dimaztiga23	Data *password* yang diisi tidak sesuai dengan yang diinginkan	Setelah menginputkan *password* yang tidak sesuai dengan yang diharapkan, maka tampil pesan kesalahan	[√] Diterima [ ] Ditolak
Data *password* tidak diisi	Menampilkan pesan kesalahan "password harus diisi"	Dapat menampilkan pesan kesalahan "password harus diisi"	[√] Diterima [ ] Ditolak
Data *password* tidak sesuai dengan *database*	Menampikan pesan kesalahan "password salah"	Dapat menampilkan pesan kesalahan "Password salah"	[√] Diterima [ ] Ditolak

Data *password* baru dan data ulangi tidak sama	Menampilkan pesan kesalahan "*Password* tidak sesuai"	Dapat menampilkan pesan kesalahan "Password tidak sesuai"	[√] Diterima [ ] Ditolak

4.2.1.2.5 Pengujian Keranjang Belanja

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah *keranjang* belanja maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.33 Pengujian Data Benar Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Jumlah barang yang dimasukan ke keranjang belanja tidak melebihi stok barang. Missal : 13 (stok barang tersedia 20 barang)	Jumlah barang yang dibeli akan masuk ke tas belanja	Jumlah barang yang dibeli sesuai dengan yang yang diinginkan dan stok barang yang ada cukup	[√] Diterima [ ] Ditolak
Memasukan data jumlah barang pada jumlah barang yang dipilih.	Tercantum pada *text box* jumlah pada *record*	Dapat tercantum pada *text box* pada *record* barang yang	[√] Diterima [ ] Ditolak

Missal : 2	barang yang dipilih	dipilih	

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah *keranjang* belanja maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.34 Pengujian Data Salah Keranjang Belanja

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Jumlah barang yang dimasukan ke keranjang belanja melebihi stok barang. Missal : 23 (stok barang tersedia 20 barang)	Menampilkan pesan kesalahan "Jumlah barang yang diminta tidak cukup"	Dapat menampilkan pesan kesalahan "Jumlah barang yang diminta tidak cukup"	[√] Diterima [ ] Ditolak
Mengisi jumlah 0 atau minus pada jumlah barang yang dipilih	Menampilkan pesan kesalahan "jumlah barang yang dipesan tidak boleh	Dapat menampilkan pesan kesalahan "jumlah barang yang dipesan tidak boleh kurang dari 1"	[√] Diterima [ ] Ditolak

	kurang dari 1”		

4.2.1.2.6 Pengujian Form Ubah Data Pribadi

Untuk penjelasan lebih lanjut tentang pengujian data benar ubah alamat pengiriman maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.35 Pengujian Data Benar Form Ubah Alamat Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama Penerima: Dimaz Alamat : Jl. Dipati Ukur Kota : Bandung Kode Pos : 40134	*Member* mengisi data dengan benar sesuai data yang tersedia	Dapat mengisi data dengan benar pada data yang akan diubah	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah ubah alamat pengiriman maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.36 Pengujian Data Salah Form Ubah Alamat Pengiriman

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data kosong	Menampilkan pesan kesalahan "nama harus diisi", "alamat diisi", "propinsi dipilih", "kota harus diisi", "kode pos diisi"	Menampilkan pesan kesalahan "nama harus diisi", "alamat diisi", "propinsi dipilih", "kota harus diisi", "kode pos diisi"	[√] Diterima [ ] Ditolak
Memasukan tanggal retur barang	Tanggal yang dipilih tidak sesuai kesepakatan, dan dapat menampilkan pesan kesalahan "gagal"	Dapat menampilkan pesan kesalahan "gagal" saat daftar tanggal retur tidak sesuai dengan yang telah disepakati	[√] Diterima [ ] Ditolak

4.2.1.2.7 Pengujian *Login Administrator* Dan *Operator*

Untuk penjelasan lebih lanjut tentang pengujian data benar *login Administrator* dan *operator* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.37 Pengujian Data Benar *Login Administrator* Dan *Operator*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan *username* : dimaz *Password* : dimaz	Memasukan data *login* dengan benar	Dapat memasukan data *login* dengan benar	[√] Diterima [ ] Ditolak
Tekan tombol *login* dengan memasukan data *login* dengan benar	Jika *username* dan *password* sesuai dengan di *database* maka petugas dapat masuk ke menu utama petugas	*Username* dan *password* sesuai dengan di *database* maka petugas dapat masuk ke menu utama petugas	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah *login Administrator* dan *operator* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.38 Pengujian Data Salah *Login Administrator* Dan *Operator*

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
memasukan data *username* dan *password* yang tidak sesuai	Menampilkan pesan kesalahan "ERROR: username atau password salah"	Dapat menampilkan pesan kesalahan "ERROR : username atau password salah"	[√] Diterima [ ] Ditolak
Memasukan data *username* atau *password* yang tidak terdaftar	Menampilkan pesan kesalahan "username atau password salah"	Dapat menampilkan pesan kesalahan "username atau password salah"	[√] Diterima [ ] Ditolak
Mengosongkan data *username* dan *password*	Menampilkan pesan kesalahan "username atau password tidak lengkap"	Dapat menampilkan pesan kesalahan "Email dan password tidak lengkap"	[√] Diterima [ ] Ditolak

4.2.1.2.8 Pengujian Tambah *User*

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah *user* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.39 Pengujian Data Benar Tambah User

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi *username* dan *password*, contoh : *Username : Operator* *Password* : *operator*	Mengisi data *operator* dengan lengkap	Dapat mengisi data dengan benar	[√] Diterima [ ] Ditolak
Tekan tombol simpan	Menampilkan pesan "Data Berhasil Dimasukan"	Dapat menampilkan pesan "Data Berhasil Dimasukan"	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah *user* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.40 Pengujian Data Salah Salah Tambah User

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi data *operator* dengan lengkap	Menampilkan pesan kesalahan "Data Tidak Boleh Kosong"	Dapat menampilkan pesan kesalahan "Data Tidak Boleh Kosong"	[√] Diterima [ ] Ditolak

4.2.1.2.9 Pengujian Tambah Konten Menu

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah *menu content* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.41 Pengujian Data Benar Tambah Konten Menu

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi Data *Menu content*, contoh : Nama Menu : *Profil Perusahaan* *Meta Keyword* : Profil	Mengisi data menu content dengan lengkap	Dapat memasukan data menu content dengan lengkap	[√] Diterima [ ] Ditolak

Perusahaan *Meta Description* : Profil Perusahaan			
Menekan tombol Simpan	Menampilkan Pesan "Data Berhasil Dimasukan"	Dapat Menampilkan Pesan "Data Berhasil Dimasukan"	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah menu content maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.42 Pengujian Data Salah Tambah Konten Menu

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi Data *Menu content* dengan lengkap	Menampilkan pesan kesalahan "Data Tidak Boleh Kosong"	Dapat menampilkan pesan kesalahan "Data Tidak Boleh Kosong"	[√] Diterima [ ] Ditolak

4.2.1.2.10 Pengujian Tambah Konten Halaman

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah *page content* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.43 Pengujian Data Benar Tambah Konten Halaman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi Data *Page content* dengan lengkap	Mengisi data *page content* dengan lengkap	Dapat memasukan data *page content* dengan lengkap	[√] Diterima [ ] Ditolak
Menekan tombol Simpan	Menampilkan Pesan "Data Berhasil Dimasukan"	Dapat Menampilkan Pesan "Data Berhasil Dimasukan"	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data Salah tambah *page content* maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.44 Pengujian Data Salah Tambah Konten Halaman

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi Data *Page Content* dengan lengkap	Menampilkan pesan kesalahan "Data Tidak	Dapat menampilkan pesan kesalahan "Data Tidak Boleh	[√] Diterima [ ] Ditolak

	Boleh Kosong”	Kosong”	

4.2.1.2.11 Pengujian Tambah Kategori Produk

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah Kategori produk maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.45 Pengujian Data Benar Kategori Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan Nama kategori produk, contoh : Nama Category : Bamboo craft	Mengisi data nama kategori produk	Dapat mengisi data nama kategori produk	[√] Diterima [ ] Ditolak
Mengisi deskripsi kategori, contoh : Barang kerajinan ini terbuat dari bambu asli	Mengisi data deskripsi kategori produk	Dapat mengisi data deskripsi kategori produk	[√] Diterima [ ] Ditolak
Menekan tombol Simpan	Menampilkan Pesan “Data Berhasil	Dapat Menampilkan Pesan “Data Berhasil	[√] Diterima [ ] Ditolak

	Dimasukan”	Dimasukan”	

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah Kategori produk maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.46 Pengujian Data Salah Kategori Produk

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi Data kategori produk dengan lengkap	Menampilkan pesan kesalahan “Data Tidak Boleh Kosong”	Dapat menampilkan pesan kesalahan “Data Tidak Boleh Kosong”	[√] Diterima [ ] Ditolak

4.2.1.2.12 Pengujian Tambah Produk

Untuk penjelasan lebih lanjut tentang pengujian data benar tambah produk maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.47 Pengujian Data Benar Tambah Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi kategori	Mengisi kategori	Dapat memasukan	[√] Diterima

produk, contoh : Kategori : Bamboo craft	produk dengan lengkap	data kategori produk dengan lengkap	[ ] Ditolak
Mengisi Kode Produk, contoh : Kode Produk : 11239	Mengisi kode produk dengan lengkap	Dapat memasukan kode produk dengan lengkap	[√] Diterima [ ] Ditolak
Mengisi nama produk, contoh : Nama Produk : BM123	Mengisi nama produk dengan lengkap	Dapat memasukan nama produk dengan lengkap	[√] Diterima [ ] Ditolak
Mengisi Dimensi Produk, contoh : Dimensi Produk : 12x13x15	Mengisi dimensi produk dengan lengkap	Dapat memasukan dimensi produk dengan lengkap	[√] Diterima [ ] Ditolak
Mengisi Berat Produk, contoh : Berat Produk : 1Kg	Mengisi berat produk dengan lengkap	Dapat memasukan berat produk dengan lengkap	[√] Diterima [ ] Ditolak
Mengisi Deskripsi Produk, contoh : Deskripsi : Barang ini berbentuk bebek yang dihasi rumput kering	Mengisi deskripsi produk dengan lengkap	Dapat memasukan deskripsi produk dengan lengkap	[√] Diterima [ ] Ditolak
Mengisi Stok Produk,	Mengisi stok	Dapat memasukan	[√] Diterima

contoh: Stok : 20	produk dengan lengkap	stok produk dengan lengkap	[ ] Ditolak
Mengisi Harga Produk, contoh : Harga : Rp. 60.000	Mengisi harga produk dengan lengkap	Dapat memasukan harga produk dengan lengkap	[√] Diterima [ ] Ditolak
Memasukan gambar produk, contoh : Gambar : 			[√] Diterima [ ] Ditolak
Menekan tombol Simpan	Menampilkan Pesan “Data Berhasil Dimasukan”	Dapat Menampilkan Pesan “Data Berhasil Dimasukan”	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah tambah produk maka dapat dilihat pada tabel sebagai berikut :

Tabel 4.48 Pengujian Data Salah Tambah Produk

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Tidak mengisi Data produk dengan lengkap	Menampilkan pesan kesalahan "Data Tidak Boleh Kosong"	Dapat menampilkan pesan kesalahan "Data Tidak Boleh Kosong"	[√] Diterima [ ] Ditolak

4.2.1.3 Kesimpulan Pengujian *Black Box*

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

4.2.2 Pengujian *Beta*

Pengujian *beta* merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan mengenai kepuasan *user* dengan kandungan poin yaitu pemenuhan kebutuhan dari tujuan awal

pembangunan *website e-commerce* dan tampilan antarmuka dari *website e-commerce* tersebut. Pengujian beta dilakukan melalui kuisioner

Pengujian beta melalui kuesioner dilakukan dengan membuat kuesioner mengenai kepuasan user dengan kandungan poin syarat *user friendly* untuk selanjutnya dibagikan kepada sebagian *user* dengan mengambil sample sebanyak 10 orang. Dari hasil kuesioner tersebut akan dilakukan perhitungan untuk dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru.

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

1. Lingkungan penjualan di Bayan Handicraft

   Penelitian di lingkungan penjual di Bayan Handicraft dilakukan supaya dapat mengetahui sejauh mana sistem yang dibangun dapat menjadi alternative 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 sitem yang dibangun dapat memberikan kemudahan masyarakat umum untuk melakukan transaksi pemesanan dan pembelian barang secara *online*.

4.2.2.1 Wawancara Pengguna

Wawancara dilakukan dengan menggunakan teknik kualitatif melalui pertanyaan terbuka. Wawancara mengenai halaman *administrator* dilakukan terhadap 1 orang yaitu pemilik Bayan Handicraft, sedangkan untuk wawancara mengenai halaman *operator* dilakukan terhadap 2 orang pegawai Bayan Handicraft. Dari hasil wawancara tersebut akan dilakukan pengambilan kesimpulan terhadap penilaian penerapan sistem yang baru. Wawancara terdiri dari 5 pertanyaan terbuka yang akan ditanyakan kepada narasumber.

4.2.2.1.1 Wawancara Pengujian Beta Untuk Administrator

Wawancara dilakukan untuk pengujian beta terhadap *website e-commerce* Bayan Handicraft kepada pemilik toko dengan total narasumber sebanyak 1 orang.

1. Secara *interface* bagaimana pendapat anda tentang *website e-commerce* Bayan Handicraft?

   Jawab : Ya, Sudah Cukup, tampilannya cukup *friendly* dengan tampilannya dan saya tidak banyak menemukan kesulitan dalam mengoprasikan halaman *admin* ini.

2. Apakah anda mengerti instruksi pada website tersebut ?

   Jawab : ya, saya cukup mengerti intruksi yang ada di tiap menu-menu dalam *website* ini dan tidak tidak banyak kesulitan

3. Apakah anda setuju dengan adanya *website e-commerce* Bayan Handicraft dapat menjadi sarana efektif dalam memasarkan produknya?

Jawab : Bisa saja, karena konsumen yang jauh dari toko pun dapat mengakses *website* ini sehingga dapat memperluas cakupan pemasaran Bayan Handicraft.

4. Menurut pendapat anda, bagaimana sistem pengolahan data *operator* dan pelanggan yang ada di halaman *administrator website e-commerce* ini ?

   Jawab : sudah baik, dengan adanya *website* ini saya dapat dengan mudah mengelola data petugas yang menangani penjualan .

5. Menurut pendapat anda, apakah situs *operator website e-commerce* Bayan Handicraft dapat membantu dalam pengelolaan laporan penjualan dan produk?

   Jawab : sudah cukup membantu, karena semua data-data tersimpan secara otomatis, sehingga meminimalisasikan *human error* dan tersimpan melalui media *online* sehingga mudah ketika dipanggil sewaktu-waktu untuk digunakan sebagai laporan dan laporan yang dikeluarkan pun sesuai dengan proses transaksi yang terjadi

4.2.2.1.2 Wawancara Pengujian Beta Untuk *Operator*

1. Secara *interface* bagaimana pendapat anda tentang *website e-commerce* Bayan Handicraft ?

   Jawaban 1 : cukup menarik, simpel dan tidak membingungkan

   Jawaban 2 : Bagus, menu-menunya tidak membingungkan

2. Menurut anda, apakah fungsionalitas dari situs *operator* pada *website e-commerce* Bayan Handicraft sudah memenuhi kebutuhan dalam pengolahan datanya ?

   Jawaban 1 : sudah cukup baik, semua fungsionalitas yang saya perlukan sudah ada

   Jawaban 2 : sudah, hal yang berhubungan dengan proses penjualan telah tersedia

3. Menurut pendapat anda, apakah situs *operator website e-commerce* Bayan Handicraft dapat membantu dalam mengelola data transaksi dan data produk ?
   Jawaban 1 : ya, terutama pada fungsionalitas pencarian barang
   Jawaban 2 : ya, karena prosesnya mudah dan memberikan informasi yang cukup

4. Menurut pendapat anda, bagaimana kelengkapan informasi yang diberikan oleh *website e-commerce* Bayan Handicraft terutama pada halaman *operator* ?
   Jawaban 1 : sudah bagus, semua informasi telah tersedia untuk operator
   Jawaban 2 : sudah cukup lengkap

5. Menurut pendapat anda, apakah situs *operator website e-commerce* Bayan Handicraft dapat membantu dalam pengelolaan laporan penjualan dan produk?
   Jawaban 1 : cukup membantu, karena data disimpan dalam *database* secara online sehingga tidak perlu lagi khawatir terselip atau hilang
   Jawaban 2 : membantu, data tersebut tersimpan otomatis sehingga laporan yang laporan yang akan tercetak pun sesuai dengan proses transaksi sebenarnya

Berdasarkan hasil wawancara dari setiap jawaban yang diberikan oleh narasumber terhadap pertanyaan pengujian beta untuk halaman *administrator* dan

operator, dapat ditarik kesimpulan utama yaitu *website e-commerce* Bayan Handicraft sudah memenuhi tujuan awal pembangunan *website* tersebut serta *website e-commerce* memiliki fungsionalitas yang sesuai dengan kebutuhan pegawai di Bayan Handicraft.

4.2.2.2 Kuesioner Pengguna

Kuesioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 10 responden. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuesioner ini dianalisis dengan membuat skor A=5, B=4, C=3, D=2, dan E=1 untuk masing-masing jawaban dari setiap pertanyaan.Kuisioner ini terdiri dari 5 pertanyaan (contoh kuisioner dapat dilihat pada lampiran) dengan menggunakan skala *likert* dengan skala 1 sampai 5.

4.2.2.2.1 Kuesioner Pengujian Beta Untuk Pelanggan

1. Menurut anda, dengan adanya *website e-commerce* Bayan Handicraft memudahkan anda untuk melakukan transaksi tanpa harus datang langsung ke toko Bayan Handicraft?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

2. Menurut anda, apakah *website e-commerce* Bayan Handicraft sudah dapat menyajikan informasi barang (stok, keterangan, deskripsi barang) dan informasi lain yang dibutuhkan oleh *member*?

A. Sangat Setuju
B. Setuju
C. Cukup Setuju
D. Kurang Setuju
E. Tidak Setuju

Tabel 4.49 Perhitungan Jumlah Skor Untuk Seluruh Responden Masyarakat Umum

Jawaban	**Skor**	**Responden**	**Jumlah Pertanyaan**	**(Skor x Pertanyaan)**	**Responden x (Skor x Pertanyaan)**
A	5	30	1	5	150
B	4	30	1	4	120
C	3	30	1	3	90
D	2	30	1	2	60
E	1	30	1	1	30

Berdasarkan perhitungan dari jumlah skor untuk seluruh responden dari rumus Responden x (skor x pertanyaan) maka didapat Jumlah skor ideal (maksimum) adalah 150. Berikut ini adalah hasil skor perhitungan masing-masing jawaban yang sudah dihitung nilainya. Kuisioner ini diujikan kepada 30 responden sebagai *sample* yang diambil dari masyarakat umum.

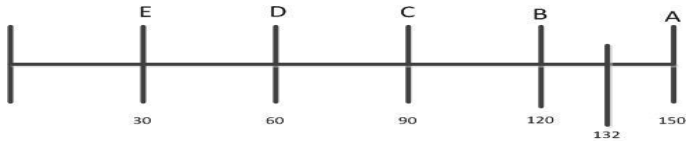
Tabel 4.50 Hasil Pengujian Kuesioner Pengujian Beta Untuk Pelanggan

Responden	Pertanyaan	Total
	No.2	
1	4	4
2	4	4
3	5	5
4	4	4
5	4	4
6	5	5
7	5	5
8	4	4
9	5	5
10	4	4
11	4	4
12	4	4
13	5	5
14	4	4
15	4	4
16	5	5
17	5	5

18	4	4
19	5	5
20	4	4
21	4	4
22	4	4
23	5	5
24	4	4
25	4	4
26	5	5
27	5	5
28	4	4
29	5	5
30	4	4
Total		132

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 132.

Secara kontinum dapat digambarkan sebagai berikut.



Berdasarkan data yang diperoleh, skor yang didapat terletak pada posisi melebihi jawaban B, maka skor penilaian sistem informasi ini berada diatas rata-rata skor dan sebagian besar responden sangat setuju bahwa *website e-commerce* Bayan Handicraft sudah dapat menyajikan informasi barang (stok, keterangan, deskripsi barang) dan informasi lain yang dibutuhkan oleh *member*.

3. Apakah anda setuju dengan adanya *website e-commerce* Bayan Handicraft dapat menjadi sarana promosi yang efektif untuk memperkenalkan produk yang dijualnya?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

Tabel 4.51 Perhitungan Jumlah Skor Untuk Seluruh Responden Masyarakat Umum

Jawaban	Skor	Responden	Jumlah Pertanyaan	(Skor x Pertanyaan)	Responden x (Skor x Pertanyaan)
A	5	30	1	5	150

B	4	30	1	4	120
C	3	30	1	3	90
D	2	30	1	2	60
E	1	30	1	1	30

Berdasarkan perhitungan dari jumlah skor untuk seluruh responden dari rumus Responden x (skor x pertanyaan) maka didapat Jumlah skor ideal (maksimum) adalah 150. Berikut ini adalah hasil skor perhitungan masing-masing jawaban yang sudah dihitung nilainya. Kuisioner ini diujikan kepada 30 responden sebagai *sample* yang diambil dari masyarakat umum.

Tabel 4.52 Hasil Pengujian Kuesioner Pengujian Beta Untuk Pelanggan

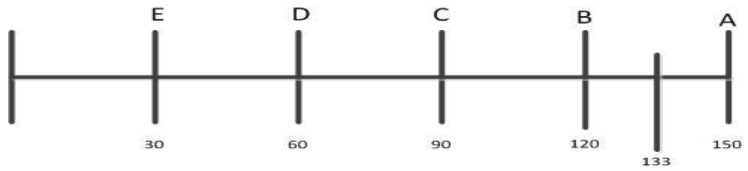
Responden	**Pertanyaan**	**Total**
	No.3	
1	4	4
2	5	5
3	5	5
4	4	4
5	5	5
6	5	5
7	4	4
8	4	4

9	4	4
10	4	4
11	5	5
12	4	4
13	5	5
14	5	5
15	5	5
16	4	4
17	5	5
18	4	4
19	4	4
20	4	4
21	4	4
22	4	4
23	5	5
24	4	4
25	4	4
26	5	5
27	5	5
28	4	4
29	5	5

30	4	4
Total		133

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 133.

Secara kontinum dapat digambarkan sebagai berikut.



Berdasarkan data yang diperoleh, skor yang didapat terletak pada posisi melebihi jawaban B, maka skor penilaian sistem informasi ini berada diatas rata-rata skor dan sebagian besar responden sangat setuju bahwa *website e-commerce* Bayan Handicraft dapat menjadi sarana promosi yang efektif untuk memperkenalkan produk yang dijualnya.

4. Menurut anda, apakah kelengkapan proses pengiriman dan pembayaran yang terdapat pada *website e-commerce* Bayan Handicraft telah sesuai dengan kebutuhan transaksi *online*?

   A. Sangat Setuju
   B. Setuju
   C. Cukup Setuju
   D. Kurang Setuju
   E. Tidak Setuju

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 300. Berikut ini adalah hasil skor perhitungan masing-masing jawaban yang sudah dihitung nilainya. Kuisioner ini diujikan kepada 30 responden sebagai *sample* yang diambil dari masyarakat umum.

Tabel 4.54 Hasil Pengujian Kuesioner Pengujian Beta Untuk Pelanggan

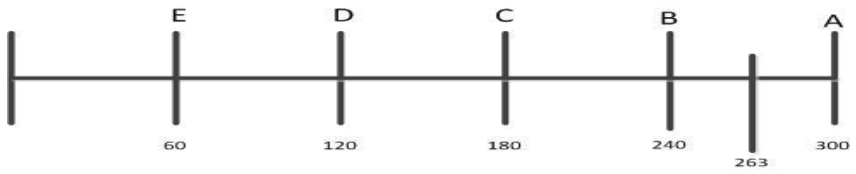
Responden	Pertanyaan		Total
	No. 1	No.4	
1	5	5	10
2	4	5	9

3	4	5	9
4	5	5	10
5	4	5	9
6	4	4	8
7	5	5	10
8	4	3	7
9	4	4	8
10	4	4	8
11	5	5	10
12	5	4	9
13	5	4	9
14	5	5	10
15	5	4	10
16	4	4	8
17	5	5	10
18	3	4	7
19	4	4	8
20	4	4	8
21	4	4	8
22	4	5	9
23	5	5	10

24	4	4	8
25	4	5	9
26	5	3	8
27	5	4	9
28	4	4	8
29	5	4	9
30	4	4	8
Total			263

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 263.

Secara kontinum dapat digambarkan sebagai berikut.



Berdasarkan data yang diperoleh, skor yang didapat terletak pada posisi melebihi jawaban B, maka skor penilaian sistem informasi ini berada diatas rata-rata skor dan sebagian besar responden sangat setuju bahwa

website e-commerce Bayan Handicraft memudahkan untuk melakukan transaksi tanpa harus datang langsung ke toko Bayan Handicraft dan telah sesuai dengan kebutuhan transaksi *online*.

5. Apakah anda setuju bahwa tampilan *website e-commerce* Bayan Handicraft cukup menarik dan mudah dipahami/*user friendly*?

A. Sangat Setuju
B. Setuju
C. Cukup Setuju
D. Kurang Setuju
E. Tidak Setuju

Tabel 4.55 Perhitungan Jumlah Skor Untuk Seluruh Responden Masyarakat Umum

Jawaban	**Skor**	**Responden**	**Jumlah Pertanyaan**	**(Skor x Pertanyaan)**	**Responden x (Skor x Pertanyaan)**
A	5	30	1	5	150
B	4	30	1	4	120
C	3	30	1	3	90
D	2	30	1	2	60
E	1	30	1	1	30

Berdasarkan perhitungan dari jumlah skor untuk seluruh responden dari rumus Responden x (skor x pertanyaan) maka didapat Jumlah skor ideal (maksimum) adalah 150. Berikut ini adalah hasil skor perhitungan masing-masing jawaban yang sudah dihitung nilainya. Kuisioner ini diujikan kepada 30 responden sebagai *sample* yang diambil dari masyarakat umum.

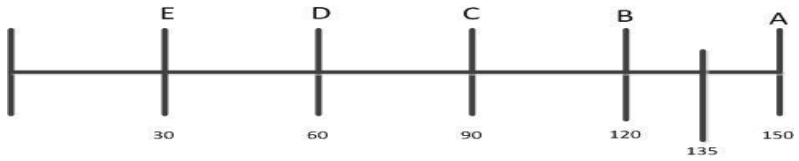
Tabel 4.56 Hasil Pengujian Kuesioner Pengujian Beta Untuk Pelanggan

Responden	**Pertanyaan**	**Total**
	No.5	
1	5	5
2	4	4
3	5	5
4	5	5
5	5	5
6	4	4
7	3	3
8	4	4
9	4	4
10	3	3
11	4	4
12	4	4

13	5	5
14	4	4
15	4	4
16	5	5
17	5	5
18	4	4
19	5	5
20	4	4
21	3	3
22	4	4
23	4	4
24	5	5
25	4	4
26	4	4
27	5	5
28	4	4
29	4	4
30	5	5
Total		135

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 135.

Secara kontinum dapat digambarkan sebagai berikut.



Berdasarkan data yang diperoleh, skor yang didapat terletak pada posisi melebihi jawaban B, maka skor penilaian sistem informasi ini berada diatas rata-rata skor dan sebagian besar responden sangat setuju bahwa tampilan *website e-commerce* Bayan Handicraft cukup menarik dan mudah dipahami/*user friendly*.

4.2.2.3 Kesimpulan Pengujian *Beta*

Berdasarkan hasil persentase diatas yang didapatkan dari pengujian beta pengguna backend dan masyarakat umum maka pembangunan website *e-commerce* ini sudah sesuai dengan tujuan yaitu:

1. Untuk memberikan kemudahan pembelian produk bagi pelanggan yang berada di daerah yang jauh.
2. Untuk mempermudah pelanggan dalam mendapatkan informasi produk-produk yang ditawarkan.
3. Untuk membangun sarana untuk mempromosikan produk yang efektif kepada konsumen.
4. Untuk mempermudah dalam pembuatan laporan penjualan.