BAB 4
IMPLEMENTASI DAN PENGUJIAN

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 serta merupakan tahap untuk menerjemahkan perancangan berdasarkan hasil analisis dalam bahasa yang dapat dimengerti oleh mesin, yang terdiri dari penjelasan mengenai lingkungan implementasi, dan implementasi program. Wujud dari hasil tahap implementasi ini nantinya adalah sebuah sistem yang siap untuk diuji dan digunakan.

4.1.1 Perangkat Keras yang Digunakan

Spesifikasi perangkat keras (*hardware*) yang digunakan dalam pembangunan aplikasi *e-commerce* di Toko AC Jaya Abadi dapat dilihat pada tabel 4.1.

Tabel 4.1 Perangkat Keras yang Digunakan

Perangkat Keras	Keterangan
Processor	Intel(R) Pentium (R) 4 CPU 3.20 GHz
Memory	DDR2 1GB
Harddisk	SATA 500 GB
Video Card	Nvidia 6800 XT 256MB
Monitor	Resolusi 1360x768
Keyboard, Mouse	DELL

4.1.2 Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak (*software*) yang digunakan dalam pembangunan aplikasi *e-commerce* di Toko AC Jaya Abadi dapat dilihat pada tabel 4.2.

Tabel 4.2 Perangkat Lunak yang Digunakan

Perangkat Lunak	Keterangan
Windows XP SP3 Professional	Sebagai Sistem Operasi
PHP	Sebagai Bahasa Pemrograman
MySQL	Sebagai *Database*
PhpMyAdmin	Sebagai *Interface Database*
Wamp Server 2.0	Sebagai *Web Server Local*
Adobe Dreamweaver CS3	Sebagai *Code Editor*
Mozilla Firefox	Sebagai *Web Browser*
Microsoft Visio 2007	Memodelkan analisis perangkat lunak

4.2 Implementasi Basis Data

Pada tahap awal pembangunan sebuah aplikasi, dibutuhkan media penyimpanan data (*database*). Implementasi *database* untuk aplikasi *e-commerce* di Toko AC Jaya Abadi ini adalah sebagai berikut.

1. Tabel Admin

```
CREATE TABLE IF NOT EXISTS `tbladmin` (
  `id` varchar(5) NOT NULL DEFAULT '0',
  `tipepegawai ` varchar(50) NOT NULL,
  `nama_pegawai` varchar(100) NOT NULL,
  `email` varchar(100) NOT NULL,
  `username` varchar(100) NOT NULL,
```

```
    `password` varchar(50) NOT NULL,

    `telp1` varchar(20) NOT NULL,

    `telp2` varchar(20) NOT NULL,

    `status` tinyint(1) NOT NULL DEFAULT '0',

    `dibuat` datetime NOT NULL,

    `terakhir_dirubah` datetime NOT NULL,

    `login_terakhir` datetime NOT NULL,

    PRIMARY KEY (`id`),

    UNIQUE KEY `email` (`email`),

    UNIQUE KEY `username` (`username`),

    KEY `fk_tipepegawai` (`tipepegawai_id`),

  ADD CONSTRAINT `tblpegawai_ibfk_1` FOREIGN KEY (`tipepegawai_id`) REFERENCES `tblpegawaitipe` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

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

2. Tabel Bank

```
  CREATE TABLE IF NOT EXISTS `tblbank` (

    `id` varchar(5) NOT NULL,

    `nama_bank` varchar(100) NOT NULL,

    `no_rek` varchar(50) NOT NULL,

    `nama_pemilik` varchar(50) NOT NULL,

    `gambar` varchar(200) NOT NULL,

    `status` tinyint(1) NOT NULL DEFAULT '0',
```

```
        PRIMARY KEY (`id`)

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

3. Tabel Detail Pesanan

```
        CREATE TABLE IF NOT EXISTS `tbldetailpesanan` (

         `id` varchar(6) NOT NULL,

         `produk_id` varchar(5) NOT NULL,

         `pesanan_id` varchar(6) NOT NULL,

         `qty_pesanan` int(11) NOT NULL,

         `harga` decimal(10,2) NOT NULL,

         `diskon` int(11) NOT NULL,

         `berat` float NOT NULL,

         PRIMARY KEY (`id`),

         KEY `fk_pesanan` (`pesanan_id`),

         KEY `fk_produk` (`produk_id`),

        ADD CONSTRAINT `tbldetailpesanan_ibfk_2` FOREIGN KEY (`produk_id`) REFERENCES `tblproduk` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

        ADD CONSTRAINT `tbldetailpesanan_ibfk_1` FOREIGN KEY (`pesanan_id`) REFERENCES `tblpesanan` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

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

4. Tabel Detail Retur

```
CREATE TABLE IF NOT EXISTS `tbldetailretur` (

 `id` int(11) NOT NULL AUTO_INCREMENT,

 `retur_id` int(11) NOT NULL,

 `produk_id` varchar(5) NOT NULL,

 `jumlah` int(11) NOT NULL,

 `alasan` enum('TIDAK SESUAI PESANAN','KABEL PUTUS') NOT NULL,

 PRIMARY KEY (`id`),

 KEY `fk_retur` (`retur_id`),

 KEY `fk_produk` (`produk_id`),

ADD CONSTRAINT `tbldetailretur_ibfk_2` FOREIGN KEY (`produk_id`) REFERENCES `tblproduk` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

ADD CONSTRAINT `tbldetailretur_ibfk_1` FOREIGN KEY (`retur_id`) REFERENCES `tblretur` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

5. Tabel Jasa Pengiriman

```
CREATE TABLE IF NOT EXISTS `tbljasapengiriman` (

 `id` varchar(5) NOT NULL,

 `nama_jasa` varchar(50) NOT NULL,

 `status` tinyint(1) NOT NULL DEFAULT '0',

 PRIMARY KEY (`id`)

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

6. Tabel Jenis Pengiriman

```
CREATE TABLE IF NOT EXISTS `tbljenispengiriman` (

 `id` varchar(5) NOT NULL,

 `jasa_id` varchar(5) NOT NULL,

 `nama_paket` varchar(50) NOT NULL,

 `status` tinyint(1) NOT NULL,

 PRIMARY KEY (`id`),

 KEY `fk_jasapengiriman` (`jasa_id`),

ADD CONSTRAINT `tbljenispengiriman_ibfk_1` FOREIGN KEY (`jasa_id`) REFERENCES `tbljasapengiriman` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

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

7. Tabel Kabkota

```
CREATE TABLE IF NOT EXISTS `tblkabkota` (

 `prov_id` int(11) NOT NULL,

 `id` int(11) NOT NULL,

 `nama_kabkota` varchar(100) NOT NULL,

 `status` tinyint(1) NOT NULL DEFAULT '0',

 PRIMARY KEY (`id`),

 KEY `fk_prov` (`prov_id`),

ADD CONSTRAINT `tblkabkota_ibfk_1` FOREIGN KEY (`prov_id`) REFERENCES `tblprovinsi` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

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

8. Tabel Merk

```
CREATE TABLE IF NOT EXISTS `tblmerk` (
  `id` varchar(5) NOT NULL,
  `nama_merk` varchar(50) NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT '0',
  `dibuat` datetime NOT NULL,
  `terakhir_dirubah` datetime NOT NULL,
  PRIMARY KEY (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

9. Tabel Ongkos

```
CREATE TABLE IF NOT EXISTS `tblongkos` (
  `id` varchar(5) NOT NULL,
  `jasa_id` varchar(5) NOT NULL,
  `jenis_id` varchar(5) NOT NULL,
  `kabkota_id` int(11) NOT NULL,
  `ongkos_kirim` int(11) NOT NULL,
  `status` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`id`),
  KEY `fk_jenispengiriman` (`jenis_id`),
  KEY `fk_kabkota` (`kabkota_id`),
ADD CONSTRAINT `tblongkos_ibfk_4` FOREIGN KEY (`jenis_id`) REFERENCES `tbljenispengiriman` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
ADD CONSTRAINT `tblongkos_ibfk_5` FOREIGN KEY (`kabkota_id`)
```

```
        REFERENCES `tblkabkota` (`id`) ON DELETE NO ACTION ON
        UPDATE NO ACTION

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

10. Tabel Pelanggan

```
        CREATE TABLE IF NOT EXISTS `tblpelanggan` (

          `id` varchar(5) NOT NULL,

          `no_identitas` varchar(50) NOT NULL,

          `nama` varchar(100) NOT NULL,

          `username` varchar(100) NOT NULL,

          `password` varchar(50) NOT NULL,

          `pass_key` varchar(32) NOT NULL,

          `email` varchar(100) NOT NULL,

          `alamat` varchar(200) NOT NULL,

          `prov_id` int(11) NOT NULL,

          `kabkota_id` int(11) NOT NULL,

          `kodepos` int(5) NOT NULL,

          `telp1` varchar(20) NOT NULL,

          `telp2` varchar(20) NOT NULL,

          `status` tinyint(1) NOT NULL,

          `dibuat` datetime NOT NULL,

          `terakhir_dirubah` datetime NOT NULL,

          `login_terakhir` datetime NOT NULL,
```

```
        PRIMARY KEY (`id`)

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

11. Tabel Pembayaran

```
        CREATE TABLE IF NOT EXISTS `tblpembayaran` (

         `id` varchar(6) NOT NULL,

         `pesanan_id` varchar(6) NOT NULL,

         `bank_id` varchar(5) DEFAULT NULL,

         `tgl_bayar` date NOT NULL,

         `jumlah_bayar` decimal(10,2) NOT NULL,

         `jenis_pembayaran` varchar(50) NOT NULL,

         `atas_nama` varchar(50) NOT NULL,

         `no_transaksi` varchar(50) NOT NULL,

         `no_rekening` varchar(45) NOT NULL,

         `id_paypal` varchar(45) NOT NULL,

         `bank_asal` varchar(45) NOT NULL,

         PRIMARY KEY (`id`),

         KEY `fk_pesanan` (`pesanan_id`),

         KEY `fk_bank` (`bank_id`),

        ADD CONSTRAINT `tblpembayaran_ibfk_1` FOREIGN KEY (`pesanan_id`) REFERENCES `tblpesanan` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

        ADD CONSTRAINT `tblpembayaran_ibfk_2` FOREIGN KEY (`bank_id`) REFERENCES `tblbank` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
```

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

12. Tabel Pesanan

```
CREATE TABLE IF NOT EXISTS `tblpesanan` (

  `id` varchar(6) NOT NULL,

  `pelanggan_id` varchar(5) NOT NULL,

  `ongkos_id` varchar(5) NOT NULL,

  `tgl_dipesan` datetime NOT NULL,

  `tgl_expired` datetime NOT NULL,

  `tgl_diterima` datetime DEFAULT NULL,

  `tarif_ongkos_kirim` int(20) NOT NULL,

  `nama_penerima` varchar(50) NOT NULL,

  `alamat_penerima` text NOT NULL,

  `telp_penerima` varchar(15) NOT NULL,

  `status_pengiriman` enum('Diterima','Sudah Dikirim','Belum Dikirim') NOT NULL DEFAULT 'Belum Dikirim',

  `status_pesanan` enum('Dipesan','Dibatalkan','Dibayar','Konfirmasi','Direturkan') NOT NULL DEFAULT 'Dipesan',

  `total_berat` float NOT NULL,

  `total_bayar` int(15) DEFAULT NULL,

  `no_resi` varchar(20) DEFAULT NULL,

  PRIMARY KEY (`id`),

  KEY `fk_pelanggan` (`pelanggan_id`),
```

```
        KEY `fk_ongkos` (`ongkos_id`),

        ADD CONSTRAINT `tblpesanan_ibfk_1` FOREIGN KEY (`pelanggan_id`) REFERENCES `tblpelanggan` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,

        ADD CONSTRAINT `tblpesanan_ibfk_2` FOREIGN KEY (`ongkos_id`) REFERENCES `tblongkos` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

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

13. Tabel Produk

```
        CREATE TABLE IF NOT EXISTS `tblproduk` (

         `id` varchar(5) NOT NULL,

         `kategori_id` varchar(5) NOT NULL,

         `subkategori_id` varchar(5) NOT NULL,

         `spec_id` varchar(5) NOT NULL,

         `merk_id` varchar(5) NOT NULL,

         `nama_tipe` varchar(100) NOT NULL,

         `berat` int(11) NOT NULL,

         `ketersediaan` int(11) NOT NULL,

         `gambar` varchar(200) NOT NULL,

         `harga` int(11) NOT NULL,

         `diskon` varchar(5) NOT NULL,

         `dilihat` int(11) NOT NULL,

         `deskripsi` varchar(200) NOT NULL,

         `status` tinyint(1) NOT NULL DEFAULT '0',
```

```
    `dibuat` datetime NOT NULL,

    `terakhir_dirubah` datetime NOT NULL,

    PRIMARY KEY (`id`),

    KEY `fk_kategori` (`kategori_id`),

    KEY `fk_subkategori` (`subkategori_id`),

    KEY `fk_spec` (`spec_id`),

    KEY `fk_merk` (`merk_id`),

  ADD CONSTRAINT `tblproduk_ibfk_5` FOREIGN KEY (`kategori_id`)
  REFERENCES `tblkategori` (`id`) ON DELETE NO ACTION ON
  UPDATE NO ACTION,

  ADD CONSTRAINT `tblproduk_ibfk_6` FOREIGN KEY
  (`subkategori_id`) REFERENCES `tblsubkategori` (`id`) ON DELETE
  NO ACTION ON UPDATE NO ACTION,

  ADD CONSTRAINT `tblproduk_ibfk_7` FOREIGN KEY (`spec_id`)
  REFERENCES `tblspec` (`id`) ON DELETE NO ACTION ON UPDATE
  NO ACTION,

  ADD CONSTRAINT `tblproduk_ibfk_8` FOREIGN KEY (`merk_id`)
  REFERENCES `tblmerk` (`id`) ON DELETE NO ACTION ON UPDATE
  NO ACTION

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

14. Tabel Provinsi

```
  CREATE TABLE IF NOT EXISTS `tblprovinsi` (

    `id` int(11) NOT NULL,

    `nama_provinsi` varchar(100) NOT NULL,
```

```
  `status` tinyint(1) NOT NULL DEFAULT '0',

  PRIMARY KEY (`id`)

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

15. Tabel Retur

```
CREATE TABLE IF NOT EXISTS `tblretur` (

  `id` int(11) NOT NULL AUTO_INCREMENT,

  `pelanggan_id` varchar(5) NOT NULL,

  `pesanan_id` varchar(6) NOT NULL,

  `tgl_retur` date NOT NULL,

  `nama_penerima` varchar(50) NOT NULL,

  `alamat_penerima` text NOT NULL,

  `status_retur` enum('Belum Disetujui','Sudah Disetujui','Tidak Disetujui') NOT NULL,

  `status_pengiriman_retur` enum('Belum Dikirim','Sudah Dikirim','Diterima') NOT NULL,

  `no_resi` varchar(30) NOT NULL,

  `no_resi_kirim` varchar(30) NOT NULL,

  `status_pengembalian_retur` enum('Belum Dikirim','Sudah Dikirim','Diterima') NOT NULL,

  PRIMARY KEY (`id`),

  KEY `fk_pelanggan` (`pelanggan_id`),

  KEY `fk_pesanan` (`pesanan_id`),

ADD CONSTRAINT `tblretur_ibfk_2` FOREIGN KEY (`pesanan_id`) REFERENCES `tblpesanan` (`id`) ON DELETE NO ACTION ON
```

```
        UPDATE NO ACTION,

        ADD CONSTRAINT `tblretur_ibfk_1` FOREIGN KEY (`pelanggan_id`)
        REFERENCES `tblpelanggan` (`id`) ON DELETE NO ACTION ON
        UPDATE NO ACTION

) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

16. Tabel Spec

```
        CREATE TABLE IF NOT EXISTS `tblspec` (

          `id` varchar(5) NOT NULL,

          `subkategori_id` varchar(5) NOT NULL,

          `nama_spec` varchar(50) NOT NULL,

          `status` tinyint(1) NOT NULL,

          `dibuat` datetime NOT NULL,

          `terakhir_dirubah` datetime NOT NULL,

          PRIMARY KEY (`id`),

          KEY `fk_subkategori` (`subkategori_id`),

        ADD CONSTRAINT `tblspec_ibfk_4` FOREIGN KEY (`subkategori_id`)
        REFERENCES `tblsubkategori` (`id`) ON DELETE NO ACTION ON
        UPDATE NO ACTION

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

17. Tabel Subkategori

```
        CREATE TABLE IF NOT EXISTS `tblsubkategori` (
```

```
    `id` varchar(5) NOT NULL,

    `kategori_id` varchar(5) NOT NULL,

    `nama_subkat` varchar(50) NOT NULL,

    `status` tinyint(1) NOT NULL DEFAULT '0',

    `dibuat` datetime NOT NULL,

    `terakhir_dirubah` datetime NOT NULL,

    PRIMARY KEY (`id`),

    KEY `fk_kategori` (`kategori_id`),

  ADD CONSTRAINT `tblsubkategori_ibfk_1` FOREIGN KEY (`kategori_id`) REFERENCES `tblkategori` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

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

18. Tabel Kategori

```
  CREATE TABLE IF NOT EXISTS `tblkategori` (

    `id` varchar(5) NOT NULL,

    `nama_kat` varchar(50) NOT NULL,

    `status` tinyint(1) NOT NULL DEFAULT '0',

    `dibuat` datetime NOT NULL,

    `terakhir_dirubah` datetime NOT NULL,

    PRIMARY KEY (`id`)

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

19. Tabel Keranjang

```
CREATE TABLE IF NOT EXISTS `tblkategori` (

 `id` varchar(5) NOT NULL,

 `nama_kat` varchar(50) NOT NULL,

 `status` tinyint(1) NOT NULL DEFAULT '0',

 `dibuat` datetime NOT NULL,

 `terakhir_dirubah` datetime NOT NULL,

 PRIMARY KEY (`id`)

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

20. Tabel Kontak YM

```
CREATE TABLE IF NOT EXISTS `tblkontakym` (

 `id` varchar(5) NOT NULL,

 `pegawai_id` varchar(5) NOT NULL,

 `id_ym` varchar(100) NOT NULL,

 `nama_kontak` varchar(50) NOT NULL,

 `status` tinyint(1) NOT NULL DEFAULT '0',

 PRIMARY KEY (`id`),

 KEY `fk_pegawai` (`pegawai_id`),

ADD CONSTRAINT `tblkontakym_ibfk_1` FOREIGN KEY (`pegawai_id`) REFERENCES `tblpegawai` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION

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

21. Tabel Mata Uang

```
CREATE TABLE IF NOT EXISTS `tblmatauang` (
  `terakhir_dirubah` datetime NOT NULL,
  `kode` varchar(5) NOT NULL,
  `kurs` decimal(10,2) NOT NULL,
  PRIMARY KEY (`terakhir_dirubah`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.3 Implementasi Antarmuka

Implementasi antarmuka dilakukan pada 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* di Toko AC Jaya Abadi berbasis *web* akan dilampirkan pada halaman lampiran.

Tabel 4.3 Implementasi Antarmuka Administrator

Menu	Deskripsi	Nama File
Login	Digunakan sebagai halaman login administrator	index.php
Beranda	Merupakan tampilan beranda administrator	menuadmin.php
Edit data profil	Digunakan untuk merubah profil administrator	mnuedit1.php
Ubah password	Digunakan untuk merubah password administrator	lupapasswordadmin.php
Data admin	Digunakan administrator untuk melihat, menambah, menghapus	menupegawai.php

	dan mencari data petugas	
Data pelanggan	Digunakan administrator untuk melihat, menambah, menghapus dan mencari data pelanggan	menulistpelanggan.php
Data kategori	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data kategori	menukategori.php
Data subkategori	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data subkategori	menusubkategori.php
Data spesifikasi	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data spesifikasi	menuspec.php
Data produk	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data produk serta melihat data detail produk	menuprod.php
Data Provinsi	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data provinsi	menuprovinsi.php
Data Kab/kota	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data kab/kota	kabkota_list.php
Data jasa pengiriman	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data jasa pengiriman	menujasa.php
Data jenis pengiriman	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data jenis pengiriman	menujenis.php
Data ongkos kirim	Digunakan untuk melihat, menambah, merubah, menghapus dan mencari data ongkos kirim	menuongkir.php
Data retur	Digunakan untuk mengolah data retur barang	menukonfretur.php
Data pembayaran	Digunakan untuk mengolah data transaksi yaitu data pemesanan	menukonfpembayaran.php

	dan data pembayaran	
Data pengiriman	Digunakan untuk mengolah data transaksi yaitu data pemesanan dan data pengiriman	menukonfpengiriman.php
Laporan produk	Digunakan untuk melihat laporan produk per-tanggal, per-bulan dan mencetak laporan	lapproduk.php
Laporan penjualan	Digunakan untuk melihat laporan penjualan per-tanggal, per-bulan dan mencetak laporan	lappenjualan.php
Logout	Digunakan administrator atau petugas untuk keluar dari aplikasi	logout.php

Tabel 4.4 Implementasi Antarmuka Pelanggan dan Pengunjung

Menu	Deskripsi	Nama File
Login	Digunakan pelanggan sebagai halaman login	menulogin.php
Lupa password	Digunakan jika pelanggan lupa password untuk login	lupapass.php
Beranda	Merupakan tampilan beranda pelanggan dan pengunjung	index.php
Edit data profil	Digunakan untuk merubah profil pelanggan	editprofil.php
Ubah password	Digunakan untuk merubah password pelanggan	ubahpass.php
Tentang kami	Digunakan pelanggan dan pengunjung untuk melihat profil perusahaan	menutentangkami.php
Aturan	Digunakan pelanggan dan pengunjung untuk melihat cara pemesanan dan pembayaran	menuaturan.php
Riwayat transaksi	Digunakan pelanggan untuk melihat history transaksi dan digunakan untuk melakukan pembayaran	menupembayaran.php
Retur	Digunakan pelanggan untuk mereturkan pesanan	menuretur.php
Daftar	Digunakan pengunjung yang ingin mendaftar menjadi *member*	menuregistrasi.php
Detail Produk	Digunakan pelanggan dan pengunjung untuk melihat data detail produk yang akan dibeli	menutambahproduk.php
Keranjang Belanja	Digunakan pelanggan untuk melihat, merubah, menghapus data produk yang dipesan	menukeranjang.php
Isi data pengiriman	Digunakan pelanggan untuk mengisi data penerima pesanan	isidatatransaksi.php
Selesai Transaksi	Digunakan pelanggan untuk menampilkan data pesanan, mencetak nota pesanan dan pembayaran *PayPal*	menusimpantransaksi.php
Logout	Digunakan pelanggan untuk	logout.php

	keluar dari halaman pelanggan	

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 Alpha

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

4.4.1.1 Kasus dan Hasil Pengujian

1. Pengujian Pendaftaran Pelanggan

Berikut ini merupakan pengujian pendaftaran pelanggan dapat dilihat pada tabel 4.5 dibawah ini.

Tabel 4.5 Pengujian Pendaftaran Pelanggan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data pendaftaran, contoh : nama lengkap : muhamad ichsan no identitas : 123456789012345 *username* : ichan *email* : deathschyte62@gmail.com password : 54321 ulang password : 54321 alamat : JL. SMEA VI No. 6 provinsi : Jakarta kab/kota : Jakarta Timur kodepos : 13630	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftaran	Form pendaftaran terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses pendaftaran	[√] Diterima [ ] Ditolak

nomor telp1: 085722555373 nomor telp2 : 085722555373			
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data nama lengkap kosong	Muncul pesan kesalahan "Nama lengkap harus diisi"	Dapat menampilkan pesan kesalahan "Nama lengkap harus diisi"	[√] Diterima [ ] Ditolak
Data no identitas kosong	Muncul pesan kesalahan "nomor identitas harus diisi"	Dapat menampilkan pesan kesalahan "nomor identitas harus diisi"	[√] Diterima [ ] Ditolak
Data *username* kosong	Muncul pesan kesalahan "*Username* harus diisi"	Dapat menampilkan pesan kesalahan "*Username* harus diisi"	[√] Diterima [ ] Ditolak
Data *email* kosong	Muncul pesan kesalahan "*Email* harus diisi"	Dapat menampilkan pesan kesalahan "*Email* harus diisi"	[√] Diterima [ ] Ditolak
Data password kosong	Muncul pesan kesalahan "Password harus diisi"	Dapat menampilkan pesan kesalahan "Password harus diisi"	[√] Diterima [ ] Ditolak
Data ulang password	Muncul pesan kesalahan "Ulang password harus diisi"	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

		"Ulang password harus diisi"	
Data alamat kosong	Muncul pesan kesalahan "Alamat harus diisi"	Dapat menampilkan pesan kesalahan "Alamat harus diisi"	[√] Diterima [ ] Ditolak
Data propinsi kosong	Muncul pesan kesalahan "Propinsi harus diisi"	Dapat menampilkan pesan kesalahan "Propinsi harus diisi"	[√] Diterima [ ] Ditolak
Data kab/kota kosong	Muncul pesan kesalahan "Kab/kota harus diisi"	Dapat menampilkan pesan kesalahan "Kab/kota harus diisi"	[√] Diterima [ ] Ditolak
Data kodepos kosong	Muncul pesan kesalahan "Kodepos harus diisi"	Dapat menampilkan pesan kesalahan "Kodepos harus diisi"	[√] Diterima [ ] Ditolak
Data nomor telp1 kosong	Muncul pesan kesalahan "Nomor *handphone* harus diisi"	Dapat menampilkan pesan kesalahan "Nomor *handphone* harus diisi"	[√] Diterima [ ] Ditolak
Data nomor telp2 kosong	Muncul pesan kesalahan "Nomor *handphone* harus diisi"	Dapat menampilkan pesan kesalahan "Nomor *handphone* harus diisi"	[√] Diterima [ ] Ditolak

2. Pengujian Login Pelanggan

Berikut ini merupakan pengujian login pelanggan dapat dilihat pada tabel 4.6 dibawah ini.

Tabel 4.6 Pengujian Login Pelanggan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan alamat email: deathschyte62@gmail.com	Mengisikan data alamat email pada *field* alamat email	Dapat mengisi *field* alamat email pada *field* alamat email	[√] Diterima [ ] Ditolak
Memasukkan password : 54321	Mengisikan data password pada *field* password	Dapat mengisi *field* password pada *field* password	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengosongkan data email atau data password	Menampilkan pesan kesalahan "Email dan Password Harus Diisi"	Dapat menampilkan pesan kesalahan "Email dan Password Harus Diisi"	[√] Diterima [ ] Ditolak
Data *email* tidak terdaftar	Menampilkan pesan kesalahan "Email Salah"	Dapat menampilkan pesan kesalahan "Email Salah"	[√] Diterima [ ] Ditolak
Data password tidak terdaftar	Menampilkan pesan kesalahan "Password Salah"	Dapat menampilkan pesan kesalahan "Password Salah"	[√] Diterima [ ] Ditolak

3. Pengujian Lupa Password Pelanggan

Berikut ini merupakan pengujian lupa password pelanggan dapat dilihat pada tabel 4.7 dibawah ini.

Tabel 4.7 Pengujian Lupa Password Pelanggan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data email dengan benar deathschyte62@gmail.com	Data email dapat diproses	Data email dapat diproses	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data email tidak diisi	Menampilkan pesan kesalahan "Email Harus Diisi"	Dapat menampilkan pesan kesalahan "Email Harus Diisi"	[√] Diterima [ ] Ditolak
Data email tidak terdaftar "abcde@gmail.com"	Menampilkan pesan kesalahan "Email yang Anda Masukkan Tidak Terdaftar"	Dapat menampilkan pesan kesalahan "Email yang Anda Masukkan Tidak Terdaftar"	[√] Diterima [ ] Ditolak

4. Pengujian Ubah Profil Pelanggan

Berikut ini merupakan pengujian uba profil pelanggan dapat dilihat pada tabel 4.8 dibawah ini.

Tabel 4.8 Pengujian Ubah Profil Pelanggan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan nama lengkap : muhamad ichsan	Mengisi data nama lengkap pada *field* nama lengkap	Dapat mengisi *field* nama lengkap	[√] Diterima [ ] Ditolak
Memasukkan *username* : Ichan	Mengisi data *username* pada *field username*	Dapat mengisi *field username*	[√] Diterima [ ] Ditolak
Memasukkan email : deathschyte62@gmail.com	Mengisi data email pada *field* email	Dapat mengisi *field* email	[√] Diterima [ ] Ditolak
Memasukkan alamat : TUISRA No.9	Mengisi data alamat pada *field* alamat	Dapat mengisi *field* alamat	[√] Diterima [ ] Ditolak
Memasukkan propinsi : Jawa Barat	Mengisi data propinsi pada *field* propinsi	Dapat mengisi *field* propinsi	[√] Diterima [ ] Ditolak
Memasukkan kab/kota : Bandung	Mengisi data kab/kota pada *field* kab/kota	Dapat mengisi *field* kab/kota	[√] Diterima [ ] Ditolak
Memasukkan kodepos : 40134	Mengisi data kodepos pada *field* kodepos	Dapat mengisi *field* kodepos	[√] Diterima [ ] Ditolak
Memasukkan nomor *handphone* : 0813446699331	Mengisi data nomor *handphone* pada *field* nomor *handphone*	Dapat mengisi *field* nomor *handphone*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama lengkap kosong	Tampil pesan "Nama Lengkap Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak
Username kosong	Tampil pesan "*Username* Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

Email kosong	Tampil pesan "Email Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak
Alamat kosong	Tampil pesan "Alamat Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak
Propinsi kosong	Tampil pesan "Propinsi Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak
Kab/kota kosong	Tampil pesan "Kab/kota Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak
Kodepos kosong	Tampil pesan "Kodepos Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak
Nomor *handphone* kosong	Tampil pesan "Nomor *handphone* Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

5. Pengujian Ubah Password Pelanggan

Berikut ini merupakan pengujian ubah password pelanggan dapat dilihat pada tabel 4.9 dibawah ini.

Tabel 4.9 Pengujian Ubah Password Pelanggan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data email dengan benar deathschyte62@gmail.com	Data email dapat diproses	Data email dapat diproses	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Data email tidak diisi/ memasukan email yang tak terdaftar	Menampilkan pesan kesalahan "Email Tersebut	Dapat menampilkan pesan kesalahan "Email	[√] Diterima [ ] Ditolak

	Tidak Terdaftar Didatabase Kami”	Tersebut Tidak Terdaftar Didatabase Kami”	

6. Pengujian Keranjang Belanja

Berikut ini merupakan pengujian keranjang belanja dapat dilihat pada tabel 4.10 dibawah ini.

Tabel 4.10 Pengujian Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memilih produk yang akan dipesan	Data produk yang dipesan tersimpan di keranjang	Dapat menyimpan data produk yang dipesan ke kreanjang belanja	[√] Diterima [ ] Ditolak
Menampilkan data jumlah pesanan	Tercantum pada *textbox* data yang akan diubah	Dapat mengisi jumlah data dengan benar	[√] Diterima [ ] Ditolak
Mengubah jumlah pemesanan	Data jumlah yang diubah tersimpan dalam data keranjang	Dapat menampilkan perubahan data jumlah	[√] Diterima [ ] Ditolak
Menghapus data pemesanan	Data yang dihapus terhapus dari keranjang belanja	Dapat menghapus data pada keranjang belanja	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan jumlah pemesanan produk	Melebihi stok yang ada akan muncul pesan kesalahan “Maaf Jumlah yang Dipesan Melebihi Stok yang Ada”	Dapat menampilkan pesan kesalahan “Maaf Jumlah yang Dipesan Melebihi Stok yang Ada”	[√] Diterima [ ] Ditolak

7. Pengujian Isi Data Pengiriman

Berikut ini merupakan pengujian isi data pengiriman dapat dilihat pada tabel 4.11 dibawah ini.

Tabel 4.11 Pengujian Isi Data Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data pengiriman	Data pengiriman baru akan tersimpan di dalam *database*	Dapat menyimpan data pengiriman baru di dalam *database*	[√] Diterima [ ] Ditolak
Proses data pengiriman	Data alamat pengiriman sama dengan data alamat pelanggan yang ada dalam *database*	Dapat menyimpan data pengiriman dalam *database*	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengosongkan salah satu *field*	Menampilkan pesan kesalahan "Field Harus Diisi"	Dapat menampilkan pesan kesalahan "Field Harus Diisi"	[√] Diterima [ ] Ditolak

8. Pengujian Konfirmasi Pembayaran

Berikut ini merupakan pengujian konfirmasi pembayaran dapat dilihat pada tabel 4.12 dibawah ini.

Tabel 4.12 Pengujian Konfirmasi Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Masukkan data konfirmasi pembayaran contoh : jenis pembayaran : rekening tujuan nomor rekening : BCA No. Rekening 280-354-1302 transfer dari bank : Bank BCA no rek : 0131442281 atas nama : yordie uang yang di transfer : 100.000	Jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* tersebut harus terisi.	Dapat memunculkan jika salah satu *field* tidak terisi atau salah dalam pengisian, maka akan muncul pesan kesalahan bahwa *field* tersebut harus terisi.	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengosongkan salah satu *field*	Muncul pesan kesalahan "*Field* Harus Diisi"	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

9. Pengujian Retur Barang

Berikut ini merupakan pengujian klaim retur barang dapat dilihat pada tabel 4.13 dibawah ini.

Tabel 4.13 Pengujian Retur Barang

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memilih nomor pesanan yang akan direturkan : PD002	Menampilkan data detail pesanan yang akan direturkan	Dapat menampilkan data detail pesanan	[√] Diterima [ ] Ditolak
Mengubah jumlah produk yang akan diretur	Data jumlah yang diubah tersimpan dalam data detail retur barang	Dapat menampilkan perubahan data jumlah	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan data jumlah retur melebihi jumlah pesanan	Menampilkan pesan kesalahan “Maaf produk yang diretur melebihi jumlah pesanan”	Dapat menampilkan pesan kesalahan “Maaf produk yang diretur melebihi jumlah pesanan”	[√] Diterima [ ] Ditolak

10. Pengujian Logout Pelanggan

Berikut ini merupakan pengujian logout pelanggan dapat dilihat pada tabel 4.14 dibawah ini.

Tabel 4.14 Pengujian Logout Pelanggan

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Session email	Dapat keluar dan menghapus session pengguna	*User* keluar dari aplikasi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Session email	Tidak dapat keluar dan menghapus session pengguna	*User* tidak dapat keluar	[√] Diterima [ ] Ditolak

11. Pengujian Login Administrator

Berikut ini merupakan pengujian aplikasi *login* administrator dapat dilihat pada tabel 4.15 dibawah ini.

Tabel 4.15 Pengujian Login Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan *username* : admin	Mengisikan data *username* pada *field username*	Dapat mengisi *field username* pada field *username*	[√] Diterima [ ] Ditolak
Memasukkan password : admin	Mengisikan data password pada *field* password	Dapat mengisi *field* password pada *field* password	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan data *username* dan password yang tidak terdaftar	Tampilkan pesan kesalahan "Login *Invalid*"	Dapat menampilkan pesan kesalahan "Login *Invalid*"	[√] Diterima [ ] Ditolak
Mengosongkan data *username* dan password	Tampilkan pesan kesalahan "Login *Invalid*"	Dapat menampilkan pesan kesalahan "Login *Invalid*"	[√] Diterima [ ] Ditolak
Mengosongkan *username*	Muncul pesan kesalahan	Muncul pesan kesalahan	[√] Diterima

	"*Username* atau Password yang Anda Masukkan Salah"	"*Username* atau Password yang Anda Masukkan Salah"	[ ] Ditolak
Mengosongkan password	Muncul pesan kesalahan "*Username* atau Password yang Anda Masukkan Salah"	Muncul pesan kesalahan "*Username* atau Password yang Anda Masukkan Salah"	[√] Diterima [ ] Ditolak

12. Pengujian Ubah Profil Administrator

Berikut ini merupakan pengujian aplikasi ubah profil administrator dapat dilihat pada tabel 4.16 dibawah ini.

Tabel 4.16 Pengujian Ubah Profil Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan nama lengkap : Administrator	Mengisi nama lengkap pada *field* nama lengkap	Dapat mengisi *field* nama lengkap	[√] Diterima [ ] Ditolak
Memasukkan *username* : Administrator	Mengisi username pada *field username*	Dapat mengisi *field username*	[√] Diterima [ ] Ditolak
Memasukkan email : admin@yahoo.com	Mengisi email pada *field* email	Dapat mengisi *field* email	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Nama lengkap kosong	Tampil pesan "Nama Lengkap Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak
Username kosong	Tampil pesan "*Username* Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

Email kosong	Tampil pesan "Email Harus Diisi"	Tampil pesan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

13. Pengujian Ubah Password Administrator

Berikut ini merupakan pengujian aplikasi ubah password administrator dapat dilihat pada tabel 4.17 dibawah ini.

Tabel 4.17 Pengujian Ubah Password Administrator

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan data yang benar	Password baru berhasil disimpan ke dalam data administrator	Dapat menyimpan data password administrator yang baru	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Memasukkan data *username* yang salah	Muncul pesan kesalahan "Data *username* anda salah"	Dapat menampilkan pesan kesalahan "Data *username* anda salah"	[√] Diterima [ ] Ditolak
Memasukkan data password lama yang salah	Muncul pesan kesalahan "Data password lama anda salah"	Dapat menampilkan pesan kesalahan "Data password lama anda salah"	[√] Diterima [ ] Ditolak
Memasukkan data kosong	Muncul pesan kesalahan "*Field* harus diisi"	Dapat menampilkan pesan kesalahan "*Field* harus diisi"	[√] Diterima [ ] Ditolak

14. Pengujian Tambah Petugas

Berikut ini merupakan pengujian tambah data petugas dapat dilihat pada tabel 4.18 dibawah ini.

Tabel 4.18 Pengujian Tambah Petugas

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data petugas, Contoh : nama lengkap : ichsan aristya *username* : jaky *email* : deathschyte62@gmail.com password : 12345 *level* admin : administrator	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data petugas semua terisi	Data petugas akan tersimpan dalam data petugas	Data petugas akan tersimpan dalam data petugas	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

15. Pengujian tambah kategori

Berikut ini merupakan pengujian tambah data kategori dapat dilihat pada tabel 4.19 dibawah ini.

Tabel 4.19 Pengujian Tambah Kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data kategori, Contoh : nama kategori : ac	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data kategori semua terisi	Data kategori akan tersimpan dalam data kategori	Data kategori akan tersimpan dalam data kategori	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

16. Pengujian ubah kategori

Berikut ini merupakan pengujian ubah data kategori dapat dilihat pada tabel 4.20 dibawah ini.

Tabel 4.20 Pengujian Ubah Kategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data kategori, Contoh : nama kategori : ac	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

	diisi		
Field di form ubah kategori semua terisi	Data kategori akan tersimpan dalam data kategori	Data kategori akan tersimpan dalam data kategori	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

17. Pengujian tambah subkategori

Berikut ini merupakan pengujian tambah data subkategori dapat dilihat pada tabel 4.21 dibawah ini.

Tabel 4.21 Pengujian Tambah Subkategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data subkategori, Contoh : nama kategori : compressor ac nama subkategori : compressor ac	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data subkategori semua terisi	Data subkategori akan tersimpan dalam data subkategori	Data subkategori akan tersimpan dalam data subkategori	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

18. Pengujian ubah subkategori

Berikut ini merupakan pengujian ubah data subkategori dapat dilihat pada tabel 4.22 dibawah ini.

Tabel 4.22 Pengujian Ubah Subkategori

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data subkategori, Contoh : nama kategori : ac nama subkategori : compressor ac	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field di form ubah subkategori semua terisi	Data subkategori akan tersimpan dalam data subkategori	Data subkategori akan tersimpan dalam data subkategori	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

19. Pengujian tambah spesifikasi

Berikut ini merupakan pengujian tambah data spesifikasi dapat dilihat pada tabel 4.23 dibawah ini.

Tabel 4.23 Pengujian Tambah spesifikasi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data	Jika *field* tidak terisi atau salah	Dapat menampilkan	[√] Diterima

ukuran, Contoh : nama spesifikasi : 1pk	maka muncul pesan kesalahan bahwa *field* harus diisi	pesan kesalahan bahwa *field* tersebut harus diisi	[ ] Ditolak
Field data spesifikasi semua terisi	Data spesifikasi akan tersimpan dalam data ukuran	Data spesifikasi akan tersimpan dalam data spec	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

20. Pengujian ubah spesifikasi

Berikut ini merupakan pengujian ubah data spesifikasi dapat dilihat pada tabel 4.24 dibawah ini.

Tabel 4.24 Pengujian Ubah spesifikasi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data ukuran, Contoh : nama spesifikasi : 1pk	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field di form ubah spesifikasi semua terisi	Data spesifikasi akan tersimpan dalam data spesifikasi	Data spesifikasi akan tersimpan dalam data spec	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus	[√] Diterima [ ] Ditolak

		Diisi”	

21. Pengujian tambah merk

Berikut ini merupakan pengujian tambah data merk dapat dilihat pada tabel 4.25 dibawah ini.

Tabel 4.25 Pengujian Tambah merk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data warna, Contoh : nama merk : sanyo	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data merk semua terisi	Data merk akan tersimpan dalam data warna	Data merk akan tersimpan dalam data merk	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan “*Field* Harus Diisi”	[√] Diterima [ ] Ditolak

22. Pengujian tambah produk

Berikut ini merupakan pengujian tambah data produk dapat dilihat pada tabel 4.26 dibawah ini.

Tabel 4.26 Pengujian Tambah Produk

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

Mengisi data produk, Contoh : tipe produk : 2RS 110D nama kategori : ac nama subkategori : compressor nama spesifikasi : 1pk nama merk : Sanyo berat : 16 ketersediaan : 10 harga : 970.000 diskon : 15 garansi : 6 diskon expired : 10 Feb 2012 gambar : 110d.jpg deskripsi : compressor ac sanyo 2rs 110d	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data produk semua terisi	Data produk akan tersimpan dalam data produk	Data produk akan tersimpan dalam data produk	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

23. Pengujian ubah produk

Berikut ini merupakan pengujian ubah data produk dapat dilihat pada tabel 4.27 dibawah ini.

Tabel 4.27 Pengujian Ubah Produk

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data produk, Contoh : tipe produk : 2RS 110D nama kategori : ac nama subkategori : compressor nama spesifikasi : 1pk nama merk : Sanyo berat : 16 ketersediaan : 10 harga : 970.000 diskon : 15 garansi : 6 diskon expired : 10 Feb 2012 gambar : 110d.jpg deskripsi : compressor ac	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

sanyo 2rs 110d			
Field di form ubah produk semua terisi	Data produk akan tersimpan dalam data produk	Data produk akan tersimpan dalam data produk	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

24. Pengujian tambah propinsi

Berikut ini merupakan pengujian tambah data propinsi dapat dilihat pada tabel 4.28 dibawah ini.

Tabel 4.28 Pengujian Tambah Propinsi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data propinsi, Contoh : nama propinsi : kalimantan barat	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data warna semua terisi	Data warna akan tersimpan dalam data warna	Data warna akan tersimpan dalam data warna	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

25. Pengujian ubah propinsi

Berikut ini merupakan pengujian ubah data propinsi dapat dilihat pada tabel 4.29 dibawah ini.

Tabel 4.29 Pengujian Ubah Propinsi

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data propinsi, Contoh : nama propinsi : kalimantan barat	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field di form ubah propinsi semua terisi	Data propinsi akan tersimpan dalam data propinsi	Data propinsi akan tersimpan dalam data propinsi	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

Pengujian tambah kab/kota

Berikut ini merupakan pengujian tambah data kab/kota dapat dilihat pada tabel 4.30 dibawah ini.

Tabel 4.30 Pengujian Tambah Kab/kota

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data kab/kota, Contoh : nama propinsi : kalimantan barat nama kab/kota : pontianak	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data kab/kota semua terisi	Data kab/kota akan tersimpan dalam data kab/kota	Data kab/kota akan tersimpan dalam data kab/kota	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

26. Pengujian ubah kab/kota

Berikut ini merupakan pengujian ubah data kab/kota dapat dilihat pada tabel 4.31 dibawah ini.

Tabel 4.31 Pengujian Ubah Kab/kota

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data kab/kota, Contoh : nama propinsi : kalimantan barat nama kab/kota : pontianak	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field di form ubah kab/kota semua terisi	Data kab/kota akan tersimpan dalam data kab/kota	Data kab/kota akan tersimpan dalam data kab/kota	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

27. Pengujian tambah jasa pengiriman

Berikut ini merupakan pengujian tambah data jasa pengiriman dapat dilihat pada tabel 4.32 dibawah ini.

Tabel 4.32 Pengujian Tambah jasa Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data jenis pengiriman, Contoh : nama jasa pengiriman : jne	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data jenis pengiriman semua	Data jenis pengiriman akan	Data jenis pengiriman akan	[√] Diterima

terisi	tersimpan dalam data jasa pengiriman	tersimpan dalam data jasa pengiriman	[ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

28. Pengujian tambah jenis pengiriman

Berikut ini merupakan pengujian tambah data jenis pengiriman dapat dilihat pada tabel 4.33 dibawah ini.

Tabel 4.33 Pengujian Tambah Jenis Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data jenis pengiriman, Contoh : nama jenis pengiriman : jne reguler	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field data jenis pengiriman semua terisi	Data jenis pengiriman akan tersimpan dalam data jenis pengiriman	Data jenis pengiriman akan tersimpan dalam data jenis pengiriman	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

29. Pengujian ubah jenis pengiriman

Berikut ini merupakan pengujian ubah data jenis pengiriman dapat dilihat pada tabel 4.34 dibawah ini.

Tabel 4.34 Pengujian Ubah Jenis Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data jenis pengiriman, Contoh : nama jenis pengiriman : jne reguler	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak
Field di form ubah jenis pengiriman semua terisi	Data jenis pengiriman akan tersimpan dalam data jenis pengiriman	Data jenis pengiriman akan tersimpan dalam data jenis pengiriman	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

30. Pengujian tambah ongkos kirim

Berikut ini merupakan pengujian tambah data ongkos kirim dapat dilihat pada tabel 4.35 dibawah ini.

Tabel 4.35 Pengujian Tambah Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data jenis pengiriman,	Jika *field* tidak terisi atau salah	Dapat menampilkan	[√] Diterima

Contoh : nama jenis pengiriman : jne regular nama kab/kota : sumedang biaya : 5.000	maka muncul pesan kesalahan bahwa *field* harus diisi	pesan kesalahan bahwa *field* tersebut harus diisi	[ ] Ditolak
Field data ongkos kirim semua terisi	Data ongkos kirim akan tersimpan dalam data ongkos kirim	Data ongkos kirim akan tersimpan dalam data ongkos kirim	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

31. Pengujian ubah ongkos kirim

Berikut ini merupakan pengujian ubah data ongkos kirim dapat dilihat pada tabel 4.36 dibawah ini.

Tabel 4.36 Pengujian Ubah Ongkos Kirim

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Mengisi data jenis pengiriman, Contoh : nama jenis pengiriman : jne regular nama kab/kota : sumedang biaya : 5.000	Jika *field* tidak terisi atau salah maka muncul pesan kesalahan bahwa *field* harus diisi	Dapat menampilkan pesan kesalahan bahwa *field* tersebut harus diisi	[√] Diterima [ ] Ditolak

Field di form ubah ongkos kirim semua terisi	Data ongkos kirim akan tersimpan dalam data ongkos kirim	Data ongkos kirim akan tersimpan dalam data ongkos kirim	[√] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data Masukan	Yang Diharapkan	Pengamatan	Kesimpulan
Tidak mengisi salah satu *field*	Tampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan "*Field* Harus Diisi"	[√] Diterima [ ] Ditolak

4.4.1.2 Kesimpulan Hasil Pengujian Alpha

Berdasarkan hasil pengujian alpha (fungsional) dengan kasus uji diatas dapat ditarik kesimpulan bahwa pembangunan aplikasi *e-commerce* berbasis web di Toko AC Jaya Abadi bebas kesalahan proses dan secara fungsional mengeluarkan hasil yang sesuai dengan yang diharapkan.

4.4.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan, dengan menggunakan kuesioner mengenai kepuasan user terhadap aplikasi yang telah dibangun, dengan kandungan point syarat *user friendly* serta apakah sesuai dengan yang diharapkan. Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sampel dari *user*.

4.4.2.1 Skenario Pengujian Beta

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

1. Lingkungan Perusahaan

   Penelitian di lingkungan perusahaan dilakukan agar 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*.

   Adapun beberapa poin yang diteliti di lingkungan perusahaan adalah :

   1. Sistem promosi produk Toko AC Jaya Abadi.
   2. Fungsional aplikasi *e-commerce* di Toko AC Jaya Abadi.
   3. Pengolahan data produk.
   4. Pengolahan data transaksi penjualan
   5. Pengolahan laporan produk, pemesanan dan penjualan.
   6. Tampilan *website* Toko AC Jaya Abadi.

   Berikut ini adalah hasil wawancara yang telah dilakukan kepada 1 orang karyawan di Toko Jaya Abadi.

1. Apakah menurut anda, dengan adanya aplikasi *e-commerce* ini memudahkan perusahaan dalam mempromosikan dan mengenalkan produk-produknya ke pelanggan ?

   Jawab : iya, karena dengan promosi melalui internet dapat dengan mudah mempromosikan produk dan layanan dari toko kami ke cakupan yang lebih luas lagi. Calon konsumen dapat dengan mudah melihat produk dan melakukan pemesanan tanpa harus datang ke toko, dan bagi kami dengan berpromosi di internet dapat lebih banyak lagi konsumen.

2. Apakah anda dapat memahami berbagai fungsional (menu,tombol) yang ada dalam aplikasi *e-commerce* ini ?

   Jawab : iya, kami dapat memahami menu dan tombol-tombol dalam aplikasi ini. Aplikasi ini dirancang dengan sesuai dengan apa yang kami butuhkan dan kami inginkan.

3. Apakah menurut anda, aplikasi *e-commerce* ini dapat membantu perusahaan dalam melakukan pengolahan data produk dan data-data master lainnya ?

   Jawab : cukup membantu, dengan aplikasi ini memudahkan kami untuk mengolah data-data mengenai produk yang ada, mengatur stok barang, dan menata produk-produk sesuai dengan kategorinya.

4. Apakah menurut anda, aplikasi *e-commerce* ini dapat membantu perusahaan dalam melakukan pengolahan data transaksi penjualan secara *online* ?

   Jawab : iya, dengan aplikasi ini kami bisa mengelola transaksi penjualan secara online setiap harinya, memudahkan bagi kami untuk mengetahui

besarnya nilai transaksi dan menentukan produk mana yang paling banyak terjual.

5. Apakah menurut anda, aplikasi *e-commerce* ini dapat membantu anda dalam menghasilkan sebuah laporan, baik itu laporan produk dan laporan penjualan ?

   Jawab : sangat membantu, dengan adanya pengolahan data produk dan transaksi pada toko, kami bias lebih mudah untuk menyusun laporan *inventory* produk dan penjualan produk untuk tiap-tiap bulannya sehingga kami dapat menganalisis mengenai pengaturan stok barang mana yang populer yang sebaiknya mempunyai persediaan yang cukup dan kami dapat mengetahui nilai total penjualan untuk menentukan program penjualan yang selanjutnya dapat meningkatkan nilai penjualan kami.

6. Apakah menurut anda tampilan *backend* aplikasi *e-commerce* ini cukup menarik dan mudah untuk dipahami ?

   Jawab : cukup menarik, kami dapat dengan mudah memahaminya, namu ada yang kurang dalam mempresentasikan tombol logout, kami harapkan adanya perbaikan dalam hal itu.

Sedangkan beberapa poin yang diteliti di lingkungan masyarakat umum adalah :

1. Mudah dalam melakukan pemesanan dan pembelian.
2. Dapat memenuhi kebutuhan informasi kepada pelanggan.
3. Memudahkan pelanggan yang berada di kota Bandung maupun di luar kota Bandung dalam melakukan transaksi secara *online*.
4. Pelanggan mengetahui produk-produk dari Toko AC Jaya Abadi.

5. Proses pemesanan dan proses pembayaran.
6. Tampilan *website* Toko AC Jaya Abadi.

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

Keterangan :

P = 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 30 orang di lingkungan masyarakat umum.

1. Apakah menurut anda, dengan adanya aplikasi *e-commerce* ini memudahkan anda dalam melakukan pemesanan dan pembelian produk-produk Toko AC Jaya Abadi ?

Tabel 4.37 Hasil Pengujian Kuisioner Masyarakat Umum Soal No.1

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
1	1	Sangat Membantu	12	40%
	2	Membantu	14	47%
	3	Cukup Membantu	4	13%
	4	Kurang Membantu	0	0%
	5	Tidak Membantu	0	0%
			30	100%

Berdasarkan hasil prosentase diatas maka dapat disimpulkan bahwa sebanyak 12 orang atau 40% mengatakan bahwa aplikasi yang dibuat sangat membantu, 14 orang atau 47% mengatakan bahwa aplikasi yang dibuat membantu, 4 orang atau 13% mengatakan bahwa aplikasi yang dibuat cupkup membantu. Sehingga secara keseluruhan aplikasi tersebut membantu dalam melakukan pemesanan dan pembelian produk.

2. Bagaimana menurut anda, mengenai kelengkapan informasi yang diberikan melalui e-mail ataupun informasi yang ada di aplikasi *e-commerce* ini ?

Tabel 4.38 Hasil Pengujian Kuisioner Masyarakat Umum Soal No.2

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
2	1	Sangat Lengkap	6	20%
	2	Lengkap	14	47%
	3	Cukup Lengkap	10	33%
	4	Kurang Lengkap	0	0%
	5	Tidak Lengkap	0	0%
			30	100%

Berdasarkan hasil prosentase diatas maka dapat disimpulkan bahwa sebanyak 6 orang atau 20% mengatakan bahwa aplikasi yang dibuat sangat lengkap, 14 orang atau 47% mengatakan bahwa aplikasi yang dibuat lengkap, 10 orang atau 33% mengatakan bahwa aplikasi yang dibuat cukup lengkap. Sehingga secara keseluruhan aplikasi tersebut lengkap dalam memberikan informasi ke pelanggan.

3. Apakah anda setuju, dengan adanya aplikasi *e-commerce* ini dapat membantu para calon pelanggan lainnya yang berada di kota Jakarta maupun yang berada di luar kota Jakarta ?

Tabel 4.39 Hasil Pengujian Kuisioner Masyarakat Umum Soal No.3

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
3	1	Sangat Setuju	6	20%
	2	Setuju	15	50%
	3	Cukup Setuju	9	30%
	4	Kurang Setuju	0	0%
	5	Tidak Setuju	0	0%
			30	100%

Berdasarkan hasil prosentase diatas maka dapat disimpulkan bahwa sebanyak 6 orang atau 20% mengatakan sangat setuju, 15 orang atau 50% mengatakan setuju, 9 orang atau 30% mengatakan cukup setuju. Sehingga secara keseluruhan mengatakan setuju bahwa *website* ini membantu pelanggan yang berada di kota Jakarta atau yang berada di luar kota Jakarta dalam melakukan transaksi secara *online*.

4. Apakah menurut anda, promosi dan pengenalan produk melalui aplikasi *e-commerce* ini membantu anda dalam mengetahui produk yang promosi atau dalam mengenal produk-produk Toko AC Jaya Abadi ?

Tabel 4.40 Hasil Pengujian Kuisioner Masyarakat Umum Soal No.4

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
4	1	Sangat Membantu	8	27%
	2	Membantu	15	50%
	3	Cukup Membantu	7	23%
	4	Kurang Membantu	0	0%
	5	Tidak Membantu	0	0%
			30	100%

Berdasarkan hasil prosentase diatas maka dapat disimpulkan bahwa sebanyak 8 orang atau 27% mengatakan sangat membantu, 15 orang atau 50% mengatakan membantu, 7 orang atau 23% mengatakan cukup membantu. Sehingga secara keseluruhan aplikasi tersebut membantu pelanggan dalam mengetahui produk-produk Toko AC Jaya Abadi secara detail.

5. Bagaimana pendapat anda mengenai proses transaksi yang ada di aplikasi *e-commerce* ini, dari mulai proses pemesanan sampai dengan proses pembayaran ?

Tabel 4.41 Hasil Pengujian Kuisioner Masyarakat Umum Soal No.5

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
5	1	Sangat Mudah	4	13%
	2	Mudah	17	57%
	3	Cukup Mudah	9	30%
	4	Kurang Mudah	0	0%
	5	Tidak Mudah (sulit)	0	0%
			30	100%

Berdasarkan hasil prosentase diatas maka dapat disimpulkan bahwa sebanyak 4 orang atau 13% mengatakan sangat mudah, 17 orang atau 57% mengatakan mudah, 9 orang atau 30% mengatakan cukup mudah. Sehingga secara

keseluruhan aplikasi tersebut mudah dalam melakukan pemesanan sampai dengan proses pembayaran.

6. Apakah menurut anda tampilan aplikasi *e-commerce* ini cukup menarik dan mudah untuk dipahami ?

Tabel 4.42 Hasil Pengujian Kuisioner Masyarakat Umum Soal No.6

Pertanyaan	No.	Keterangan	Responden	Prosentase (%)
6	1	Sangat menarik dan sangat mudah dipahami	6	20%
	2	Menarik dan dapat dipahami	14	47%
	3	Cukup menarik dan cukup mudah untuk dipahami	10	33%
	4	Kurang menarik dan agak sulit dipahami	0	0%
	5	Tidak menarik dan sulit untuk dipahami	0	0%
			30	100%

Berdasarkan hasil prosentase diatas maka dapat disimpulkan bahwa sebanyak 6 orang atau 20% mengatakan bahwa aplikasi yang dibuat sangat menarik dan sangat mudah dipahami, 14 orang atau 47% mengatakan bahwa aplikasi yang dibuat menarik dan dapat dipahami, 10 orang atau 33% mengatakan bahwa aplikasi yang dibuat cukup menarik dan cukup mudah dipahami. Sehingga secara keseluruhan struktur aplikasi tersebut menarik dan mudah untuk digunakan.

4.4.2.2 Kesimpulan Hasil Pengujian Beta

Berdasarkan pengujian beta yang dilakukan di lingkungan perusahaan dan di lingkungan masyarakat umum, maka dapat diambil kesimpulan bahwa :

1. Aplikasi ini dapat memberi kemudahan pada pengolahan data master dan pengolahan data transaksi penjualan produk serta pembuatan laporan produk, laporan pesanan maupun laporan penjualan.
2. Tampilan *website* Toko Jaya Abadi ini menarik dan mudah untuk dipahami baik dari situs *backend* maupun *frontend*.
3. *Website* Toko Jaya Abadi ini sudah cukup lengkap dalam memberikan informasi-informasi ke pelanggan.
4. Toko Jaya Abadi dapat dengan mudah mempromosikan produk-produk terbarunya ke pelanggan melalui *website* ini.
5. *Website* Toko Jaya Abadi ini membantu para pelanggan yang berada di kota Bandung maupun yang berada di luar kota bandung dalam melakukan transaksi secara *online*.