BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

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

4.1.1 Perangkat Keras Yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam pembangunan aplikasi *E-Commerce* ini tercantum pada tabel 4.1 sebagai berikut.

Tabel 4. 1 Perangkat Keras Yang Digunakan

Perangkat	Spesifikasi
Processor	Pentium IV
RAM	1GB
HARDDISK	200GB
Monitor	Color 15”
VGA	128MB

4.1.2 Perangkat Lunak Yang Digunakan

Spesifikasi perangkat lunak yang digunakan dalam membangun aplikasi *E-Commerce* ini tercantum pada tabel 4.2 sebagai berikut.

Tabel 4. 2 Perangkat Lunak Yang Digunakan

Perangkat Lunak	**Keterangan**
Sistem Operasi	Windows XP
Bahasa Pemrograman	PHP
DBMS	MySql 5.0
Web Browser	Mozilla Firefox 4.0
Code Editor	Adobe Dreamweaver Cs3
Tool Antar Muka	Microsoft Visio 2007
Web Server Localhost	WAMP

4.2 Implementasi Basis Data

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

1. **Tabel tadmin**

```
CREATE TABLE IF NOT EXISTS `tadmin` (
  `id_admin` int(11) NOT NULL auto_increment,
  `nama_lengkap` varchar(100) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(200) NOT NULL,
  `email` varchar(50) NOT NULL,
  `no_tlp` varchar(20) default NULL,
  PRIMARY KEY  (`id_admin`),
  UNIQUE KEY `email_UNIQUE` (`email`)
) ENGINE=InnoDB;
```

2. **Tabel tmember**

```
CREATE TABLE IF NOT EXISTS `tmember` (
  `id_member` int(11) NOT NULL auto_increment,
  `nama_lengkap` varchar(100) NOT NULL,
  `no_ktp` varchar(100) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(200) NOT NULL,
  `email` varchar(50) NOT NULL,
  `alamat_member` text,
  `kodepos` varchar(7) default NULL,
  `no_tlp` varchar(20) default NULL,
  `status` enum('0','1') default NULL,
  `tgl_gabung` date NOT NULL,
  `gambar` varchar(100) default NULL,
  PRIMARY KEY  (`id_member`),
  UNIQUE KEY `email_UNIQUE` (`email`),
  UNIQUE KEY `no_ktp_UNIQUE` (`no_ktp`)
) ENGINE=InnoDB;
```

3. **Tabel thotel**

```
CREATE TABLE IF NOT EXISTS `thotel` (
  `id_hotel` int(11) NOT NULL auto_increment,
  `nama_hotel` varchar(45) NOT NULL,
  `alamat_hotel` text,
```

```
  `kelas` int(11) default NULL,
  `lokasi` varchar(100) default NULL,
  `price_deluxe` bigint(20) NOT NULL,
  `price_superior` bigint(20) NOT NULL,
  `price_extrabed` bigint(20) NOT NULL,
  `gambar` varchar(100) default NULL,
  `id_admin` int(11) NOT NULL,
  PRIMARY KEY  (`id_hotel`),
  KEY `fk_thotel_tadmin1` (`id_admin`),
  ADD CONSTRAINT `fk_thotel_tadmin1` FOREIGN KEY (`id_admin`)
  REFERENCES `tadmin` (`id_admin`) ON DELETE NO ACTION ON
  UPDATE NO ACTION
) ENGINE=InnoDB;
```

4. **Tabel tkategori**

```
CREATE TABLE IF NOT EXISTS `tkategori` (
  `id_kategori` int(11) NOT NULL auto_increment,
  `nama_kategori` varchar(45) NOT NULL,
  `id_admin` int(11) NOT NULL,
  PRIMARY KEY  (`id_kategori`),
  KEY `fk_tkategori_tadmin1` (`id_admin`),
 ADD CONSTRAINT `fk_tkategori_tadmin1` FOREIGN KEY
(`id_admin`) REFERENCES `tadmin` (`id_admin`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE = InnoDB;
```

5. **Tabel twisata**

```
CREATE TABLE IF NOT EXISTS `twisata` (
  `id_wisata` int(11) NOT NULL auto_increment,
  `nama_wisata` varchar(100) NOT NULL,
  `gambar` varchar(100) default NULL,
  `deskripsi` text NOT NULL,
  `lokasi` varchar(100) default NULL,
  `id_admin` int(11) NOT NULL,
  PRIMARY KEY  (`id_wisata`),
  KEY `fk_twisata_tadmin1` (`id_admin`),
  ADD CONSTRAINT `fk_twisata_tadmin1` FOREIGN KEY
(`id_admin`) REFERENCES `tadmin` (`id_admin`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE = InnoDB;
```

6. **Tabel tdiskon**

```
CREATE TABLE IF NOT EXISTS `tdiskon` (
  `id_diskon` int(11) NOT NULL auto_increment,
  `tgl_mulai` date NOT NULL,
```

```
  `tgl_akhir` date NOT NULL,
  `diskon` int(11) NOT NULL,
  PRIMARY KEY  (`id_diskon`)
) ENGINE = InnoDB;
```

7. Tabel tpaket

```
CREATE TABLE IF NOT EXISTS `tpaket` (
  `id_paket` int(11) NOT NULL auto_increment,
  `nama_paket` varchar(100) NOT NULL,
  `gambar1` varchar(100) default NULL,
  `gambar2` varchar(100) default NULL,
  `gambar3` varchar(100) default NULL,
  `estimasi_harga` bigint(20) default NULL,
  `desk_paket` text NOT NULL,
  `desk_harga` text,
  `desk_other` text,
  `id_kategori` int(11) NOT NULL,
  `harga_tiket` bigint(20) NOT NULL,
  `tc` bigint(20) NOT NULL,
  `tv` bigint(20) NOT NULL,
  `id_admin` int(11) NOT NULL,
  `id_diskon` int(11) default NULL,
  `id_hotel` int(11) NOT NULL,
  `tgl_buat` date NOT NULL,
  `tgl_update` date NOT NULL,
  `dilihat` int(11) default NULL,
  PRIMARY KEY  (`id_paket`),
  KEY `fk_tpaket_tkategori1` (`id_kategori`),
  KEY `fk_tpaket_tadmin1` (`id_admin`),
  KEY `fk_tpaket_tdiskon1` (`id_diskon`),
  KEY `fk_tpaket_thotel1` (`id_hotel`),
ADD CONSTRAINT `fk_tpaket_tadmin1` FOREIGN KEY (`id_admin`) REFERENCES `tadmin` (`id_admin`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_tpaket_tdiskon1` FOREIGN KEY (`id_diskon`) REFERENCES `tdiskon` (`id_diskon`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_tpaket_thotel1` FOREIGN KEY (`id_hotel`) REFERENCES `thotel` (`id_hotel`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_tpaket_tkategori1` FOREIGN KEY (`id_kategori`) REFERENCES `tkategori` (`id_kategori`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE = InnoDB;
```

8. Tabel tpemesanan

```
CREATE TABLE IF NOT EXISTS `tpemesanan` (
  `id_pemesanan` int(11) NOT NULL auto_increment,
  `tgl_pemesanan` date NOT NULL,
  `id_member` int(11) NOT NULL,
  `id_paket` int(11) NOT NULL,
  `id_hotel` int(11) NOT NULL,
  `jml_org` int(11) NOT NULL,
  `jml_org_child` int(11) NOT NULL,
  `jml_deluxe` int(11) NOT NULL,
  `jml_superior` int(11) NOT NULL,
  `jml_extrabed` int(11) NOT NULL,
  `tgl_pergi` date NOT NULL,
  `total_pesan` bigint(20) NOT NULL,
  `total_bayar` bigint(20) NOT NULL,
  `status` enum('pesan','proses','dp','bayar','cancel') NOT NULL,
  `id_agen` int(11) default NULL,
  PRIMARY KEY  (`id_pemesanan`),
  KEY `fk_tpemesanan_tmember1` (`id_member`),
  KEY `fk_tpemesanan_tpaket1` (`id_paket`),
  KEY `id_agen` (`id_agen`),
ADD CONSTRAINT `fk_tpemesanan_tmember1` FOREIGN KEY (`id_member`) REFERENCES `tmember` (`id_member`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_tpemesanan_tpaket1` FOREIGN KEY (`id_paket`) REFERENCES `tpaket` (`id_paket`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `tpemesanan_ibfk_1` FOREIGN KEY (`id_agen`) REFERENCES `tagen` (`id_agen`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE = InnoDB;
```

9. Tabel tkomentar

```
CREATE TABLE IF NOT EXISTS `tkomentar` (
  `id_komentar` int(11) NOT NULL auto_increment,
  `komentar` text NOT NULL,
  `nama` varchar(100) NOT NULL,
  `email` varchar(100) default NULL,
  `tgl_komentar` date NOT NULL,
  `id_member` int(11) default NULL,
```

```
  PRIMARY KEY  (`id_komentar`),
  KEY `id_member_fk` (`id_member`),
ADD CONSTRAINT `tkomentar_ibfk_1` FOREIGN KEY (`id_member`)
REFERENCES `tmember` (`id_member`)
) ENGINE = InnoDB;
```

10. Tabel tjenispembayaran

```
CREATE TABLE IF NOT EXISTS `tjenispembayaran` (
  `id_jenispembayaran` int(11) NOT NULL auto_increment,
  `nama_jenisbayar` varchar(45) NOT NULL,
  PRIMARY KEY  (`id_jenispembayaran`)
) ENGINE = InnoDB;
```

11. Tabel tpembayaran

```
CREATE TABLE IF NOT EXISTS `tpembayaran` (
  `id_pembayaran` int(11) NOT NULL auto_increment,
  `tgl_pembayaran` date NOT NULL,
  `no_rek` varchar(45) NOT NULL,
  `atasnama` varchar(100) NOT NULL,
  `status` enum('Y','N') default NULL,
  `id_member` int(11) NOT NULL,
  `id_admin` int(11) default NULL,
  `id_jenispembayaran` int(11) NOT NULL,
  `total_transfer` bigint(20) NOT NULL,
  `status_bayar` enum('dp','pelunasan') NOT NULL,
  `id_pemesanan` int(11) NOT NULL,
  PRIMARY KEY  (`id_pembayaran`),
  KEY `fk_tpembayaran_tmember1` (`id_member`),
  KEY `fk_tpembayaran_tadmin1` (`id_admin`),
  KEY `fk_tpembayaran_tjenispembayaran1` (`id_jenispembayaran`),
  KEY `id_pemesanan` (`id_pemesanan`),
ADD CONSTRAINT `fk_tpembayaran_tadmin1` FOREIGN KEY
(`id_admin`) REFERENCES `tadmin` (`id_admin`) ON DELETE NO
ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `fk_tpembayaran_tjenispembayaran1` FOREIGN
KEY (`id_jenispembayaran`) REFERENCES `tjenispembayaran`
(`id_jenispembayaran`) ON DELETE NO ACTION ON UPDATE NO
ACTION,
  ADD CONSTRAINT `fk_tpembayaran_tmember1` FOREIGN KEY
(`id_member`) REFERENCES `tmember` (`id_member`) ON DELETE
NO ACTION ON UPDATE NO ACTION,
  ADD CONSTRAINT `tpembayaran_ibfk_1` FOREIGN KEY
(`id_pemesanan`) REFERENCES `tpemesanan` (`id_pemesanan`)
) ENGINE = InnoDB;
```

12. Tabel tpaketwisata

```
CREATE TABLE IF NOT EXISTS `tpaketwisata` (
  `id_paketwisata` int(11) NOT NULL auto_increment,
  `id_paket` int(11) NOT NULL,
  `id_wisata` int(11) NOT NULL,
  PRIMARY KEY  (`id_paketwisata`),
  KEY `fk_tpaket_has_twisata_twisata1` (`id_wisata`),
  KEY `fk_tpaket_has_twisata_tpaket1` (`id_paket`),
ADD CONSTRAINT `tpaketwisata_ibfk_1` FOREIGN KEY (`id_paket`) REFERENCES `tpaket` (`id_paket`) ON DELETE CASCADE ON UPDATE NO ACTION,
  ADD CONSTRAINT `tpaketwisata_ibfk_2` FOREIGN KEY (`id_wisata`) REFERENCES `twisata` (`id_wisata`) ON DELETE CASCADE ON UPDATE NO ACTION
) ENGINE = InnoDB;
```

13. Tabel tpersonal

```
CREATE TABLE IF NOT EXISTS `tpersonal` (
  `id_personal` int(11) NOT NULL auto_increment,
  `id_pemesanan` int(11) NOT NULL,
  `nama_lengkap` varchar(100) NOT NULL,
  `no_ktp` varchar(100) NOT NULL,
  `no_telp` varchar(50) NOT NULL,
  `gambar` varchar(100) NOT NULL,
  PRIMARY KEY  (`id_personal`),
  KEY `id_pemesanan` (`id_pemesanan`),
 ADD CONSTRAINT `tpersonal_ibfk_1` FOREIGN KEY (`id_pemesanan`) REFERENCES `tpemesanan` (`id_pemesanan`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB;
```

14. Tabel tagen

```
CREATE TABLE IF NOT EXISTS `tagen` (
  `id_agen` int(11) NOT NULL auto_increment,
  `nama` varchar(100) NOT NULL,
  `alamat` text NOT NULL,
  `email` varchar(50) NOT NULL,
  `no_telp` varchar(20) NOT NULL,
  `gambar` varchar(100) NOT NULL,
```

```
  `id_admin` int(11) default NULL,
  PRIMARY KEY  (`id_agen`),
  KEY `fk_id_admin` (`id_admin`),
 ADD CONSTRAINT `tagen_ibfk_1` FOREIGN KEY (`id_admin`)
 REFERENCES `tadmin` (`id_admin`)
)ENGINE=InnoDB;
```

15. Tabel tmail

```
CREATE TABLE IF NOT EXISTS `tmail` (
  `id_email` int(11) NOT NULL auto_increment,
  `id_pemesanan` int(11) NOT NULL,
  `tgl` date NOT NULL,
  `title` varchar(250) NOT NULL,
  `pesan` text NOT NULL,
  `id_admin` int(11) NOT NULL,
  PRIMARY KEY  (`id_email`),
  KEY `id_admin` (`id_admin`),
  KEY `id_pemesanan` (`id_pemesanan`),
  ADD CONSTRAINT `tmail_ibfk_2` FOREIGN KEY (`id_admin`)
REFERENCES `tadmin` (`id_admin`) ON DELETE NO ACTION ON
UPDATE NO ACTION,
  ADD CONSTRAINT `tmail_ibfk_1` FOREIGN KEY (`id_pemesanan`)
REFERENCES `tpemesanan` (`id_pemesanan`) ON DELETE NO
ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB;
```

16. Tabel vote

```
CREATE TABLE IF NOT EXISTS `vote` (
  `id_vote` int(11) NOT NULL auto_increment,
  `counter` int(8) NOT NULL,
  `id_paket` int(11) NOT NULL,
  `value` int(8) NOT NULL,
  PRIMARY KEY  (`id_vote`),
  KEY `id_paket` (`id_paket`),
 ADD CONSTRAINT `vote_ibfk_1` FOREIGN KEY (`id_paket`)
REFERENCES `tpaket` (`id_paket`) ON DELETE NO ACTION ON
UPDATE NO ACTION
) ENGINE=InnoDB;
```

4.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna. Adapun untuk tampilan implementasi pembangunan aplikasi *E-commerce* ini akan dilampirkan pada halaman lampiran.

Tabel 4. 3 Implementasi Antarmuka Pengunjung

Menu	**Deskripsi**	**Nama File**
Beranda	Digunakan untuk melihat tampilan utama	index.php
Data Wisata	Digunakan untuk melihat data wisata yang ada	wisata.php
Data Hotel	Digunakan untuk melihat data hotel yang ada	hotel.php
Paket Tour	Digunakan untuk melihat paket *tour* yang ada	paket.php
Cara Pemesanan	Digunakan untuk melihat cara melakukan pemesanan	pesan.php
Cara Pembayaran	Digunakan untuk melihat cara melakukan pembayaran	pembayaran.php
Tentang Kami	Digunakan untuk melihat tentang Java Sentra madniri Utama	about.php

Login	Digunakan untuk menuju halaman member	login.php
Daftar	Digunakan untuk pendaftaran member baru	daftar.php

Tabel 4. 4 Implementasi Aantarmuka Member

Menu	**Deskripsi**	**Nama File**
Beranda	Digunakan untuk melihat tampilan utama	index.php
Data Wisata	Digunakan untuk melihat data wisata yang ada	wisata.php
Data Hotel	Digunakan untuk melihat data hotel yang ada	hotel.php
Paket Tour	Digunakan untuk melihat paket *tour* yang ada	paket.php
Cara Pemesanan	Digunakan untuk melihat cara melakukan pemesanan	pesan.php
Cara Pembayaran	Digunakan untuk melihat cara melakukan pembayaran	pembayaran.php
Tentang Kami	Digunakan untuk melihat tentang Java Sentra madniri Utama	about.php

Profil	Digunakan untuk melihat profil member bersangkutan	profile.php
Ubah Profil	Digunakan apabila member ingin mengubah profilnya	profile.php
Ubah Password	Digunakan apabila member ingin mengubah password	profile.php
Detail Paket	Digunakan untuk melihat spesifikasi detail paket *tour*	detail.php
Booking	Digunakan apabila ingin memesan salah satu paket *tour*	pemesanan.php
Update	Digunakan apabila ada perubahan dalam melakukan pemesanan	pemesanan.php
Proses	Digunakan apabila telah selesai melakukan pemesanan	pemesanan.php
Konfirmasi	Digunakan untuk melakukan konfirmasi pembayaran	konfirmasi.php
Lupa Password	Digunakan apabila member lupa password	lupa_password.php
Logout	Digunakan untuk keluar dari tampilan member	logout.php

Tabel 4. 5 Implementasi Antarmuka Admin

Menu	Deskripsi	Nama File
Login Admin	Digunakan sebagai halaman login admin	Admin/index.php
Tampilan Utama Admin	Digunakan untuk melihat halaman admin	Admin/admin.php
Ubah Profil Admin	Digunakan untuk mengubah profil admin	Admin/ubahprofil.php
Ganti Password	Digunakan untuk mengganti password admin	Admin/ganti_password.php
Member	Digunakan untuk melihat dan menghapus data member	Admin/datamember.php
Admin	Digunakan untuk melihat, menambah, mengedit, dan menghapus data admin	Admin/dataadmin.php
Wisata	Digunakan untuk melihat, menambah, mengedit, dan menghapus data wisata	Amin/datawisata.php
Kategori	Digunakan untuk melihat, meambah,mengedit, dan menghapus data kategori	Admin/datakategori.php

Hotel	Digunakan untuk melihat, meambah, mengedit, dan menghapus data hotel	Admin/datahotel.php
Paket	Digunakan untuk melihat , meambah, mengedit, dan menghapus data paket	Admin/datapaket.php
Pemesanan	Digunakan untuk melihat, mengupdate pemesanan yang pernah terjadi	Admin/datapemesanan.php
Harga	Digunakan untuk melihat , meambah, mengedit, dan menghapus data harga	Admin/dataharga.php
Diskon	Digunakan untuk melihat , meambah, mengedit, dan menghapus data paket	Admin/datadiskon.php
Database	Digunakan untuk mem-*backup* database	Admin/backupdb.php
Laporan	Digunakan untuk melihat daftar data pemesanan paket lebih rinci	Admin/laporan.php
Logout	Digunakan admin untuk keluar dari aplikasi	Admin/logout.php

4.4 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untuk menemukan kesalahan–kesalahan atau kekurangan–kekurangan pada perangkat lunak yang diuji. Pengujian bermaksud untuk mengetahui perangkat lunak yang dibuat sudah memenuhi kriteria yang sesuai dengan tujuan perancangan perangkat lunak tersebut. Pengujian perangkat lunak ini menggunakan pengujian *black box*. Pengujian *black box* berfokus pada persyaratan fungsional perangkat lunak.

4.5 Pengujian Alpa

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

4.5.1 Skenario Pengujian Alpa

4.5.1.1 Skenario pengujian Aplikasi Member

Tabel 4. 6 Skenario Pengujian Aplikasi Member

Kelas Uji	Detail Pengujian	Jenis Pengujian
Registrasi Member	Menampilkan halaman pengisian *member* baru	*Black Box*
	Isi *Form* pendaftaran	
	Simpan data *member* baru	
Login	Isi data *login*	*Black Box*

	Verifikasi *Email*	
	Verifikasi *Password*	
Profil	Menampilkan *form* profil	*Black Box*
	Ubah profil data *member*	
	Simpan Perubahan data *member*	
	Ganti *password member*	
	Verifiksi *password* lama	
	Verifikasi *password* baru	
	Verifikasi konfirmasi *password*	
Cara Pemesanan	Menampilkan *Form* Cara Pemesanan	*Black Box*
Cara Pembayaran	Menampilkan *Form* Cara Pembayaran	*Black Box*
Data Wisata	Menampilkan *Form* Data Wisata	*Black Box*
Data Hotel	Menampilkan *Form* Data Hotel	*Black Box*
Paket Tour	Menampilkan *Form* Paket Tour	*Black Box*
Tentang Kami	Menampilkan *Form* Tentang Kami	*Black Box*
Keranjang	Menampilkan data pemesanan	*Balck Box*

Pemesanan	Mengupdate data pemesanan	
	Menghapus data pemesanan	
Konfirmasi	Menampilkan *Form* Konfirmasi	*Black Box*
	Menampilkan *Form* isi data pemesanan	
	Pilih jenis Pembayaran	
Logout	Menghapus *session member*	*Black Box*

4.5.1.2 Skenario Pengujian Aplikasi Admin

Tabel 4. 7 Skenario Pengujian Aplikasi Admin

Kelas Uji	**Detail Pengujian**	**Jenis Pengujian**
Login	Isi data *login*	*Black Box*
	Verifikasi *username*	
	Verivikasi *password*	
Ganti *Password*	Masukan *password* lama	*Black Box*
	Masukan *password* baru	
	Menampilkan konfirmasi	

	password	
Member	Tampilan data *member*	*Black Box*
Diskon	Tampilan data diskon	*Black Box*
	Tambah data diskon	
	Edit data diskon	
	Hapus data diskon	
Kategori	Tampilan data kategori	*Black Box*
	Tambah data kategori	
	Edit data kategori	
	Hapus data kategori	
Wisata	Tampilan data wisata	*Black Box*
	Tambah data wisata	
	Edit data wisata	
	Hapus data wisata	
Hotel	Tampilan data hotel	*Black Box*
	Tambah data hotel	

	Edit data hotel	
	Hapus data hotel	
Paket	Tampilan data paket	*Black Box*
	Tambah data paket	
	Edit data paket	
	Hapus data paket	
Pembayaran	Tampilan data pembayaran	*Black Box*
	Ubah status pembayaran	
	Edit data kota	
	Hapus data kota	
Harga	Tampilan data harga	*Black Box*
	Tambah data harga	
	Edit data harga	
	Hapus data harga	
Data Laporan	Tampilan data laporan	*Black Box*
	Download data laporan	

	Cetak data laporan	
Logout	Menghapus session Admin	*Black Box*

4.5.2 Kasus dan Hasil Pengujian

4.5.2.1 Pengujian Registrasi Member

Berikut dapat dilihat pengujian dari registrasi *member* yang terdapat pada tabel 4.8 dibawah ini:

Tabel 4. 8 Pengujian Registrasi Member

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Menu Registrasi	Tampil Form registrasi	Dapat Menampilkan Form registrasi	[√] Diterima [ ] Ditolak
Masukan data pendaftaran, contoh masukan : Nama Lengkap : Roby Miftahurrizki	Jika semua field terisi dengan benar, maka proses registrasi berhasil dan sudah terdaftar dalam	Form registrasi terisi atau benar dalam pengisian, maka sudah terdaftar	[√] diterima [ ] ditolak

Email : roby.miftahurrizki@gmail.com Password : roby Ulangi password : roby No KTP : 1234567834567 Alamat : antapani Kode Pos : 40291 No Tlp : 0227213158	database	sebagai member.	
Klik Tombol Simpan	Jika data masukan benar muncul form "Anda Telah Terdaftar di Situs Sentra Tour sebagai **nama anda** Aktifasi Member Telah Kami Kirim ke **email anda**"	Data masukan benar muncul form "Anda Telah Terdaftar di Situs Sentra Tour sebagai **nama anda** Aktifasi Member Telah Kami	[√] diterima [ ] ditolak

		Kirim ke **email anda**”	
Kasus dan Hasil uji (Data Salah)			
Memasukan data kosong	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	[√] diterima [ ] ditolak
Nama Lengkap tidak diisi	Tampilkan pesan kesalahan”Nama tidak boleh kosong”	Dapat menampilkan Pesan kesalahan “Nama tidak boleh kosong”	[√] diterima [ ] ditolak

Email tidak diisi	Tampilkan pesan kesalahan "Email tidak valid"	Dapat menampilkan pesan kesalahan "Email tidak valid"	[√] diterima [ ] ditolak
password tidak diisi	Tampilkan pesan kesalahan "Password tidak boleh kosong"	Dapat menampilkan pesan kesalahan "Password tidak boleh kosong i"	[√] diterima [ ] ditolak
No KTP	Tampilkan pesan kesalahan "No KTP tidak boleh kosong"	Dapat menampilkan pesan kesalahan "No KTP tidak boleh kosong"	[√] diterima [ ] ditolak
Alamat tidak diisi	Tampilkan pesan kesalahan "alamat	Dapat menampilkan pesan	[√] diterima [ ] ditolak

	tidak boleh kosong"	kesalahan "alamat tidak boleh kosong"	
Kode Pos tidak diisi	Tampilkan pesan kesalahan "kode pos tidak boleh kosong"	Dapat menampilkan pesan kesalahan "kode pos tidak boleh kosong"	[√] diterima [ ] ditolak

4.5.2.2 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 (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik Menu Profile	Menampilkan profile member, update data profile, dan ganti password member	Dapat menampilkan profile member, update data profile, dan ganti password member	[√] diterima [ ] ditolak

Data menu profil yang akan diubah	Dapat mengisi *field* data yang akan diubah	Dapat mengisi *field* data yang akan diubah	[√] diterima [ ] ditolak
Klik tombol simpan	Data berhasil di update di*database*	Data berhasil di update di*database*	[√] diterima [ ] ditolak
Kasus dan hasil Uji (Data Salah)			
Memasukan data kosong	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Tampilkan pesan kesalahan dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	[√] diterima [ ] ditolak
Nama Lengkap tidak diisi	Tampilkan pesan kesalahan"Nama tidak boleh kosong"	Dapat menampilkan Pesan kesalahan "Nama tidak boleh kosong"	[√] diterima [ ] ditolak
Email tidak diisi	Tampilkan pesan kesalahan "Email tidak valid"	Dapat menampilkan pesan kesalahan "Email tidak valid"	[√] diterima [ ] ditolak

password tidak diisi	Tampilkan pesan kesalahan "Password tidak boleh kosong"	Dapat menampilkan pesan kesalahan "Password tidak boleh kosong i"	[√] diterima [ ] ditolak
No KTP	Tampilkan pesan kesalahan "No KTP tidak boleh kosong"	Dapat menampilkan pesan kesalahan "No KTP tidak boleh kosong"	[√] diterima [ ] ditolak
Alamat tidak diisi	Tampilkan pesan kesalahan "alamat tidak boleh kosong"	Dapat menampilkan pesan kesalahan "alamat tidak boleh kosong"	[√] diterima [ ] ditolak
Kode Pos tidak diisi	Tampilkan pesan kesalahan "kode pos tidak boleh kosong"	Dapat menampilkan pesan kesalahan "kode pos tidak boleh kosong"	[√] diterima [ ] ditolak

4.5.2.3 Pengujian Ganti Password Member

Berikut dapat dilihat pengujian dari menu ganti password member pada tabel 4.10 dibawah ini:

Tabel 4. 10 Pengujian Ganti Password Member

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik Menu Profil	Menampilkan profil member, update data profil, dan ganti password member	Dapat menampilkan profil member, update data profil, dan ganti password member	[√] diterima [ ] ditolak
Ganti Password member	Dapat mengisi *field* data yang akan diubah	Dapat mengisi *field* data yang akan diubah	[√] diterima [ ] ditolak
Klik tombol simpan	Password berhasil di update di*database*	Password berhasil di update di*database*	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Salah satu field belum terisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* tersebut harus diisi”	[√] diterima [ ] ditolak
Password lama tidak diisi	muncul pesan kesalahan "password lama tidak boleh kosong”	Dapat menampilkan pesan kesalahan "password lama tidak boleh kosong”	[√] diterima [ ] ditolak

Password baru tidak diisi	muncul pesan kesalahan "password baru tidak boleh kosong”	Dapat menampilkan pesan kesalahan "password baru tidak boleh kosong”	[√] diterima [ ] ditolak
Ulangi Password baru tidak diisi	muncul pesan kesalahan "ulang password baru tidak boleh kosong”	Dapat menampilkan pesan kesalahan "ulangi password baru tidak boleh kosong”	[√] diterima [ ] ditolak

4.5.2.4 Pengujian Cara Pemesanan

Berikut dapat dilihat pengujian dari menu cara pemesanan pada tabel 4.11 dibawah ini:

Tabel 4. 11 Pengujian Cara Pemesanan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik Menu Cara Pemesanan	Menampilkan cara pemesanan	Dapat menampilkan cara pemesanan	[√] diterima [ ] ditolak

4.5.2.4 Pengujian Cara Pembayaran

Berikut dapat dilihat pengujian dari menu cara pembayaran pada tabel 4.12 dibawah ini:

Tabel 4. 12 Pengujian Cara Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik Menu Cara Pembayaran	Menampilkan cara pembayaran	Dapat menampilkan cara pembayaran	[√] diterima [ ] ditolak

4.5.2.5 Pengujian Tentang Kami

Berikut dapat dilihat pengujian dari menu tentang kami pada tabel 4.13 dibawah ini:

Tabel 4. 13 Pengujian Tentang Kami

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik Menu tentang Kami	Menampilkan tentang profile perusahaan secara singkat	Dapat menampilkan tentang profile perusahaan secara singkat	[√] diterima [ ] ditolak

4.5.2.6 Pengujian Menu Keranjang Pemesanan

Berikut dapat dilihat pengujian dari menu keranjang pemesanan pada tabel 4.14 dibawah ini:

Tabel 4. 14 Pengujian Menu Keranjang Pemesanan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik Menu Keranjang Pemesanan	Menampilkan form keranjang pemesanan	Dapat menampilkan form keranjang pemesanan	[√] diterima [ ] ditolak
Jumlah Orang	Tercantum pada text box data yang akan diubah	Dapat mengisi data jumlah dengan benar	[√] diterima [ ] ditolak
Klik Tombol Update	Dapat menampilkan perubahan data jumlah	Dapat menampilkan perubahan data jumlah	[√] diterima [ ] ditolak
Klik tombol hapus	Data yang dihapus terhapus dari keranjang pemesanan	Dapat menghapus data pada keranjang pemesanan	[√] diterima [ ] ditolak
Klik proses	Data pemesanan berhasil disimpan di *database*	Data pemesanan berhasil disimpan di *database*	[√] diterima [ ] ditolak

4.5.2.7 Pengujian Form Konfirmasi Pembayaran

Berikut dapat dilihat pengujian dari *Form* konfirmasi pembayaran pada tabel 4.15 dibawah ini:

Tabel 4. 15 Pengujian Form Konfirmasi Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Masukan data konfirmasi pembayaran, contoh No Rekening : 101062877777 Atas Nama : Roby Miftahurizki Uang yang ditransfer : 100000	Menampilkan *form* konfirmasi	Dapat memunculkan pesan *form* konfirmasi.	[√] diterima [ ] ditolak
Memasukan data kosong	Tampilkan pesan kesalahan ”*field* ini harus diisi”	Dapat menampilkan pesan kesalahan ”*field* ini harus diisi”	[√] diterima [ ] ditolak
Uang yang di transfer kurang dari total pembayaran	Menampilkan pesan “uang yang di transfer kurang”	Dapat menampilkan pesan “uang yang di transfer kurang”	[√] diterima [ ] ditolak

1.5.2.8 Pengujian Login Admin

Berikut dapat dilihat pengujian dari login admin pada tabel 4.16 dibawah ini:

Tabel 4. 16 Pengujian Login Admin

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Username : admin	Tercantum pada text box *username*	Dapat mengisi field *username* dengan benar	[√] diterima [ ] ditolak
Password : admin	Tercantum pada text box *password*	Dapat mengisi field *password* dengan benar	[√] diterima [ ] ditolak
Klik tombol login	jika sama dengan data login maka admin masuk ke menu utama admin	Tombol login dapat berfungsi	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Usename tidak	Menampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan	[√] diterima

diisi	"username/password salah"	"username/password salah"	[ ] ditolak
Password tidak diisi	Menampilkan pesan kesalahan "username/password salah"	Dapat menampilkan pesan kesalahan "username/password salah"	[√] diterima [ ] ditolak
Username salah	Menampulkan pesan "username/password salah"	Dapat menampulkan pesan "username/password salah"	[√] diterima [ ] ditolak
Password salah	Menampulkan pesan "username/password salah"	Dapat menampulkan pesa "username/password salah"	[√] diterima [ ] ditolak

1.5.2.9 Pengujian Ganti Password Admin

Berikut dapat dilihat pengujian dari ganti *password* admin pada tabel 4.17 dibawah ini:

Tabel 4. 17 Pengujian Ganti Password Admin

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**

Ganti Password admin	Dapat mengisi *field* data yang akan diubah	Dapat mengisi *field* data yang akan diubah	[√] diterima [ ] ditolak
Klik tombol simpan	Password berhasil di update di*database*	Password berhasil di update di*database*	[√] diterima [ ] ditolak

Kasus dan Hasil Uji (Data Salah)

Data Masukan	Yang Diharapkan	Pengujian	Kesimpulan
Salah satu field belum terisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* tersebut harus diisi”	[√] diterima [ ] ditolak
Password lama tidak diisi	muncul pesan kesalahan "password lama harus diisi”	Dapat menampilkan pesan kesalahan "password lama harus diisi”	[√] diterima [ ] ditolak
Password baru tidak diisi	muncul pesan kesalahan "password baru harus diisi”	Dapat menampilkan pesan kesalahan "password baru harus diisi”	[√] diterima [ ] ditolak

Ulangi Password baru tidak diisi	muncul pesan kesalahan "ulang password baru harus diisi"	Dapat menampilkan pesan kesalahan "ulangi password baru harus diisi"	[√] diterima [ ] ditolak

1.5.2.10 Pengujian Pengolahan Data Kategori

Berikut dapat dilihat pengujian dari pengolahan data kategori pada tabel 4.18 dibawah ini:

Tabel 4. 18 Pengujian Pengolahan Data Kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik menu kategori	Menampilkan *form* lihat kategori	Dapat menampilkan *form* lihat kategori	[√] diterima [ ] ditolak
Klik tombol tambah	Menampilkan *form* tambah kategori	Dapat menampilkan *form* tambah kategori	[√] diterima [ ] ditolak
Klik tombol simpan	Data kategori bertambah	Data kategori bertambah	[√] diterima [ ] ditolak

Klik tombol edit	Menampilkan *form* edit kategori	Dapat menampilkan *form* edit kategori	[√] diterima [ ] ditolak
Klik tombol hapus	Menampilkan *form* hapus kategori	Dapat menampilkan *form* hapus kategori	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Field tidak diisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* tersebut harus diisi”	[√] diterima [ ] ditolak

1.5.2.11 Pengujian Pengolahan Data Paket

Berikut dapat dilihat pengujian dari pengolahan data Paket pada tabel 4.19 dibawah ini :

Tabel 4. 19 Pengujian Pengolahan Data Paket

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik menu paket	Menampilkan *form* lihat paket	Dapat menampilkan *form* lihat paket	[√] diterima [ ] ditolak

Klik tombol tambah	Menampilkan *form* tambah paket	Dapat menampilkan *form* tambah paket	[√] diterima [ ] ditolak
Klik tombol simpan	Data paket bertambah	Data paket bertambah	[√] diterima [ ] ditolak
Klik tombol edit	Menampilkan *form* edit paket	Dapat menampilkan *form* edit paket	[√] diterima [ ] ditolak
Klik tombol hapus	Menampilkan *form* hapus paket	Dapat menampilkan *form* hapus paket	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Field tidak diisi	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat menampilkan pesan kesalahan ”*field* tersebut harus diisi”	[√] diterima [ ] ditolak
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] diterima [ ] ditolak

1.5.2.12 Pengujian Tambah Paket

Berikut dapat dilihat pengujian dari pengolahan tambah data paket pada tabel 4.20 dibawah ini:

Tabel 4. 20 Pengujian Tambah Paket

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik menu Tambah	Menampilkan *form* tambah paket	Dapat menampilkan *form* tambah paket	[√] diterima [ ] ditolak
Mengisi data paket, contoh : Nama Kategori : Mancanegara Nama Paket : Kualalumpur Tour Nama Hotel : Picollo Hotel Temmpat Wisata : Petronas Tower Gambar :	Jika semua field terisi semua maka data paket berhasil tersimpan	Jika semua field terisi semua maka data paket berhasil tersimpan	[√] diterima [ ] ditolak

 Deskripsi : Paket ini hanya berwisata di daerah Kualalumpur saja			
Klik tombol simpan	Data paket akan tersimpan didalam *database.*	Data paket akan tersimpan didalam *database.*	[√] diterima [ ] ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Data paket masih kosong	Tampilkan pesan kesalahan "*field* tersebut harus diisi"	Dapat menampilkan pesan kesalahan" *field* tersebut harus diisi"	[√] diterima [ ] ditolak
Nama paket tidak diisi	Muncul pesan kesalahan"nama	Dapat menampilkan pesan kesalahan "nama paket	[√] diterima [ ] ditolak

	paket harus diisi"	harus diisi"	
Nama Hotel tidak diisi	Muncul pesan kesalahan"nama hotel harus diisi"	Dapat menampilkan pesan kesalahan "nama hotel harus diisi"	[√] diterima [ ] ditolak
Tempat wisata tidak diisi	Muncul pesan kesalahan"tempat wisata harus diisi"	Dapat menampilkan pesan kesalahan "tempat wisata harus diisi"	[√] diterima [ ] ditolak
Nama Kategori tidak diisi	Muncul pesan kesalahan"nama kategori harus diisi"	Dapat menampilkan pesan kesalahan "nama kategori harus diisi"	[√] diterima [ ] ditolak

4.5.2.13 Pengujian Pengolahan Data Hotel

Berikut dapat dilihat pengujian dari pengolahan data hotel pada tabel 4.21 dibawah ini:

Tabel 4. 21 Pengujian Pengolahan Data Hotel

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik menu hotel	Menampilkan data hotel	Dapat menampilkan data hotel	[√] diterima [ ] ditolak

Klik tombol tambah	Menampilkan *form* tambah hotel	Dapat menampilkan *form* tambah hotel	[√] diterima [ ] ditolak
Klik tombol simpan	Data hotel bertambah	Data hotel bertambah	[√] diterima [ ] ditolak
Klik tombol ubah	Menampilkan *form* ubah provinsi	Dapat menampilkan *form* ubah hotel	[√] diterima [ ] ditolak
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] diterima [ ] ditolak
Data Hotel kosong	Tampilkan pesan kesalahan "Data hotel tidak boleh kosong"	Tampilkan pesan kesalahan "Data hotel tidak boleh kosong"	[√] diterima [ ] ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] diterima [ ] ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] diterima [ ] ditolak

4.5.2.14 Pengujian Tambah Hotel

Berikut dapat dilihat pengujian dari tambah hotel pada tabel 4.22 dibawah ini:

Tabel 4. 22 Pengujian Tambah Hotel

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik tambah hotel	Menampilkan *form* tambah hotel	Dapat menampilkan *form* tambah hotel	[√] diterima [ ] ditolak
Mengisi data hotel Contoh: Nama Hotel : Picollo Hotel	Jika semua field terisi maka akan muncul pesan "data berhasil ditambah"	Jika semua field terisi maka akan muncul pesan "data berhasil ditambah".	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Nama hotel masih kosong	Tampilkan pesan kesalahan "nama hotel harus diisi"	Dapat menampilkan pesan kesalahan " nama hotel harus diisi"	[√] diterima [ ] ditolak

4.5.2.15 Pengujian Pengolahan Data Wisata

Berikut dapat dilihat pengujian dari pengolahan data wisata pada tabel 4.23 dibawah ini:

Tabel 4. 23 Pengujian Pengolahan Data Wisata

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik menu wisata	Menampilkan data wisata	Dapat menampilkan data wisata	[√] diterima [ ] ditolak
Klik tombol tambah	Menampilkan *form* tambah wisata	Dapat menampilkan *form* tambah wisata	[√] diterima [ ] ditolak
Klik tombol simpan	Data wisata bertambah	Data wisata bertambah	[√] diterima [ ] ditolak
Klik tombol ubah	Menampilkan *form* ubah wisata	Dapat menampilkan *form* ubah wisata	[√] diterima [ ] ditolak
Klik tombol simpan	Data berhasil diubah di*database*	Tampil pesan "Data berhasil diubah"	[√] diterima [ ] ditolak

Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] diterima [ ] ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Salah satu *field* kosong	Muncul pesan kesalahan *field* tersebut harus diisi	Dapat muncul pesan kesalahan *field* tersebut harus diisi	[√] diterima [ ] ditolak
Nama Kota kosong	Tampilkan pesan kesalahan "Data wisata tidak boleh kosong"	Tampilkan pesan kesalahan " Data wisata tidak boleh kosong"	[√] diterima [ ] ditolak

4.5.2.16 Pengujian Tambah Wisata

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

Tabel 4. 24 Pengujian Tambah Wisata

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik tambah wisata	Menampilkan *form* tambah wisata	Dapat menampilkan *form* tambah wisata	[√] diterima [ ] ditolak
Mengisi data wisata Contoh: Nama Wisata : Petronas Tower	Jika salah satu field tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa field tersebut harus terisi.	Dapat memunculkan pesan kesalahan bahwa field tersebut harus terisi jika salah satu field tidak terisi atau salah dalam pengisian.	[√] diterima [ ] ditolak
Klik tombol simpan	Data pengiriman akan tersimpan didalam *database*	Data pengiriman akan tersimpan didalam *database* dan muncul pesan ”data berhasil di tambah”	[√] diterima [ ] ditolak
Klik tombol batal	Kembali ke menu sebelumnya	Tampil menu sebelumnya	[√] diterima [ ] ditolak

Kasus dan Hasil Uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Nama wisata masih kosong	Tampilkan pesan kesalahan nama wisata harus disi	Dapat menampilkan pesan kesalahan nama wisata harus disi	[√] diterima [ ] ditolak

4.5.2.17 Pengujian Data Member

Berikut dapat dilihat pengujian dari pengolahan data member pada tabel 4.25 dibawah ini:

Tabel 4. 25 Pengujian Data Member

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik member	Menampilkan data member	Dapat menampilkan data member	[√] diterima [ ] ditolak

4.5.2.18 Pengujian Data Pembayaran

Berikut dapat dilihat pengujian dari lihat pembayaran pada tabel 4.26 dibawah ini:

Tabel 4. 26 Pengujian Data Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik menu pembayaran	Menampilkan *form* pembayaran	Dapat menampilkan *form* lihat pembayaran	[√] diterima [ ] ditolak
Klik Ubah status pembayaran	Menampilkan *form* ubah status pembayaran	Dapat menampilkan *form* ubah status pembayaran	[√] diterima [ ] ditolak

4.5.2.19 Pengujian Pengolahan Data Laporan

Berikut dapat dilihat pengujian dari lihat laporan pada tabel 4.27 di bawah ini:

Tabel 4. 27 Pengujian Pengolahan Data Laporan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengujian**	**Kesimpulan**
Klik menu laporan	Menampilkan *form*	Dapat menampilkan	[√] diterima

	pengisian laporan	*form* pengisian laporan	[ ] ditolak

4.5.3 Kesimpulan Hasil Pengujian Alpa

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

4.6 Pengujian Betha

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan dengan membuat kuesioner mengenai kepuasan *user* dengan kandungan poin syarat *user friendly* untuk selanjutnya dibagikan kepada sebagian *user* dengan mengambil *sample* sebanyak 5 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 sekitar perusahaan

   Penelitian di lingkungan sekitar perusahaan 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 pemesanan paket *tour* secara *online*.

4.6.1 Kuisioner Pengguna

` Kuesioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 10 pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuesioner ini terdiri dari 5 pertanyaan dengan menggunakan skala *likert* dengan skala 1 sampai 10, dengan ketentuan skala untuk tiap pertanyaan sebagai berikut :

4.6.1.1 Kuisioner Pengujian Betha Untuk Pengguna

1. Apakah anda setuju bahwa tampilan website Sentra Tour cukup menarik dan mudah dipahami (user friendly) ?
   a. Sangat setuju
   b. Setuju
   c. Cukup setuju

d. Kurang setuju

e. Tidak setuju

2. Apakah anda setuju bahwa website *e-commerce* ini dapat membantu anda untuk mendapatkan informasi terbaru tentang paket *tour* yang ada di PT. Java Sentra Mandiri Utama?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

   E. Tidak Setuju

3. Apakah anda setuju dengan adanya *website* ini dapat memudahkan *member* dalam melakukan pemesanan/transaksi?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

   E. Tidak Setuju

4. Apakah anda setuju proses pemesanan paket *tour* pada *website* ini mudah dilakukan oleh anda selaku *member*?

   A. Sangat Setuju

   B. Setuju

   C. Cukup Setuju

   D. Kurang Setuju

E. Tidak Setuju

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

A. Sangat Setuju

B. Setuju

C. Cukup Setuju

D. Kurang Setuju

E. Tidak Setuju

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

Keterangan :

R = Banyaknya jawaban responden tiap soal.

Q = Jumlah responden

Y = Nilai persentase

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

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

Tabel 4. 28 Hasil Pengujian Kuisioner Soal No 1

Pertanyaan	**No**	**Keterangan**	**Responder**	**Persentase(%)**
	1.	Sangat Setuju	3	30

1	2.	Setuju	5	50
	3.	Cukup Setuju	2	20
	4.	Kurang Setuju	0	0
	5.	Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil *persentase* diatas maka dapat disimpulkan bahwa sebanyak 3 orang atau 30% mengatakan sangat setuju, 5 orang atau 50% mengatakan setuju dan 2 orang atau 20% mengatakan cukup setuju mengenai tampilan *website*.

2. Apakah anda setuju bahwa website *e-commerce* ini dapat membantu anda untuk mendapatkan informasi tentang paket *tour* yang ada di PT. Java Sentra Mandiri Utama?

Tabel 4. 29 Hasil Pengujian Kuisioner Soal No 2

Pertanyaan	**No**	**Keterangan**	**Responder**	**Persentase(%)**
2	1.	Sangat Setuju	3	30
	2.	Setuju	6	60
	3.	Cukup Setuju	1	10
	4.	Kurang Setuju	0	0
	5.	Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil *persentase* diatas maka dapat disimpulkan bahwa sebanyak 3 orang atau 30% mengatakan sangat setuju, 6 orang atau 60% mengatakan setuju dan 1 orang atau 10% mengatakan cukup setuju mengenai informasi yang didapat dari *website*.

3. Apakah anda setuju dengan adanya *website* ini dapat memudahkan *member* dalam melakukan pemesanan/transaksi?

Tabel 4. 30 Hasil Pengujian Kuisioner Soal No 3

Pertanyaan	**No**	**Keterangan**	**Responder**	**Persentase(%)**
3	1.	Sangat Setuju	3	30
	2.	Setuju	7	70
	3.	Cukup Setuju	0	0
	4.	Kurang Setuju	0	0
	5.	Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil *persentase* diatas maka dapat disimpulkan bahwa sebanyak 3 orang atau 30% mengatakan sangat setuju, 7 orang atau 70% mengatakan setuju mengenai mudahnya pemesanan/transakssi didalam *website* ini.

4. Apakah anda setuju proses pemesanan paket *tour* pada *website* ini mudah dilakukan oleh anda selaku *member*?

Tabel 4. 31 Hasil Pengujian Kuisioner Soal No 4

Pertanyaan	No	Keterangan	Responder	Persentase(%)
4	1.	Sangat Setuju	3	30
	2.	Setuju	2	20
	3.	Cukup Setuju	5	50
	4.	Kurang Setuju	0	0
	5.	Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil *persentase* diatas maka dapat disimpulkan bahwa sebanyak 3 orang atau 30% mengatakan sangat setuju, 2 orang atau 20% mengatakan setuju dan 5 orang atau 50% mengatakan cukup setuju mengenai pemesanan paket *tour* ini.

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

Tabel 4. 32 Hasil Pengujian Kuisioner Soal No 5

Pertanyaan	No	Keterangan	Responder	Persentase(%)
5	1.	Sangat Setuju	2	20
	2.	Setuju	4	40
	3.	Cukup Setuju	4	40
	4.	Kurang Setuju	0	0

	5.	Tidak Setuju	0	0
Jumlah			10	100

Berdasarkan hasil *persentase* diatas maka dapat disimpulkan bahwa sebanyak 2 orang atau 20% mengatakan sangat setuju, 4 orang atau 40% mengatakan setuju dan 4 orang atau 40% mengatakan cukup setuju mengenai sistem pembayaran dalam *website*.

4.6.2 Kesimpulan Pengujian Betha

Berdasarkan hasil *persentase* diatas yang didapatkan dari pengujian betha pengguna, maka sistem penerapan *E-commerce* untuk system ini sudah sesuai dengan tujuan yaitu PT. Java Sentra Mandiri Utama dapat mepermudah dalam proses promosi paket jasa *tour* yang ditawarkan, memudahkan masyarakat dalam mencari informasi mengenai jasa *tour* yang ditawarkan, pelanggan dapat melakukan transaksi pemesanan secara *online*, tampilan bersifat *user friendly*, memudahkan dalam pengelolaan data dan memudahkan dalam membuat laporan.