BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi sistem merupakan tahap penciptaan perangkat lunak, tahap kelanjutan dari kegiatan perancangan sistem. Tahap ini merupakan penterjemahan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin serta penerapan perangkat lunak pada keadaan yang sebenarnya.

4.1.1 Perangkat keras yang digunakan

Spesifikasi perangkat keras (*hardware*) yang digunakan untuk menjalankan aplikasi *E-Commerce* di PT. Asep Hendro Racing Sport (AHRS) adalah tercantum pada tabel 4.1 sebagai berikut:

Tabel 4.1 Perangkat keras yang digunakan

No	Perangkat Keras	Spesifikasi
1	Processor	Pentium dual core 2,4 GHz
2	RAM	1 Gb on board
3	Harddisk	500 Gb
4	Monitor	14 inci
5	Keyboard	-
6	Mouse	-

4.1.2 Perangkat Lunak yang digunakan

Spesifikasi perangkat lunak (*Software*) yang digunakan untuk mengembangkan sistem penerapan *E-commerce* untuk sistem penjualan ini adalah tercantum pada tabel 4.2 sebagai berikut :

Tabel 4.2 Perangkat lunak yang digunakan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP profesional
2	PHP	Bahasa Pemrograman
3	*MySQL*	Penyimpanan *database*
4	*PhpMyOperator*	*interface database*
5	*WampServer 2.0*	*web server*
6	Macromedia Dreamweaver 8	Kode dan perancangan tampilan
7	*Mozilla Firefox*	*Web browser*

4.2 Implementasi Basis Data

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

1. Tabel Admin

```
CREATE TABLE IF NOT EXISTS `admin` (
  `id_admin` int(15) NOT NULL auto_increment,
  `nama_admin` varchar(30) NOT NULL,
  `email` varchar(30) NOT NULL,
  `password` varchar(30) NOT NULL,
  `ym` varchar(20) NOT NULL,
  `status` enum('0','1') NOT NULL,
  `type` enum('admin','amember') NOT NULL,
  PRIMARY KEY  (`id_admin`)
 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

2. Tabel Detail_Barang

```
CREATE TABLE IF NOT EXISTS `barang` (
  `id_barang` int(15) NOT NULL auto_increment,
  `id_kategori` int(15) NOT NULL,
  `id_jenis` int(15) NOT NULL,
  `keterangan` text NOT NULL,
  `berat` float NOT NULL,
  `harga` int(13) NOT NULL,
  `stok` int(15) NOT NULL,
  `gambar` varchar(50) NOT NULL,
  `view` int(10) NOT NULL,
  `diskon` int (2) NOT NULL,
  `rating` int(10) NOT NULL,
  PRIMARY KEY  (`id_barang`),
  ADD CONSTRAINT `detailbarang_fk_1` FOREIGN KEY (`id_jenis`)
REFERENCES `jenis` (`id_jenis`),
  ADD CONSTRAINT `detailbarang_fk_2` FOREIGN KEY (`id_kategori`)
REFERENCES `kategori` (`id_kategori`),
 ) ENGINE= InnoDB DEFAULT CHARSET=latin1
```

3. Tabel Kategori

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(10) NOT NULL auto_increment,
  `nama_kategori` varchar(50) NOT NULL,
  PRIMARY KEY  (`id_kategori`)
) ENGINE= InnoDB  DEFAULT CHARSET=latin1;
```

4. Tabel Jenis

```
CREATE TABLE IF NOT EXISTS `jenis` (
  `id_jenis` int(10) NOT NULL auto_increment,
  `nama_jenis` varchar(50) NOT NULL,
  PRIMARY KEY  (`id_jenis`)
) ENGINE= InnoDB  DEFAULT CHARSET=latin1;
```

5. Tabel *Member*

```
CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(10) NOT NULL auto_increment,
  `id_kota` int(15) NOT NULL,
  `email` varchar(50) NOT NULL,
  `password` varchar(30) NOT NULL,
  `nama` varchar(50) NOT NULL,
  `alamat` text NOT NULL,
  `telepon` varchar(15) NOT NULL,
  `kodepos` varchar(5) NOT NULL,
  `status` enum('0','1') NOT NULL,
  `kode_verifikasi` varchar(32) NOT NULL,
  PRIMARY KEY  (`id_member`)
  ADD CONSTRAINT `member_fk_1` FOREIGN KEY (`id_kota`) REFERENCES
`kota` (`id_kota`),
) ENGINE= InnoDB  DEFAULT CHARSET=latin1 ;
```

6. Tabel Pembelian

```
CREATE TABLE IF NOT EXISTS `Penjualan` (
  `id_Pembelian` int(15) NOT NULL auto_increment,
  `id_member` int(15) NOT NULL,
  `id_ongkoskirim` int (15) NOT NULL,
  `id_barang` int (15) NOT NULL,
  `tanggal` date NOT NULL,
  `totalharga` int(15) NOT NULL,
  `no_resi` int(15) NOT NULL
  `id_kategoripengiriman` int(11) NOT NULL,
  `nama_pemesan` varchar(50) NOT NULL,
  `kota_pemesan` varchar(30) NOT NULL,
  `alamat_pemesan` text NOT NULL,
  `telepon_pemesan` varchar(15) NOT NULL,
  `kodepos_pemesan` int(5) NOT NULL,
  `status_pengiriman` enum('0','1') NOT NULL,
  `tglbayar` date NOT NULL,
  PRIMARY KEY  (`id_Ongkoskirim`)
 ADD CONSTRAINT `pembelian_fk_1` FOREIGN KEY (`id_barang`)
REFERENCES `detailbarang` (`id_barang`),
 ADD CONSTRAINT `pembelian_fk_2` FOREIGN KEY (`id_admin`)
REFERENCES `admin` (`id_admin`),
) ENGINE= InnoDB  DEFAULT CHARSET=latin1 ;
```

7. Tabel Jasa Pengiriman

```
CREATE TABLE IF NOT EXISTS `jasapengiriman` (
  `id_jasapengiriman` int(15) NOT NULL auto_increment,
  `nama_jasapengiriman` varchar(50) NOT NULL,
  `keterangan` text NOT NULL,
  PRIMARY KEY  (`id_jasapengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

8. Tabel Jenis Pengiriman

```
CREATE TABLE IF NOT EXISTS `jenispengiriman` (
  `id_jenispengiriman` int(15) NOT NULL auto_increment,
  `id_jasapengiriman` int(15) NOT NULL,
  `nama_jenispengiriman` varchar(50) NOT NULL,
  `keterangan` text NOT NULL,
ADD CONSTRAINT `jenispengiriman_fk_1` FOREIGN KEY
(`id_jasapengiriman`) REFERENCES `jasapengiriman`
(`id_jasapengiriman`),
  PRIMARY KEY  (`id_jenispengiriman`)
  FOREIGN KEY1  (`id_jasapengiriman`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

9. Tabel Ongkos Kirim

```
CREATE TABLE IF NOT EXISTS `ongkoskirim` (
  `id_ongkoskirim` int(15) NOT NULL auto_increment,
  `id_kota` int(15) NOT NULL,
  `id_jenispengiriman` int(15) NOT NULL,
  `harga` int(15) NOT NULL,
  UNIQUE KEY  id_ongkoskirim(`id_ongkoskirim`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

10. Tabel Kontak

```
CREATE TABLE IF NOT EXISTS `kontak` (
  `id_kontak` smallint(10) NOT NULL auto_increment,
  `nama` varchar(50) NOT NULL,
  `email` varchar(50) NOT NULL,
  `telepon` varchar(15) NOT NULL,
  `pesan` text NOT NULL,
  `tanggal` datetime NOT NULL,
  `dibaca` enum('0','1') NOT NULL,
  PRIMARY KEY  (`id_kontak`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

11. Tabel Pembayaran

```
CREATE TABLE IF NOT EXISTS `pembayaran` (
  `id_pembayaran` int(10) NOT NULL auto_increment,
  `id_Penjualan` int(11) NOT NULL,
  `kategori_pembayaran`
enum('transfer_bni','transfer_mandiri','paypal') NOT NULL,
  `paypal_id` varchar(50) NOT NULL,
  `nama_bank` varchar(50) NOT NULL,
  `no_rekening` varchar(50) NOT NULL,
  `atas_nama` varchar(50) NOT NULL,
  `jumlah_transfer` int(10) NOT NULL,
  UNIQUE KEY  id_pembayaran(`id_pembayaran`)
) ENGINE= InnoDB  DEFAULT CHARSET=latin1;
```

12. Tabel Kota

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(10) NOT NULL auto_increment,
  `id_provinsi` smallint(5) NOT NULL,
  `nama_kota` varchar(50) NOT NULL,
  PRIMARY KEY  (`id_kota`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

13. Tabel Provinsi

```
   CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(10) NOT NULL auto_increment,
  `nama_provinsi` varchar(50) NOT NULL,
  PRIMARY KEY  (`id_provinsi`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1;
```

14. Tabel detail Pembelian

```
CREATE TABLE IF NOT EXISTS `pembelian` (
  `id_detailpembelian` int(15) NOT NULL auto_increment,
  `id_Pembelian` int(15) NOT NULL,
  `tanggal` date NOT NULL,
  `id_barang` int (15) NOT NULL,
ADD CONSTRAINT `detailpembelian_fk_1` FOREIGN KEY (`id_pembelian`)
REFERENCES `pembelian` (`id_pembelian`),
  PRIMARY KEY  (`id_pembelian`)
) ENGINE= InnoDB  DEFAULT CHARSET=latin1 ;
```

15. Tabel Detail Retur

```
CREATE TABLE IF NOT EXISTS `detailretur` (
  `id_detailretur` int(15) NOT NULL auto_increment,
  `id_detailpembelian` int(15) NOT NULL,
 `tanggal` datetime NOT NULL,
  `id_retur`  (15) NOT NULL,
ADD CONSTRAINT `detailretur_fk_1` FOREIGN KEY (`id_retur`)
REFERENCES `retur` (`id_retur`),
ADD CONSTRAINT `detailretur_fk_2` FOREIGN KEY
(`id_detailpembelian`) REFERENCES `detailpembelian`
(`id_detailpembelian`),
  UNIQUE KEY  id_detailretur (`id_detailretur`)
) ENGINE= InnoDB  DEFAULT CHARSET=latin1 ;
```

16. Tabel Retur

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(15) NOT NULL auto_increment,
    status int(15) NOT NULL,
   qty int(15) NOT NULL,
  `telepon` int(15) NOT NULL,
  `alamat` varchar(30) NOT NULL,
`harga` int(15) NOT NULL,
`id_pembelian` int (15) NOT NULL,
`kodepos` int(5) NOT NULL,
`alasan` varchar(30) NOT NULL,
`berat` varchar(15) NOT NULL,
ADD CONSTRAINT `retur_fk1` FOREIGN KEY (`id_pembelian`) REFERENCES
`pembelian` (`id_pembelian`),
  UNIQUE KEY  id_retur(`id_retur`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 ;
```

4.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna pada. Adapun untuk tampilan implementasi pembangunan aplikasi *E-commerce* di PT. Asep Hendro Racing Sport (AHRS) berbasis *Web* akan dilampirkan pada halaman lampiran.

Tabel 4.3 Implementasi antarmuka Pengunjung dan *Member*

Menu	Deskripsi	Nama File
Daftar	Digunakan untuk melakukan pendaftaran *member* baru.	Daftar.php
Login	Digunakan sebagai halaman *login member*	*login*.php
Lupa *password*	Digunakan jika *member* lupa dengan *password*	lupa-*password*.php
Tentang Kami	Digunakan untuk melihat data profil perusahaan	tentang-kami.php
Cara pembayaran	Digunakan untuk mengetahui informasi pembayan terhadap pembelian barang	Cara-pembayaran.php
Cara pemesanan dan retur	Digunakan untuk melihat cara pemesanan dan aturan dalam retur barang yang telah di beli	Cara-pemesanan.php
Ubah Profil	Digunakan untuk mengubah profil *member*.	ubah-profil.php
Ubah *password*	Digunakan jika *member* ingin	ubah-*password*.php

	merubah *password*	
Kategori	Digunakan untuk melihat kategori barang	kategori.php
Jenis	Digunakan untuk melihat jenis barang	jenis.php
Hubungi Kami	Digunakan untuk menyampaikan pesan terhadap perusahaan lewat website ini.	hubungi-kami.php
Keranjang belanja	Digunakan untuk melihat, merubah, menghapus data pembelian yang dilakukan.	cart.php
Riwayat Pembelian	Digunakan untuk melihat *history* pembelian yang pernah dilakukan.	riwayat-pembelian.php
Pencarian	Digunakan untuk mencari kategori dan jenis barang	cari.php
Info Barang	Digunakan untuk melihat detail barang	Detail.php
Beranda	Digunakan untuk memudahkan member kembali ke tampilan awal website	Index.php
Logout	Digunakan *member* untuk keluar dari akunnya.	*logout*.php

Tabel 4.4 Implementasi Antarmuka Admin

Menu	Deskripsi	Nama File
Login	Digunakan sebagai halaman *login* operator	login.php
Lupa *Password*	Digunakan jika admin lupa terhadap *password*nya	Lupa-pasword.php
Dashboard	Digunakan untuk kembali pada	Index.php

	tampilan awal admin	
Kota	Digunakan untuk melihat, menambah, dan merubah data kota	kota.php
Provinsi	Digunakan untuk melihat, menambah, dan merubah data provinsi	provinsi.php
Barang	Digunakan untuk melihat, menambah, dan merubah data barang	barang.php
Kategori	Digunakan untuk melihat, menambah, dan merubah data kategori	kategori.php
Jenis	Digunakan untuk melihat, menambah, dan merubah data jenis	jenis.php
Jasa Pengiriman	Digunakan untuk melihat, menambah, dan merubah data Jasa Pengiriman	jasapengiriman.php
Jenis Pengiriman	Digunakan untuk melihat, menambah, dan merubah data Jenis Pengiriman	jenispengiriman.php
Ongkos Kirim	Digunakan untuk melihat data ongkos kirim, merubah, menghapus ongkos kirim	ongkoskirim.php
Kontak	Digunakan untuk melihat pesan yang disampainkan member pada website ini.	kontak.php
Member	Digunakan untuk melihat dan menghapus data *member*	member.php
Penjualan	Digunakan untuk melihat data transaksi yang masuk dan meubah statusnya.	Penjualan.php
Laporan	Digunakan untuk melihat data laporan penjualan dan retur.	laporan.php
Cetak Penjualan	Digunakan untuk mencetak laporan penjualan	Cetakpenjualan.php
Cetak Barang	Digunakan untuk mencetak laporan barang yg tersedia di perusahaan	Cetakbarang.php
Cetak Retur	Digunakan untuk mencetak laporan Retur	Cetakretur.php
Retur	Digunakan untuk melihat retur barang	Retur.php
Logout	Digunakan operator Untuk dapat	logout.php

	keluar dari aplikasi	
Login Administrator	Digunakan sebagai halaman *login* administrator	index.php
Lupa *Password*	Digunakan jika operator lupa terhadap *password*nya	forgotpassword.php
Ganti *Password*	Digunakan jika administrator ingin mengganti *passord*	Gantipassword.php

4.4 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan – kesalahan atau kekurangan – kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut.

Pengujian perangkat lunak ini menggunakan pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

4.4.1 Pengujian Alpa

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

4.4.1.1 Skenario Pengujian Alpa

Tabel 4.5 Skenario Pengujian Aplikasi *Member*

Kelas Uji	Butir Uji	Kategori Pengujian
Daftar *member*	Menampilkan *form user* baru	*Black Box*
	Isi *form* pendaftaran	*Black Box*
	Simpan data baru	*Black Box*
Login	Isi data *login*	*Black Box*

	Verifikasi *Email*	*Black Box*
	Verifikasi *password*	*Black Box*
Ubah profil	Menampilkan *form* ubah profil	*Black Box*
	Ubah data *member*	*Black Box*
	Simpan perubahan	*Black Box*
ubah *password*	Menampilkan *form* ubah *password*	*Black Box*
	Verifikasi *password* lama	*Black Box*
	Verifikasi *password* baru	*Black Box*
	Verifikasi konfirmasi *password*	*Black Box*
Menu Produk	Menampilkan data barang	*Black Box*
	Menampilkan data barang	*Black Box*
	Cari barang	*Black Box*
Menu Tentang-Kami	Menampilkan profil perusahaan	*Black Box*
Menu Hubungi Kami	Menampilkan *form*	*Black Box*
	Isi *form* hubungi	*Black Box*
Menu keranjang belanja	Menampilkan data pembelian	*Black Box*
	Meng*update* data pembelian	*Black Box*
	Menghapus data pembelian	*Black Box*
Form konfirmasi	Menampilkan konfirmasi transaksi	*Black Box*
Form pengiriman	Menampilkan *form* pengiriman	*Black Box*
	Pilih kategori pengiriman	*Black Box*
Logout	Menghapus *session user*	*Black Box*

Tabel 4.6 Skenario Pengujian Aplikasi Admin

Kelas Uji	**Butir Uji**	**Kategori Pengujian**
Login	Isi data *login*	*Black Box*
	Verifikasi *username*	*Black Box*
	Verifikasi *password*	*Black Box*
Lupa *password*	Menampilkan *form* Lupa *password*	*Black Box*
Ganti *password*	Tampilkan form ganti *password*	*Black Box*
Penjualan	Tampilkan data transaksi	*Black Box*
	Ubah status transaksi	*Black Box*
Barang	Tampilkan data barang	*Black Box*
	Tambah data barang	*Black Box*
	Ubah data barang	*Black Box*
Kategori Barang	Tampilkan data Kategori barang	*Black Box*
	Tambah data kategori barang	*Black Box*
	Ubah data kategori barang	*Black Box*
Jenis barang	Tampilkan data jenis barang	*Black Box*
	Tambah data jenis barang	*Black Box*
	Ubah data jenis barang	*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*
Kontak	Tampilkan data kontak	*Black Box*
	Baca kontak	*Black Box*
Jasa Pengiriman	Tampilkan data jasa pengiriman	*Black Box*
	Tambah data jasa pengiriman	*Black Box*

	Ubah data jasa pengiriman	*Black Box*
Jenis Pengiriman	Tampilkan data kategori pengiriman	*Black Box*
	Tambah data kategori pengiriman	*Black Box*
	Ubah data kategori pengiriman	*Black Box*
Ongkos Kirim	Tampilkan data Ongkos Kirim	*Black Box*
	Tambah data Ongkos Kirim	*Black Box*
	Ubah data Ongkos Kirim	*Black Box*
Member	Tampilan Data *Member*	*Black Box*
Logout	Menghapus *session member*	*Black Box*

4.4.2 Kasus dan Hasil Pengujian

4.4.2.1 Pengujian Daftar *Member*

Berikut dapat dilihat pengujian dari daftar *member* yang terdapat pada tabel 4.7 dibawah ini:

Tabel 4.7 Pengujian Daftar *member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data pendaftaran, contoh masukan : Nama : Rodeo Email : deo_doank@yahoo.co.id	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses daftar.	*Form* daftar diisi semua maka sesuai yang di harapkan	[√] Diterima [ ] Ditolak

Bersambung,

Tabel 4.7 Pengujian Daftar *member* (Lanjutan)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Password : 123rodeo Konfirmasi *password* : 123rodeo Provinsi : Jawa Barat Kota : Bandung Alamat : Jl. Dipati Telepon: 08996160587 Kodepos : 16431			
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data kosong	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[ ] Diterima [√] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data huruf pada form telepon	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus angka.	Dapat menampilkan pesan kesalahan ” harus angka”	[ ] Diterima [√] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data huruf pada kodepos	Tampilkan pesan kesalahan, maka akan muncul pesan kesalahan bahwa field harus angka.	Dapat menampilkan pesan kesalahan ” harus angka”	[ ] Diterima [√] Ditolak

4.4.2.2 Pengujian *Login Member*

Berikut dapat dilihat pengujian *login member* pada tabel 4.8 dibawah ini:

Tabel 4.8 Pengujian *Login Member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data alamat *email* contoh *email*: rodeodoank@gmail.com	data alamat *email* pada *field* alamat *email* terisi dengan benar	Dapat mengisi *field* alamat *email* dan dapat mengisi data *login* dengan benar	[√] Diterima [ ] Ditolak
Memasukan data *Password* : mapaligi112	Mengisikan data *password* pada *field password*. Jika benar akan masuk ke menu utama	Mengisikan data *password* pada *field password*. Setelah benar *member* masuk ke menu utama	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data *email* atau *password* yang tidak terdaftar	Tidak dapat *login*	Muncul pesan "Maaf account tidak di temukan"	[ ] Diterima [√] Ditolak
mengosongkan data *email* atau *password*	Tidak dapat *login*	Dapat menampilkan pesan kesalahan " Maaf account tidak di temukan"	[ ] Diterima [√] Ditolak

4.4.2.3 Pengujian Ubah Profil *Member*

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

Tabel 4.9 Pengujian Ubah Profil *Member*

Kasus dan Hasil Uji Benar (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data profil yang akan diubah	Tercantum pada text box data yang akan diubah	Dapat mengisi *field* data yang akan diubah dengan benar	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan ada yang tidak diisi	Menampilkan pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan bahwa field tersebut harus terisi.	[ ] Diterima [√] Ditolak
Data masukan semuanya tidak diisi	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[ ] Diterima [√] Ditolak
Data kosong	Muncul pesan kesalahan field harus diisi	Dapat menampilkan pesan kesalahan field harus diisi	[ ] Diterima [√] Ditolak

4.4.2.4 Pengujian Ubah *Password Member*

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

Tabel 4.10 Pengujian Ubah *Password Member*

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data yang dimasukkan,contoh *password* :mapaligi112	*Password* baru	data *password* yang	[√] Diterima

Bersambung,

Tabel 4.10 Pengujian Ubah *Password Member* (Lanjutan)

semuanya benar	berhasil disimpan di *database*	baru berhasil dirubah dengan muncul pesan “ Data berhasil diubah”	[ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Salah satu field belum terisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi” bila ada field yang belum diisi	[ ] Diterima [√] Ditolak
Data *password* lama salah	muncul pesan "*password* salah"	Dapat menampilkan pesan kesalahan ”*Password* salah”	[ ] Diterima [√] Ditolak
Data *password* baru dan konfirmasi *password* baru tidak sama	Muncul pesan "*password* baru dan konfirmasi *password* harus sama"	Dapat menampilkan pesan "*password* baru dan konfirmasi *password* harus sama"	[ ] Diterima [√] Ditolak
Data kosong	Muncul pesan kesalahan field harus diisi	Dapat menampilkan pesan kesalahan field harus diisi	[ ] Diterima [√] Ditolak

4.4.2.5 Pengujian *Form* Pengiriman

Berikut dapat dilihat pengujian dari *form* pengiriman pada tabel 4.11 dibawah ini:

Tabel 4.11 Pengujian *Form* Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data masukan benar, contoh Nama lengkap : rodeo Alamat : Jl.Dipatiukur No 112 Kota : Depok Kodepos : 16431 Telepon : 08996160587	Data pengiriman baru akan disimpan di dalam *database*	Dapat data pengirimah baru akan disimpan di dalam *database*	[√] Diterima [ ] Ditolak
Salah satu *field* kosong	Menampilkan pesan kesalahan field tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] Diterima [ ] Ditolak

4.4.2.6 Pengujian *Form* Konfirmasi Pembayaran

Berikut dapat dilihat pengujian dari *form* konfirmasi pembayaran pada tabel 4.20 dibawah ini:

Tabel 4.20 Pengujian *Form* Konfirmasi Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data konfirmasi pembayaran, contoh No. Penjualan : 2 Nama Bank : BNI Atas Nama : Rodeo Jumlah Transfer : 600000	Menampilkan *form* konfirmasi	Dapat memunculkan pesan *form* konfirmasi.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan

Bersambung,

Tabel 4.20 Pengujian *Form* Konfirmasi Pembayaran (Lanjutan)

Memasukan data jumlah pembayaran kurang dari total pembayaran	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa pembayaran yang dilakukan kurang	Dapat menampilkan pesan kesalahan ”jumlah pembayaran kurang”	[ ] Diterima [√] Ditolak
Memasukan data kosong	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[ ] Diterima [√] Ditolak

4.4.2.7 Pengujian Menu Hubungi Kami

Berikut dapat dilihat pengujian dari menu hubungi kami pada tabel 4.21 dibawah ini:

Tabel 4.21 Pengujian Menu Hubungi Kami

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data hubungi kami	Jika semua field terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses hubungi kami.	*Form* hubungi kami diisi semua maka sesuai yang di harapkan	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data kosong	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[ ] Diterima [√] Ditolak

Bersambung,

Tabel 4.21 Pengujian Menu Hubungi Kami (Lanjutan)

Memasukan data huruf pada form telepon	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus angka.	Dapat menampilkan pesan kesalahan ”harus angka”	[ ] Diterima [√] Ditolak

4.4.2.8 Pengujian *Login* Admin

Berikut dapat dilihat pengujian dari *login* Admin pada tabel 4.22 dibawah ini:

Tabel 4.22 Pengujian *Login* Admin

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data email Admin	Data *email* sesuai dengan yang terseimpan di *database*	Dapat mengisi field *username* dengan benar	[√] Diterima [ ] Ditolak
memasukan *password* admin	Data password sesuai dengan yang terseimpan di *database*	Dapat mengisi field *password* dengan benar	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *email* atau *password* tidak terdaftar	Tidak dapat *login*	Muncul pesan “maaf account tidak ditemukan.”	[ ] Diterima [√] Ditolak
Data *email* atau *password* kosong	Tidak dapat *login*	Muncul pesan kesalahan “account tidak ditemukan.”	[ ] Diterima [√] Ditolak

4.4.2.9 Pengujian Pengolahan Tambah Barang

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

Tabel 4.24 Pengujian Pengolahan Tambah Barang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi *form* tambah barang, Kategori barang : kenalpot Jenis barang : sparepart Keterangan : Ninja 250 Harga : 1500000 Berat : 1800 Stok : 50 Gambar : knlapotninja.jpg	Semua *field* terisi dengan benar	*Field* diisi data barang	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukkan	Yang diharapkan	Pengamatan	Kesimpulan
Data barang masih kosong	Tampilkan pesan kesalahan “data harus diisi”	Dapat menampilkan pesan kesalahan ”data harus diisi”	[ ] Diterima [√] Ditolak
Data barang sudah terdaftar	Tampilkan pesan “Barang dengan kategori, jenis dan ukuran sudah tersedia”	Dapat menampilkan pesan kesalahan ”Barang dengan kategori, jenis dan ukuran sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.10 Pengujian Pengolahan Tambah Kategori

Berikut dapat dilihat pengujian dari pengolahan tambah kategori pada tabel 4.16 dibawah ini :

Tabel 4.16 Pengujian Pengolahan Tambah Kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi *form* tambah kategori,contoh Nama kategori : cakram	Semua *field* terisi dengan benar	*Field* diisi data kategori	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data kategori masih kosong	Tampilkan pesan kesalahan “Nama kategori harus diisi”	Dapat menampilkan pesan kesalahan ”Nama kategori harus diisi”	[ ] Diterima [√] Ditolak
Data kategori sudah terdaftar	Tampilkan pesan “kategori barang dengan nama “ “ sudah tersedia”	Dapat menampilkan pesan “kategori barang dengan nama “ “ sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.11 Pengujian Pengolahan Tambah Jenis

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

Tabel 4.28 Pengujian Pengolahan Tambah Jenis

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi *form* tambah jenis,contoh Nama jenis barang : aksesoris	Semua *field* terisi dengan benar	*Field* diisi data jenis	[√] Diterima [ ] Ditolak

Kasus dan Hasil Uji (Data salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data jenis masih kosong	Tampilkan pesan kesalahan “nama jenis harus diisi”	Dapat menampilkan pesan kesalahan ”nama jenis harus diisi”	[ ] Diterima [√] Ditolak
Data jenis sudah terdaftar	Tampilkan pesan “jenis barang dengan nama “ “ sudah tersedia”	Dapat menampilkan pesan “jenis barang dengan nama “ “ sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.12 Pengujian Pengolahan Tambah Jasa Pengiriman

Berikut dapat dilihat pengujian dari pengolahan tambah jasa pengiriman pada tabel 4.30 dibawah ini:

Tabel 4.30 Pengujian Pengolahan Tambah Jasa Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi data npada *form* tambah jasa pengiriman,contoh nama jasa pengiriman : TIKI	Semua *field* terisi dengan benar	*Field* diisi data jasa pengiriman	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Data jasa pengiriman masih kosong	Tampilkan pesan kesalahan “nama jasa pengiriman harus diisi”	Dapat menampilkan pesan kesalahan ”nama jasa pengiriman harus diisi”	[ ] Diterima [√] Ditolak
Data jasa pengiriman sudah terdaftar	Tampilkan pesan “jasa pengiriman dengan nama “ “ sudah tersedia”	Dapat menampilkan pesan “jasa pengiriman dengan nama “ “ sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.13 Pengujian Pengolahan Tambah Jenis Pengiriman

Berikut dapat dilihat pengujian dari pengolahan tambah kategori pengiriman pada tabel 4.32 dibawah ini:

Tabel 4.32 Pengujian Pengolahan Tambah Jenis Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi *form* tambah Jenis pengiriman, contoh nama jenis pengiriman : YES (Yakin Esok Sampai)	Semua *field* terisi dengan benar	*Field* diisi data kategori pengiriman	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Data kategori pengiriman masih kosong	Tampilkan pesan kesalahan “nama kategori pengiriman harus diisi”	Dapat menampilkan pesan kesalahan ”nama kategori pengiriman harus diisi”	[ ] Diterima [√] Ditolak
Data kategori pengiriman sudah terdaftar	Tampilkan pesan “kategori pengiriman dengan nama “ “ sudah tersedia”	Dapat menampilkan pesan “kategori pengiriman dengan nama “ “ sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.14 Pengujian Pengolahan Tambah Ongkos Kirim

Berikut dapat dilihat pengujian dari pengolahan tambah ongkos kirim pada tabel 4.34 dibawah ini:

Tabel 4.34 Pengujian Pengolahan Tambah Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi *form* tambah ongkos kirim, contoh jenis pengiriman : Reguler Kota : Serang Harga : 15000	Semua *field* terisi dengan benar	*Field* diisi data ongkos kirim	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Data ongkos kirim masih kosong	Tampilkan pesan kesalahan “nama ongkos kirim harus diisi”	Dapat menampilkan pesan kesalahan ”nama ongkos kirim harus diisi”	[ ] Diterima [√] Ditolak
Data ongkos kirim sudah terdaftar	Tampilkan pesan “ongkos kirim dengan nama “ “ sudah tersedia”	Dapat menampilkan pesan “ongkos kirim dengan nama “ “ sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.15 Pengujian Pengolahan Tambah Kota

Berikut dapat dilihat pengujian dari pengolahan tambah kota pada tabel 4.37 dibawah ini:

Tabel 4.37 Pengujian Pengolahan Tambah Kota

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi *form* tambah kota, contoh nama kota : Serang Provinsi : Banten	Semua *field* terisi dengan benar	*Field* diisi data kota	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan

Data kota masih kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”harus diisi”	[ ] Diterima [√] Ditolak
Data kota sudah terdaftar	Tampilkan pesan “kota dengan nama “ “ di provinsi “ “ sudah tersedia”	Dapat menampilkan pesan “kota dengan nama “ “ di provinsi “ “ sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.16 Pengujian Pengolahan Tambah Provinsi

Berikut dapat dilihat pengujian dari pengolahan tambah provinsi pada tabel 4.39 dibawah ini:

Tabel 4.39 Pengujian Pengolahan Tambah Provinsi

Kasus dan Hasil Uji (Data Benar)			
Data masukan	Yang diharapkan	Pengamatan	Kesimpulan
Mengisi *form* tambah provinsi,contoh Nama provinsi : jawabarat	Semua *field* terisi dengan benar	*Field* diisi data provinsi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapakan	Pengamatan	Kesimpulan
Data provinsi masih kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”nama provinsi harus diisi”	[ ] Diterima [√] Ditolak
Data provinsi sudah terdaftar	Muncul pesan kesalahan *field* tersebut nama provinsi tersebut sudah ada	Dapat menampilkan pesan “provinsi dengan nama “ “ sudah tersedia”	[ ] Diterima [√] Ditolak

4.4.2.17 Pengujian *Login* Administrator

Berikut dapat dilihat pengujian dari *login* administrator pada tabel 4.41 dibawah ini:

Tabel 4.41 Pengujian *Login* Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data *email* administrator, contoh : *email :* deo_doank@yahoo.co.id	Form *email* terisi dengan benar	Dapat mengisi field *username* dengan benar	[√] Diterima [ ] Ditolak
Memasukan data *Password* , contoh : *Password :* mapaligi	Tercantum pada text box *password*	Dapat mengisi field *password* dengan benar	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data *email* atau *password* tidak terdaftar	Tidak dapat *login*	Muncul pesan "maaf account tidak ditemukan."	[ ] Diterima [√] Ditolak
Data *email* atau *password* kosong	Tidak dapat *login*	Muncul pesan kesalahan "account tidak ditemukan."	[ ] Diterima [√] Ditolak

4.4.2.18 Pengujian Ganti *Password* Administrator

Berikut dapat dilihat pengujian dari ganti *password administrator*pada tabel 4.42 dibawah ini:

Tabel 4.42 Pengujian Ganti *Password* Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Masukan data *password* lama,	*form password* lama terisi sesuai	Dapat memeriksa data *password* yang	[√] Diterima [ ] Ditolak

Bersambung,

Tabel 4.42 Pengujian Ganti *Password* Administrator (Lanjutan)

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
contoh : Mapaligi	dengan *password* lama	lama	
Masukan data *password* baru, contoh : unikom	*form password* lama terisi sesuai dengan *password* baru	Dapat menyimpan data *password* yang baru	[√] Diterima [ ] Ditolak
Masukan data konfirmasi *password* baru, contoh : unikom	*form password* lama terisi sesuai dengan *password* baru	Memeriksa kesamaan data *password* yang baru	[√] Diterima [ ] Ditolak
data yang dimasukkan semuanya benar	*Password* baru berhasil disimpan di *database*	Dapat menyimpan data *password* yang baru	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Data password lama masih kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*password lama harus diisi*”	[ ] Diterima [√] Ditolak
Data *password* baru masih kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*password* baru harus diisi”	[ ] Diterima [√] Ditolak
Data konfirmasi *password* baru masih kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”konfirmasi *password* baru harus diisi”	[ ] Diterima [√] Ditolak
Data *password* lama salah	muncul pesan "*password* salah"	Dapat menampilkan pesan kesalahan ”*Password* salah”	[ ] Diterima [√] Ditolak
Data *password* baru dan konfirmasi *password* baru tidak sama	Muncul pesan "*password* baru dan konfirmasi *password* harus sama"	Dapat menampilkan pesan "*password* baru dan konfirmasi *password* harus sama"	[ ] Diterima [√] Ditolak

4.4.3 Kesimpulan Hasil Pengujian Alpa

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

4.4.4 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan dengan membuat kuesioner, untuk selanjutnya dibagikan kepada pengguna dengan mengambil sample sebanyak 22 orang (3 orang sebagai admin dan 19 orang sebagai member).

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

1. Lingkungan Penjual di PT. Asep Hendro Racing Sport (AHRS)

   Penelitian di lingkungan penjual di PT. Asep Hendro Racing Sport (AHRS) 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 secara *online*.

4.4.4.1 Wawancara Pengujian Beta Untuk Admin

Wawancara yang dipakai untuk pengujian beta di PT.AHRS menggunakan wawancara terstruktur, karena digunakan sebagai teknik pengumpulan data, bila peneliti atau pengumpul data telah mengetahui dengan pasti tentang informasi apa saja yang akan diperoleh. Oleh karena itu dalam melakukan wawancara, pengumpul data telah menyiapkan instrumen penelitian berupa pertanyaan – pertanyaan alternatif yang jawabanya telah disiapkan. Dengan wawancara ini responden diberi pertanyaan yang sama dan pengumpul data mencatatnya, supaya setiap pewawancara mempunyai keterampilan yang sama [5].

Berikut adalah pertanyaan – pertanyaan yang akan di jadikan instrumen penelitian untuk mewawancara admin yang ada di PT.AHRS

1. Apakah anda setuju bahwa *website* ini dapat lebih meningkatkan pelayanan terhadap pelanggan?
2. Apakah anda setuju instruksi yang digunakan website ini mudah dimengerti dan dipahami?
3. Apakah anda setuju bahwa *website* ini dapat dijadikan sebagai sarana promosi produk-produk dari PT. Asep Hendro Racing Sport (AHRS)?
4. Apakah anda setuju dengan adanya website ini dapat membantu pengolahan data barang?
5. Apakah anda setuju website ini dapat membantu dalam pembuatan laporan penjualan?

   Dengan jawaban tiap masing – masing pertanyaan adalah sebagai berikut :

A. Sangat Setuju

B. Setuju

C. Kurang Setuju

E. Tidak Setuju

Karena jawaban secara keseluruhan untuk pertanyaan diatas adalah setuju maka sistem *e-commerce* di PT. Asep Hendro Racing Sport (AHRS) tersebut dinilai berhasil.

4.4.4.2 Kuesioner Pengujian Beta Untuk *member*

Kuesioner disebarkan menggunakan teknik penskalaan. Penskalaan adalah prosedur untuk memberikan bilangan (atau simbol lain) pada suatu obyek sehingga bilangan tersebut menunjukkan karakteristik obyek tersebut. Karakteritik tersebut lebih tepatnya diwakili oleh sejumlah indikator atau item. Teknik penskalaan yang digunakan yaitu Skala *Likert.* Likert digunakan untuk mengukur sikap, pendapat dan persepsi seseorang atau kelompok orang tentang sistem *e-commerce ini.* Dengan skala *Likert*, maka variabel yang akan diukur dijabarkan menjadi indikator variabel, kemudian indikator tersebut dijadikan sebagai titik tolak untuk menyusun item – item instrumen yang dapat berupa pernyataan atau pertanyaan. Jawaban setiap item instrumen yang menggunakan skala *Likert* mempunyai gradasi dari sangat positif sampai sangat negatif, yang dapat berupa kata – kata.

Dalam pembangunan sistem e-commerce ini digunakan skala *Likert* yang bertujuan untuk analisis kuantitatif, maka jawaban itu diberi skor pada setiap

pertanyaanya. Adapun untuk instrumen penelitian dibuat dalam bentuk kuisioner pilihan ganda. Kuesioner ini terdiri 9 pertanyaan yang ditujukan kepada *member* (hasil kuesioner dapat diliihat pada lampiran) dan 5 pertanyaan ditujukan kepada admin. Setiap pertanyaan diberi pilihan jawaban yang bersifat tertutup (*closed questionare*) dengan jumlah pilihan jawaban sebanyak 5.

1. Apakah Anda setuju bahwa tampilan website PT. Asep Hendro Racing Sport (AHRS) cukup menarik dan mudah dipahami / *user friendly*?
2. Apakah dalam proses pendaftaran member pada website ini sudah memudahkan anda dalam melakukan pendaftaran sebagai member PT. Asep Hendro Racing Sport (AHRS)?
3. Apakah Anda setuju bahwa pemesanan barang pada website ini sudah mudah dilakukan oleh anda sebagai member?
4. Apakah Anda setuju website E-commerce ini dapat membantu anda untuk mendapatkan informasi terbaru tentang produk yang ada di PT. Asep Hendro Racing Sport (AHRS)?
5. Apakah anda setuju bahwa pengguna jasa layanan pengiriman JNE pada website ini dapat memudahkan member dalam melakukan pelacakan keberadaan barang?
6. Apakah anda setuju bahwa sistem pembayaran paypal, transfer bank yang diterapkan pada website ini dapat mempermudahkan member dalam melakukan pembayaran?

7. Apakah anda setuju bahwa semua transaksi yang dikirimkan melalui email oleh website ini sudah lengkap?
8. Apakah Anda setuju bahwa transaksi jarak jauh pada website ini memudahkan pelanggan baik yang berada dalam kota maupun luar kota Depok?
9. Apakah Anda setuju bahwa transaksi pada website ini menjamin keamanan pelanggan karenan tidak perlu membawa uang chas dalam jumlah yang sangat besar?

Dengan jawaban untuk tiap – tiap pertanyaan adalah sebagai berikut :

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

Kuesioner tersebut dianalisis dengan membuat skor 5 untuk “Sangat Setuju”, 4 untuk “Setuju”, 3 untuk “Biasa Saja”, 2 untuk “Tidak Setuju”, dan 1 untuk “Sagat Tidak setuju”, untuk masing-masing jawaban dari setiap pertanyaan. Masing – masing nilai dari setiap pertanyaan akan dijumlhakan (X1+X2+X3+...+Xn = Y), yang nantinya jumlah dari setiap jawaban akan di total (Y1+Y2+Y3+...+Yn = Z), nantinya total tersebut lah yang menjadi *parameter* apakah *website* yang di bangun berhasil atau tidak. Kuesioner tersebut diisi oleh 19 responden yang menjadi member di PT. Asep Hendro Racing Sport (AHRS). Hasil skor terkecil dan terbesar untuk satu orang responden dan total semua responden dapat dilihat pada tabel 4.43

Tabel 4.43 Hasil Pengujian Kuesioner Beta *member*

Responden	Pertanyaan									Total
	No. 1	No. 2	No. 3	No. 4	No. 5	No. 6	No. 7	No. 8	No. 9	
A	2	4	4	5	3	4	4	4	5	35
B	4	5	4	4	5	5	4	4	4	39
C	4	5	4	4	5	5	3	4	5	39
D	4	5	5	4	5	4	4	5	5	41
E	3	4	4	4	5	4	4	5	3	36
F	4	4	4	4	4	4	4	4	4	36
G	5	5	5	5	5	5	5	5	5	45
H	4	3	5	5	3	4	5	5	4	38
I	5	4	4	4	3	4	4	4	3	35
J	4	4	5	4	4	5	3	4	2	35
K	3	3	3	4	4	4	4	5	5	35
L	5	4	4	4	5	4	3	5	4	38
M	5	5	5	5	5	4	4	4	5	42
N	4	5	5	4	4	5	4	4	3	38
O	4	4	4	4	4	4	4	4	4	36
P	5	3	4	4	4	5	4	5	5	40
Q	4	5	4	4	3	4	4	5	4	37
R	5	4	5	3	5	4	3	5	3	37
S	5	4	4	5	4	4	4	4	4	38
										717

Perhitungan jumlah skor untuk setiap responden dapat dilihat pada tabel 4.44 :

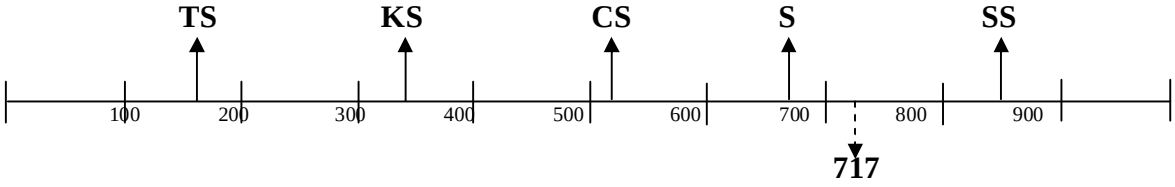
Tabel 4.44 Perhitungan Jumlah Skor Untuk Setiap Responden

Skor	Pertanyaan	(Skor x Pertanyaan)	Keterangan
5	9	45	Sangat Setuju
4	9	36	Setuju
3	9	27	Cukup Setuju
2	9	18	Kurang Setuju
1	9	9	Tidak setuju

Perhitungan jumlah skor untuk seluruh responden dapat dilihat pada tabel 4.45 :

Tabel 4.45 Perhitungan Jumlah Skor Untuk Seluruh Responden

Responden	(Skor x Pertanyaan)	Responden x (Skor x Pertanyaan)	Keterangan
19	45	855	Sangat Setuju (SS)
19	36	684	Setuju (S)
19	27	513	Cukup Setuju (CS)
19	18	342	Kurang Setuju (KS)
19	9	171	Tidak Setuju (TS)



Karena jumlah skor keseluruhan untuk kasus diatas adalah 717 maka sistem *e-commerce* di PT. Asep Hendro Racing Sport (AHRS) tersebut dinilai berhasil.

4.4.5 Kesimpulan Pengujian Beta

Berdasarkan hasil perhitungan dengan menggunakan skala *Likert* diatas, maka penerapan sistem *e-commerce* untuk sistem penjualan ini sudah sesuai dengan tujuan yang diharapkan.