BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Sistem

Implementasi sistem merupakan tahap pembangunan 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

Perangkat keras yang digunakan dalam implementasi *cloud computing software as a service* pada aplikasi *e-commerce* memiliki spesifikasi seperti yang terlihat pada tabel 4.1.

Tabel 4.1 Spesifikasi Perangkat Keras yang digunakan

No	Perangkat Keras	Spesifikasi
1	Prosessor	Kecepatan minimum 2.0 Ghz
2	Monitor	Minimum Monitor 14,1", resolusi 800 x 600
3	VGA	Kecepatan minimum 128 MB
4	Memori	Minimum DDR2 512 MB
5	Keyboard	Standard, Port USB
6	Mouse	Optical Mouse USB
7	Koneksi	Kecepatan minimum 512 kbp/s

4.1.2 Perangkat Lunak yang digunakan

Perangkat lunak yang digunakan dalam implementasi *cloud computing software as a service* pada aplikasi *e-commerce* memiliki spesifikasi seperti terlihat pada tabel 4.2.

Tabel 4.2 Spesifikasi Perangkat Lunak yang digunakan

No.	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Minimum Microsoft Windows XP Professional SP2
2	Perangkat Tambahan	Minimum *Web Browser* yang dapat digunakan dalam mengakses adalah *Mozila Firefox*, dan *Google Chrome*

4.2 Implementasi Database

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

4.2.1 Tabel admin *cloud*

Tabel admin berfungsi untuk menyimpan data admin e-toko yang terdiri dari username dan password. Struktur tabel admin dapat dilihat pada tabel 4.3.

Tabel 4.3 Tabel admin

Nama Tabel	Implementasi *Database*
admin	CREATE TABLE IF NOT EXISTS `admin` (`username` varchar(25) NOT NULL, `password` varchar(60) NOT NULL, PRIMARY KEY (`username`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.2.2 Tabel fitur

Tabel fitur berfungsi untuk menyimpan data fitur yang terdiri dari id_fitur, nama_fitur dan deskripsi_fitur. Struktur tabel fitur dapat dilihat pada tabel 4.4.

Tabel 4.4 Tabel fitur

Nama Tabel	Implementasi *Database*
fitur	CREATE TABLE IF NOT EXISTS `fitur` (`id_fitur` int(11) NOT NULL AUTO_INCREMENT, `nama_fitur` varchar(50) NOT NULL, `deskripsi_fitur` text NOT NULL, PRIMARY KEY (`id_fitur`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=13 ;

4.2.3 Tabel paket

Tabel paket berfungsi untuk menyimpan data paket yang terdiri dari id_paket, nama_paket, storage, masa_aktif, harga dan aktif. Struktur tabel paket dapat dilihat pada tabel 4.5.

Tabel 4.5 Tabel paket

Nama Tabel	Implementasi *Database*
paket	CREATE TABLE IF NOT EXISTS `paket` (`id_paket` int(11) NOT NULL AUTO_INCREMENT, `nama_paket` varchar(20) NOT NULL, `storage` int(10) NOT NULL, `masa_aktif` int(2) NOT NULL, `harga` int(11) NOT NULL, `aktif` enum('Y','T') NOT NULL, PRIMARY KEY (`id_paket`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

4.2.4 Tabel paketfitur

Tabel paketfitur berfungsi untuk menyimpan data paket dan data fitur yang terdiri dari id_paket, id_fitur. Struktur tabel paketfitur dapat dilihat pada tabel 4.6.

Tabel 4.6 Tabel paketfitur

Nama Tabel	Implementasi *Database*
paketfitur	CREATE TABLE IF NOT EXISTS `paketfitur` (`id_paket` int(11) NOT NULL, `id_fitur` int(11) NOT NULL, KEY `FK_paketfitur_paket` (`id_paket`), KEY `FK_paketfitur_fitur` (`id_fitur`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.2.5 Tabel pembayaran

Tabel pembayaran berfungsi untuk menyimpan data pembayaran yang terdiri dari id_pembayaran, nama_bank, atas_nama, transfer_ke, tanggal_bayar, deskripsi_bayar, status_bayar, id_toko. Struktur tabel pembayaran dapat dilihat pada tabel 4.7.

Tabel 4.7 Tabel pembayaran

Nama Tabel	Implementasi *Database*
Pembayaran	CREATE TABLE IF NOT EXISTS `pembayaran` (`id_pembayaran` int(11) NOT NULL AUTO_INCREMENT, `nama_bank` varchar(30) NOT NULL, `norek` int(20) NOT NULL, `atas_nama` varchar(30) NOT NULL, `transfer_ke` varchar(30) NOT NULL, `jumlah_transfer` int(20) NOT NULL, `tanggal_bayar` date NOT NULL, `deskripsi_bayar` text NOT NULL, `status_bayar` enum('menunggu','diterima') NOT NULL DEFAULT 'menunggu', `id_toko` varchar(20) NOT NULL, PRIMARY KEY (`id_pembayaran`), KEY `FK_pembayaran_toko` (`id_toko`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=18 ;

4.2.6 Tabel toko

Tabel toko berfungsi untuk menyimpan data toko yang terdiri dari id_toko, nama_toko, email_toko, password, tanggal_daftar, tanggal_aktif, lama_sewa, tanggal_berakhir, total_bayar, status, id_paket. Struktur tabel toko dapat dilihat pada tabel 4.8.

Tabel 4.8 Tabel toko

Nama Tabel	Implementasi *Database*
toko	CREATE TABLE IF NOT EXISTS `toko` (`id_toko` varchar(20) NOT NULL, `nama_toko` varchar(30) NOT NULL, `email_toko` varchar(50) NOT NULL, `password` varchar(10) NOT NULL, `tanggal_daftar` date NOT NULL, `tanggal_aktif` date NOT NULL, `lama_sewa` int(2) NOT NULL, `tanggal_berakhir` date NOT NULL, `total_bayar` int(20) NOT NULL, `status` enum('menunggu','aktif','baru','berakhir') NOT NULL DEFAULT 'menunggu', `id_paket` int(11) NOT NULL, PRIMARY KEY (`id_toko`), KEY `FK_toko_paket` (`id_paket`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.2.7 Tabel admin *e-commerce*

Tabel admin berfungsi untuk menyimpan data admin yang terdiri dari user_admin, pass_admin, email_admin. Struktur tabel admin dapat dilihat pada tabel 4.9.

Tabel 4.9 Tabel admin

Nama Tabel	Implementasi *Database*
admin	CREATE TABLE IF NOT EXISTS `admin` (`user_admin` varchar(30) NOT NULL, `pass_admin` varchar(60) NOT NULL, `email_admin` varchar(100) NOT NULL, `level` varchar(10) NOT NULL , PRIMARY KEY (`user_admin`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.2.8 Tabel bank

Tabel bank berfungsi untuk menyimpan data bank yang terdiri dari id_bank, nama_bank, norek, atas_nama. Struktur tabel bank dapat dilihat pada tabel 4.10.

Tabel 4.10 Tabel bank

Nama Tabel	Implementasi *Database*
bank	CREATE TABLE IF NOT EXISTS `bank` (`id_bank` int(11) NOT NULL AUTO_INCREMENT, `nama_bank` varchar(40) NOT NULL, `norek` int(20) NOT NULL, `atas_nama` varchar(50) NOT NULL, PRIMARY KEY (`id_bank`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.9 Tabel pesan

Tabel pesan berfungsi untuk menyimpan data pesan yang terdiri dari id_pesan, nama_pengirim, email_pengirim, isi_pesan, status_pesan, id_member. Struktur tabel pesan dapat dilihat pada tabel 4.11.

Tabel 4.11 Tabel pesan

Nama Tabel	Implementasi *Database*
pesan	CREATE TABLE IF NOT EXISTS `pesan` (`id_pesan` int(11) NOT NULL AUTO_INCREMENT, `nama_pengirim` varchar(30) NOT NULL, `email_pengirim` varchar(80) NOT NULL, `isi_pesan` text NOT NULL, `status_pesan` enum('baru','dibalas') NOT NULL DEFAULT 'baru', `id_member` int(11) NOT NULL, PRIMARY KEY (`id_tamu`), KEY `FK_member_pesan` (`id_member`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.10 Tabel jasa_pengiriman

Tabel jasa_pengiriman berfungsi untuk menyimpan data jasa_pengiriman yang terdiri dari id_jasa, nama_jasa. Struktur tabel jasa_pengiriman dapat dilihat pada tabel 4.12.

Tabel 4.12 Tabel jasa_pengiriman

Nama Tabel	Implementasi *Database*
Jasa_pengiriman	CREATE TABLE IF NOT EXISTS `jasa_pengiriman` (`id_jasa` int(11) NOT NULL AUTO_INCREMENT, `nama_jasa` varchar(30) NOT NULL, PRIMARY KEY (`id_jasa`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.11 Tabel kategori

Tabel kategori berfungsi untuk menyimpan data kategori yang terdiri dari id_kategori, nama_kategori. Struktur tabel kategori dapat dilihat pada tabel 4.13.

Tabel 4.13 Tabel kategori

Nama Tabel	Implementasi *Database*
kategori	CREATE TABLE IF NOT EXISTS `kategori` (`id_kategori` int(3) NOT NULL AUTO_INCREMENT, `nama_kategori` varchar(30) NOT NULL, PRIMARY KEY (`id_kategori`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=9 ;

4.2.12 Tabel kota

Tabel kota berfungsi untuk menyimpan data kota yang terdiri dari id_kota, nama_kota, id_provinsi. Struktur tabel kota dapat dilihat pada tabel 4.14.

Tabel 4.14 Tabel kota

Nama Tabel	Implementasi *Database*
kota	CREATE TABLE IF NOT EXISTS `kota` (`id_kota` int(11) NOT NULL AUTO_INCREMENT, `nama_kota` varchar(60) NOT NULL, `id_provinsi` int(11) NOT NULL, PRIMARY KEY (`id_kota`), KEY `FK_kota_provinsi` (`id_provinsi`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=498 ;

4.2.13 Tabel member

Tabel member berfungsi untuk menyimpan data member yang terdiri dari id_member, email_member, pass_member, nama_member, tanggal_lahir, alamat_member, kode_pos, no_telp, id_kota. Struktur tabel member dapat dilihat pada tabel 4.15.

Tabel 4.15 Tabel member

Nama Tabel	Implementasi *Database*
member	CREATE TABLE IF NOT EXISTS `member` (`id_member` int(11) NOT NULL AUTO_INCREMENT, `email_member` varchar(90) NOT NULL, `pass_member` varchar(60) NOT NULL, `nama_member` varchar(60) NOT NULL, `tanggal_lahir` date NOT NULL, `alamat_member` text NOT NULL, `kode_pos` int(6) NOT NULL, `no_telp` varchar(14) NOT NULL, `id_kota` int(11) NOT NULL, PRIMARY KEY (`id_member`), UNIQUE KEY `email_member` (`email_member`), KEY `FK_member_kota` (`id_kota`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.14 Tabel ongkos_kirim

Tabel ongkos_kirim berfungsi untuk menyimpan data ongkos pengiriman yang terdiri dari id_ongkos, harga_kirim, id_kota, id_paket. Struktur ongkos_kirim dapat dilihat pada tabel 4.16.

Tabel 4.16 Tabel ongkos_kirim

Nama Tabel	Implementasi *Database*
Ongkos_kirim	CREATE TABLE IF NOT EXISTS `ongkos_kirim` (`id_ongkos` int(11) NOT NULL AUTO_INCREMENT, `harga_kirim` int(11) NOT NULL, `id_kota` int(11) NOT NULL, `id_paket` int(11) NOT NULL, PRIMARY KEY (`id_ongkos`), KEY `FK_ongkos_kirim_paket` (`id_paket`), KEY `FK_ongkos_kirim_kota` (`id_kota`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.15 Tabel order

Tabel order berfungsi untuk menyimpan data order yang terdiri dari id_order, status_order, tgl_order, jam_order, no_resi, id_member, id_ongkos Struktur tabel order dapat dilihat pada tabel 4.17.

Tabel 4.17 Tabel order

Nama Tabel	Implementasi *Database*
order	CREATE TABLE IF NOT EXISTS `order` (`id_order` int(11) NOT NULL AUTO_INCREMENT, `status_order` enum('menunggu','lunas','baru','batal') NOT NULL DEFAULT 'menunggu', `tgl_order` date NOT NULL, `jam_order` time NOT NULL, `no_resi` varchar(40) NOT NULL, `id_member` int(11) NOT NULL, `id_ongkos` int(11) NOT NULL, PRIMARY KEY (`id_orders`), KEY `FK_orders_member` (`id_member`), KEY `FK_orders_ongkos` (`id_ongkos`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.16 Tabel order_detail

Tabel order_detail berfungsi untuk menyimpan data detail order yang terdiri dari id_order, id_produk, jumlah. Struktur tabel order_detail dapat dilihat pada tabel 4.18.

Tabel 4.18 Tabel order_detail

Nama Tabel	Implementasi *Database*
Order_detail	CREATE TABLE IF NOT EXISTS `order_detail` (`id_order` int(5) NOT NULL, `id_produk` int(5) NOT NULL, `jumlah` int(5) NOT NULL, KEY `FK_order_detail_produk` (`id_produk`), KEY `FK_order_detail_order` (`id_order`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.2.17 Tabel keranjang

Tabel keranjang berfungsi untuk menyimpan data keranjang belanja yang terdiri dari id_keranjang, id_produk, id_session, jumlah, tgl_order_keranjang, jam_order_keranjang, stok_keranjang. Struktur tabel text_reading dapat dilihat pada tabel 4.19.

Tabel 4.19 Tabel keranjang

Nama Tabel	Implementasi *Database*
keranjang	CREATE TABLE IF NOT EXISTS `keranjang` (`id_keranjang` int(11) NOT NULL AUTO_INCREMENT, `id_produk` int(5) NOT NULL, `id_session` varchar(100) NOT NULL, `jumlah` int(5) NOT NULL, `tgl_order_keranjang` date NOT NULL, `jam_order_keranjang` time NOT NULL, `stok_keranjang` int(5) NOT NULL, PRIMARY KEY (`id_orders_keranjang`), KEY `FK_order_keranajang_produk` (`id_produk`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.18 Tabel paket_pengiriman

Tabel paket_pengiriman berfungsi untuk menyimpan data paket pengiriman yang terdiri dari id_paket, nama_paket, lama_kirim, id_jasa. Struktur tabel paket_pengiriman dapat dilihat pada tabel 4.20.

Tabel 4.20 Tabel paket_pengiriman

Nama Tabel	Implementasi *Database*
Paket_pengiriman	CREATE TABLE IF NOT EXISTS `paket_pengiriman` (`id_paket` int(11) NOT NULL AUTO_INCREMENT, `nama_paket` varchar(30) NOT NULL, `lama_kirim` int(11) NOT NULL, `id_jasa` int(11) NOT NULL, PRIMARY KEY (`id_paket`), KEY `FK_paket_pengiriman_jasa` (`id_jasa`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.19 Tabel pembayaran

Tabel pembayaran berfungsi untuk menyimpan data pembayaran yang terdiri dari id_pembayaran, nama_bank, norek, atas_nama, jumlah_transfer, tanggal_transfer, deskripsi_transfer, status_pembayaran, id_bank, id_order. Struktur tabel paket_pengiriman dapat dilihat pada tabel 4.21.

Tabel 4.21 Tabel pembayaran

Nama Tabel	Implementasi *Database*
pembayaran	CREATE TABLE IF NOT EXISTS `pembayaran` (`id_pembayaran` int(11) NOT NULL AUTO_INCREMENT, `nama_bank` varchar(30) NOT NULL, `norek` int(20) NOT NULL, `atas_nama` varchar(50) NOT NULL, `jumlah_transfer` int(20) NOT NULL, `tanggal_transfer` date NOT NULL, `deskripsi_transfer` text NOT NULL, `status_pembayaran` enum('menunggu','diterima') NOT NULL DEFAULT 'menunggu', `id_bank` int(11) NOT NULL, `id_order` int(11) NOT NULL, PRIMARY KEY (`id_pembayaran`), KEY `FK_pembayaran_bank` (`id_bank`), KEY `FK_pembayaran_order` (`id_order`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.20 Tabel produk

Tabel produk berfungsi untuk menyimpan data produk yang terdiri dari id_produk, nama_produk, produk_seo, deskripsi, harga, stok, berat, tgl_masuk, gambar, dibeli, diskon, id_kategori. Struktur tabel produk dapat dilihat pada tabel 4.22.

Tabel 4.22 Tabel produk

Nama Tabel	Implementasi *Database*
produk	CREATE TABLE IF NOT EXISTS `produk` (`id_produk` int(11) NOT NULL AUTO_INCREMENT, `nama_produk` varchar(100) NOT NULL, `produk_seo` varchar(100) NOT NULL, `deskripsi` text NOT NULL, `harga` int(20) NOT NULL, `stok` int(5) NOT NULL, `berat` decimal(5,0) NOT NULL, `tgl_masuk` date NOT NULL, `gambar` varchar(100) NOT NULL, `dibeli` int(5) NOT NULL, `diskon` int(5) NOT NULL,

Nama Tabel	Implementasi *Database*
	`id_kategori` int(5) NOT NULL, PRIMARY KEY (`id_produk`), KEY `FK_produk_kategori` (`id_kategori`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=3 ;

4.2.21 Tabel profil

Tabel profil berfungsi untuk menyimpan data profil toko yang terdiri dari id_profil, nama_toko, deskripsi, keyword, email_toko, no_telp, alamat_toko, logo. Struktur tabel profil dapat dilihat pada tabel 4.23.

Tabel 4.23 Tabel profil

Nama Tabel	Implementasi *Database*
profil	CREATE TABLE IF NOT EXISTS `profil` (`id_toko` int(11) NOT NULL AUTO_INCREMENT, `nama_toko` varchar(70) NOT NULL, `deskripsi` text NOT NULL, `keyword` varchar(100) NOT NULL, `email_toko` varchar(100) NOT NULL, `no_telp` varchar(14) NOT NULL, `alamat_toko` text NOT NULL, `logo` varchar(100) NOT NULL, PRIMARY KEY (`id_toko`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;

4.2.22 Tabel provinsi

Tabel provinsi berfungsi untuk menyimpan data provinsi yang terdiri dari id_provinsi, nama_provinsi. Struktur tabel provinsi dapat dilihat pada tabel 4.24.

Tabel 4.24 Tabel provinsi

Nama Tabel	Implementasi *Database*
provinsi	CREATE TABLE IF NOT EXISTS `provinsi` (`id_provinsi` int(11) NOT NULL AUTO_INCREMENT, `nama_provinsi` varchar(50) NOT NULL, PRIMARY KEY (`id_provinsi`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=38 ;

4.2.23 Tabel retur

Tabel retur berfungsi untuk menyimpan data retur yang terdiri dari id_retur, id_order, tgl_retur, keterangan_retur, gambar_retur, status_retur. Struktur tabel retur dapat dilihat pada tabel 4.25.

Tabel 4.25 Tabel retur

Nama Tabel	Implementasi *Database*
retur	CREATE TABLE IF NOT EXISTS `retur` (`id_retur` int(11) NOT NULL AUTO_INCREMENT, `id_order` int(11) NOT NULL, `tgl_retur` int(11) NOT NULL, `keterangan_retur` text NOT NULL, `gambar_retur` varchar(100) NOT NULL, `status_retur` enum('menunggu','diproses','diterima','ditolak') NOT NULL, PRIMARY KEY (`id_retur`), KEY `FK_retur_order` (`id_order`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.2.24 Tabel retur_detail

Tabel retur_detail berfungsi untuk menyimpan data detail retur yang terdiri dari id_produk, id_retur, jml_retur. Struktur tabel retur dapat dilihat pada tabel 4.26.

Tabel 4.26 Tabel retur_detail

Nama Tabel	Implementasi *Database*
retur_detail	CREATE TABLE IF NOT EXISTS `retur_detail` (`id_produk` int(11) NOT NULL, `id_retur` int(11) NOT NULL, `jml_retur` int(11) NOT NULL, KEY `FK_retur_detail_produk` (`id_produk`), KEY `FK_retur_detail_retur` (`id_retur`)) ENGINE=InnoDB DEFAULT CHARSET=latin1;

4.2.25 Tabel ym

Tabel ym berfungsi untuk menyimpan data ym yang terdiri dari id_ym, nama_ym, user_ym. Struktur tabel ym dapat dilihat pada tabel 4.27.

Tabel 4.27 Tabel ym

Nama Tabel	Implementasi *Database*
ym	CREATE TABLE IF NOT EXISTS `ym` (`id_ym` int(11) NOT NULL AUTO_INCREMENT, `nama_ym` varchar(20) NOT NULL, `user_ym` varchar(40) NOT NULL, PRIMARY KEY (`id_ym`)) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;

4.3 Implementasi Antarmuka

Implementasi antar muka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengguna admin cloud dapat dlihat pada Tabel 4.3, untuk admin *e-commerce* dapat dilihat pada tabel 4.9, untuk member dapat dilihat pada tabel 4.15, untuk tampilan implementasi sistem etoko dan aplikasi *e-commerce* akan dilampirkan pada halaman lampiran.

Tabel 4.28 Implementasi Antarmuka untuk Admin cloud

No.	Menu	Deskripsi	Nama File
1	Halaman Login	Digunakan sebagai halaman Login bagi pengguna yaitu admin.	login.php
2.	Beranda	Digunakan untuk penyajian informasi Admin	cloud.php?module=beranda
3.	Paket	Digunakan untuk penyajian informasi data paket	cloud.php?module=paket
4.	Tambah Paket	Digunakan untuk menambah data paket	cloud.php?module=paket&op=tambahpaket
5.	Ubah paket	Digunakan untuk mengubah data paket	cloud.php?module=paket&op=editpaket&id=
6.	fitur	Digunakan untuk penyajian informasi fitur	cloud.php?module=fitur

No.	Menu	Deskripsi	Nama File
7.	Tambah fitur	Digunakan untuk menambah fitur	cloud.php?module=fitur&op=tambahfitur
8.	Ubah fitur	Digunakan untuk mengubah data fitur	cloud.php?module=fitur&op=editfitur&id=
9.	Lihat fitur	Digunakan untuk melihat detail informasi fitur	cloud.php?module=fitur #$r[id_fitur]
10.	Toko	Digunakan untuk penyajian informasi toko	cloud.php?module=toko
11.	Lihat toko	Digunakan untuk melihat detail informasi toko	cloud.php?module=toko#$r[id_toko]
12.	Ubah toko	Digunakan untuk mengubah data toko	cloud.php?module=toko&op=edittoko&id=
13.	Hapus toko	Digunakan untuk menghapus data toko	cloud.php?module=toko&op=hapustoko&id=
14.	Pembayaran	Digunakan untuk penyajian informasi pembayaran	cloud.php?module=pembayaran
15.	Lihat pembayaran	Digunakan untuk melihat detail informasi pembayaran	cloud.php?module=pembayaran#$r[id_toko]
16.	Ubah pembayaran	Digunakan untuk mengubah data pembayaran	cloud.php?module=pembayaran&op=editpembayaran&id=
17.	Hapus pembayaran	Digunakan untuk menghapus data pembayaran	cloud.php?module=pembayaran&op=hapuspembayaran&id=

Tabel 4.29 Implementasi Antarmuka untuk admin E-commerce

No.	Menu	Deskripsi	Nama File
	Halaman Login	Digunakan sebagai halaman Login bagi pengguna yaitu admin *e-commerce*	index.php

No.	Menu	Deskripsi	Nama File
2.	Beranda	Digunakan untuk penyajian informasi admin *e-commerce*	toko.php?module=beranda
3.	Produk	Digunakan untuk penyajian informasi data produk	toko.php?module=produk
4.	Tambah Produk	Digunakan untuk menambah data produk	toko.php?module=produk&op=tambahproduk
5.	Ubah produk	Digunakan untuk mengubah data produk	toko.php?module=produk&op=editproduk&id=
6.	Lihat produk	Digunakan untuk melihat detail informasi data produk	toko.php?module=produk#r[id_produk]
7.	Kategori	Digunakan untuk penyajian informasi data kategori	toko.php?module=kategori
8.	order	Digunakan untuk penyajian informasi data order	toko.php?module=order
9.	Pembayaran	Digunakan untuk penyajian informasi data pembayaran	toko.php?module=pembayaran
10.	Retur	Digunakan untuk penyajian informasi data retur	toko.php?module=retur

No.	Menu	Deskripsi	Nama File
11.	Member	Digunakan untuk penyajian informasi data member	toko.php?module=member
12.	Pesan	Digunakan untuk penyajian informasi data pesan	toko.php?module=pesan
13.	Jasa pengiriman	Digunakan untuk penyajian informasi data jasa pengiriman	toko.php?module=jasa
14.	Paket pengiriman	Digunakan untuk penyajian informasi data paket pengiriman	toko.php?module=paket
15.	Ongkos pengiriman	Digunakan untuk penyajian informasi data ongkos pengiriman	toko.php?module=ongkos
16.	Laporan	Digunakan untuk penyajian informasi data laporan	toko.php?module=laporan
17.	Provinsi	Digunakan untuk penyajian informasi data provinsi	toko.php?module=provinsi
18.	Kota	Digunakan untuk penyajian informasi data kota	toko.php?module=kota
19.	Bank	Digunakan untuk penyajian informasi data bank	toko.php?module=bank

No.	Menu	Deskripsi	Nama File
20.	Ym	Digunakan untuk penyajian informasi data ym	toko.php?module=ym
21.	Profil	Digunakan untuk penyajian informasi data profil web	toko.php?module=profil
22.	Backup & Restore	Digunakan untuk penyajian informasi data restore	toko.php?module=restore

Tabel 4.30 Implementasi Antarmuka untuk member E-commerce

No.	Menu	Deskripsi	Nama File
1	Halaman Login	Digunakan sebagai halaman Login bagi pengguna yaitu member	login.html
2.	Beranda	Digunakan untuk penyajian informasi member	Index.php/toko
3.	Tentang kami	Digunakan untuk penyajian informasi data tentang toko	tentang-kami.html
4.	Cara pembelian	Digunakan untuk penyajian informasi cara pembelian	cara-pembelian.html
5.	Konfirmasi pembayaran	Digunakan untuk konfirmasi pembayaran	konfirmasi.html
6.	Hubungi kami	Digunakan mengirim pesan	hubungi-kami.html

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 lunak tersebut.

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

4.4.1 Skenario Pengujian *Blackbox*

Skenario pengujian *blackbox* menjelaskan skenario pengujian Sistem seperti yang terlihat pada tabel 4.31

Tabel 4.31 Tabel Skenario Pengujian

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Login pengguna	*Black Box*
kategori	Penambahan Perubahan Penghapusan	*Black Box*
produk	Penambahan Perubahan Penghapusan	*Black Box*
order	Perubahan Penghapusan	*Blackbox*
retur	Penambahan Perubahan Penghapusan	*Blackbox*

Kelas Uji	Butir Uji	Jenis Pengujian
pembayaran	Penambahan Perubahan Penghapusan	*Blackbox*
pesan	Penambahan Perubahan Penghapusan	*Blackbox*
kota	Pengunggahan Pengunduhan	*Blackbox*
provinsi	Penambahan Perubahan Penghapusan	*Blackbox*
profil	Penambahan Perubahan Penghapusan	*Blackbox*
Member	Perubahan penambahan Penghapusan	*Blackbox*
Paket	Penambahan Perubahan Penghapusan	*Blackbox*
Fitur	Penambahan Perubahan	*Blackbox*
	Penghapusan	

Kelas Uji	Butir Uji	Jenis Pengujian
	Penghapusan	
Toko	Perubahan Penghapusan	*Blackbox*
Pembayaran	Perubahan Penghapusan	*Blackbox*
Bantuan	Perubahan penghapusan	*Blackbox*

4.4.2 Kasus dan Hasil Pengujian

Pengujian dilakukan dengan menguji setiap proses untuk kemungkinan kesalahan yang terjadi.

4.4.2.1 Pengujian Login Admin cloud

Login digunakan untuk Admin cloud masuk kedalam sistem. Pengujian Data login benar seperti yang terlihat pada tabel 4.32, dan untuk pengujian data login salah seperti yang terlihat pada tabel 4.33.

Tabel 4.32 Pengujian Login Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
username **Password** Contoh Masukan 1. Data login Admin Username : admin Password : admin	Mengisikan data login yang sudah terdaftar.	Data login benar dan akan masuk ke *interface* admin e-toko	[√]diterima [ ]ditolak

Tabel 4.33 Pengujian Login Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
username **Password** Contoh Masukan 1. Data login Admin username : (kosong) Password : (kosong)	Menampilkan pesan "anda belum mengisi username".	Dapat menampilkan pesan kesalahan "username harus diisi"	[√]diterima [ ]ditolak
username **Password**	Menampilkan pesan "Username	Dapat menampilkan pesan kesalahan	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan 1. Data login Admin Username: (kosong) Password : admin	harus diisi".	"username harus diisi"	
Username **Password** Contoh Masukan 1. Data login Admin Username : admin Password : (kosong)	Menampilkan pesan " Password harus diisi".	Dapat menampilkan pesan kesalahan "Password harus diisi"	[√]diterima [ ]ditolak
Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username **Password** Contoh Masukan 1. Data login Admin Username : admin Password : admi	Menampilkan pesan "Username atau Password salah".	Dapat menampilkan pesan kesalahan "username atau password salah"	[√]diterima [ ]ditolak

4.4.2.2 Pengujian Pendaftaran cloud

Pengujian pendaftaran cloud dilakukan oleh admin cloud dan pengunjung aplikasi provider. Seperti yang terlihat pada tabel 4.34, sedangkan untuk pengujian data karyawan yang salah seperti terlihat pada tabel 4.35.

Tabel 4.34 Pengujian Pendaftaran Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data pendaftaran Contoh Masukan : Nama toko : tokotokoan Email : toko@gmail.com	Mengisikan data pendaftaran dan menyimpan data.	Data pendaftaran dapat tersimpan.	[√]diterima [ ]ditolak

Tabel 4.35 Pengujian pendaftaran salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data pendaftaran Contoh Masukan : Nama toko : (kosong) email : (kosong)	Menampilkan pesan "Field harus diisi".	Dapat menampilkan pesan kesalahan " Field harus diisi "	[√]diterima [ ]ditolak
Data pendaftaran Contoh Masukan : Nama toko : (tokoku) email : (tokoku)	Menampilkan pesan "*please include an @ in the email addres*".	Dapat menampilkan pesan kesalahan " *please include an @ in the email addres*"	[√]diterima [ ]ditolak

4.4.2.3 Pengujian konfirmasi pembayaran cloud

Pengujian konfirmasi pembayaran dilakukan oleh pengunjung aplikasi provider seperti terlihat pada tabel 4.36, sedangkan untuk pengujian data pendidikan yang salah dapat dilihat pada tabel 4.37.

Tabel 4.36 Pengujian konfirmasi pembayaran Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data konfirmasi pembayaran Contoh Masukan : id toko : ID28011445 nama bank : bni no rekening : 10109407 atas nama : eka wibawa transfer ke : bni tanggal bayar :01-30-2014 keterangan : ini pembayaran	Mengisikan data konfirmasi pembayaran	Data konfirmasi pembayaran dapat tersimpam	[√]diterima [ ]ditolak

Tabel 4.37 Pengujian konfirmasi pembayaran yang salah

Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data konfirmasi pembayaran Contoh Masukan : id toko : (kosong) nama bank : (kosong) no rekening : (kosong)	Dapat menampilkan "*please fill out this field*".	Dapat menampilkan pesan "*please fill otu this field* ".	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
atas nama : (kosong) transfer ke : bni tanggal bayar :01-30-2014 keterangan : (kosong)			

4.4.2.4 Pengujian login aplikasi *e-commerce*

Pengujian login aplikasi *e-commerce* dilakukan oleh member *e-commerce* seperti terlihat pada tabel 4.38, sedangkan untuk pengujian login yang salah dapat dilihat pada tabel 4.39.

Tabel 4.38 Pengujian login aplikasi *e-commerce* benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : **email** **Password** Contoh Masukan 1. Data login member Username : admin@gmail.com Password : admin	Menampilkan *interface* member	Dapat Menampilkan *interface* member	[√]diterima [ ]ditolak

Tabel 4.39 Pengujian login aplikasi *e-commerce* yang salah

Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : **email** **Password** Contoh Masukan 1. Data login member Username : (kosong) Password : admin	Menampilkan "*please fill out this field*".	Dapat menampilkan "*please fill out this field*".	[√]diterima [ ]ditolak
Contoh Masukan : **email** **Password** Contoh Masukan 1. Data login member Username : (kosong) Password : (kosong)	Menampilkan "*please fill out this field*".	Dapat menampilkan "*please fill out this field*".	[√]diterima [ ]ditolak

4.4.2.5 Pengujian Pendaftaran member

Pengujian pendaftaran member dilakukan oleh pengunjung *e-commerce* seperti terlihat pada tabel 4.40, sedangkan untuk pendaftaran member yang salah dapat dilihat pada tabel 4.41.

Tabel 4.40 Pengujian Pendaftaran member Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Email : eka@gmail.com Password :10109407 nama lengkap : eka no telepon :08990905041 tanggal lahir : 01-10-1988 alamat : jl.jalan jalan kode pos :40973	Menyimpan data pendaftaran	Dapat menyimpan data pendaftaran	[√]diterima [ ]ditolak

Tabel 4.41 Pengujian Pendaftaran member Salah

Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Email : (kosong) Password : (kosong) nama lengkap : (kosong) no telepon : (kosong) tanggal lahir : (kosong) alamat : (kosong) kode pos : (kosong)	Menampilkan "*please fill out this field*".	Menampilkan "*please fill out this field*".	[√]diterima [ ]ditolak
Contoh Masukan : Email : eka Password :10109407 nama lengkap : eka no telepon :08990905041 tanggal lahir : 01-10-1988 alamat : jl.jalan jalan kode pos :40973	Menampilkan pesan "*please include an @ in the email addres*".	Dapat Menampilkan pesan "*please include an @ in the email addres*".	[√]diterima [ ]ditolak

4.4.2.6 Pengujian order

Pengujian order dilakukan oleh member dan pengunjung *e-commerce* seperti terlihat pada tabel 4.42, sedangkan untuk pengujian order yang salah dapat dilihat pada tabel 4.43.

Tabel 4.42 Pengujian order benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Nama jasa pengiriman: tiki Paket pengiriman : regular Pembayaran :bni	Mengisikan order dan menyimpan data.	Data order tersimpan	[√]diterima [ ]ditolak

Tabel 4.43 Pengujian order salah

Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Nama jasa : (kosong) Paket pengiriman : (kosong) Pembayaran :(ksoong)	Menampilkan pesan "semua *field* harus diisi"	Dapat menampilkan pesan "semua *field* harus diisi"	[√]diterima [ ]ditolak

4.4.2.7 Pengujian Penambahan data produk

Pengujian penambahan data produk dilakukan oleh admin *e-commerce* seperti terlihat pada tabel 4.44, sedangkan untuk pengujian data produk yang salah dapat dilihat pada tabel 4.45.

Tabel 4.44 Pengujian Penambahan data produk benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : nama produk : contoh berat : 1 harga :20000 diskon : 1 stok :10 deskripsi produk :ini produk gambar : wow.jpg	Mengisikan data produk dan menyimpan data.	Data produk dapat tersimpan.	[√]diterima [ ]ditolak

Tabel 4.45 Pengujian Penambahan Data Jenis Soal Salah

Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : nama produk : (kosong) berat : (kosong) harga : (kosong) diskon : (kosong) stok : (kosong) deskripsi produk : (kosong) gambar : (kosong)	Menampilkan "*please fill out this field*".	Dapat Menampilkan "*please fill out this field*".	[√]diterima [ ]ditolak
Contoh Masukan : nama produk : contoh berat : 1 harga :20000 diskon : 1 stok :10 deskripsi produk :ini produk gambar : wow.pdf	Menampilkan "gambar harus berektensi jpeg"	Dapat Menampilkan "gambar harus berektensi jpeg"	[√]diterima [ ]ditolak

4.4.2.8 Pengujian Penambahan kategori

Pengujian penambahan kategori dilakukan oleh admin *e-commerce* seperti terlihat pada tabel 4.46, sedangkan untuk pengujian kategori yang salah dapat dilihat pada tabel 4.47.

Tabel 4.46 Pengujian Penambahan Data kategori Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Nama kategori:baju	Mengisikan data kategori dan menyimpan data.	Data kategori dapat tersimpan.	[√]diterima [ ]ditolak

Tabel 4.47 Pengujian Penambahan Data kategori Salah

Kasus dan hasil uji (Data Salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Nama kategori : (kosong)	Mengisikan data "*field* harus diisi".	Dapat menampilkan pesan "*field* harus diisi".	[√]diterima [ ]ditolak

4.4.2.9 Pengujian penambahan data provinsi

Pengujian penambahan provinsi dilakukan oleh admin *e-commerce* seperti terlihat pada tabel 4.48, sedangkan untuk pengujian kategori yang salah dapat dilihat pada tabel 4.49.

Tabel 4.48 Pengujian penambahan data provinsi Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan Nama provinsi : aceh	Mengisikan data provinsi dan menyimpan data	Data provinsi tersimpan	[√]diterima [ ]ditolak

Tabel 4.49 Pengujian penambahan data provinsi Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan Nama provinsi : (kosong)	Menampilkan "*please fill out this field*".	Dapat menampilkan "*please fill out this field*".	[√]diterima [ ]ditolak

4.4.2.10 Pengujian Penambahan kota

Pengujian penambahan kota dilakukan oleh admin *e-commerce* seperti terlihat pada tabel 4.50, sedangkan untuk pengujian kota yang salah dapat dilihat pada tabel 4.51.

Tabel 4.50 Pengujian Penambahan kota Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Nama kota :bandung Nama provinsi :jawa barat	Mengisikan data kota dan menyimpan data.	Data kota dapat tersimpan.	[√]diterima [ ]ditolak

Tabel 4.51 Pengujian Penambahan kota salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Nama kota : (kosong) Nama provinsi :jawa barat	Menampilkan "*please fill out this field*".	Dapat Menampilkan "*please fill out this field*".	[√]diterima [ ]ditolak

4.4.2.11 Pengujian pengubahan profil

Pengujian pengubahan profil dilakukan oleh admin *e-commerce* seperti terlihat pada tabel 4.52, sedangkan untuk pengujian profil yang salah dapat dilihat pada tabel 4.53.

Tabel 4.52 Pengujian pengubahan profil Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan nama toko : tokoku email :toko@gmail.com alamat : contoh alamat no telepon :5456546 keyword : toko, jual deskripsi : deskripsi toko logo :logo.png	Mengisikan pengubahan data profil	Data profil yang sudah diubah tersimpan	[√]diterima [ ]ditolak

Tabel 4.53 Pengujian pengubahan profil Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan nama toko : tokoku email :toko@gmail.com alamat : contoh alamat no telepon :5456546 keyword : toko, jual deskripsi : deskripsi toko logo :logo.docx	Menampilkan pesan "logo harus berektensi jpeg/png"	Dapat Menampilkan pesan "logo harus berektensi jpeg/png"	[√]diterima [ ]ditolak

4.4.3 Kesimpulan Pengujian *Blackbox*

Berdasarkan hasil pengujian dengan kasus *sample* uji yang telah dilakukan memberikan kesimpulan bahwa pada proses sudah benar. Penyaringan kesalahan proses dalam bentuk arahan tampilan halaman pesan sudah cukup maksimal. Secara fungsional sistem sudah dapat menghasilkan *output* yang diharapkan.

4.5 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara lansung ke lapangan, dengan menggunakan kuisioner mengenai kepuasan pengguna terhadap aplikasi yang telah dibangun, dengan kandungan point syarat mudah digunakan serta apakah sesuai dengan yang diharapkan. Adapun metode penilaian pengujian yang digunakan yaitu metode kuantitatif berdasarkan data sample *user*.

4.5.1 Skenario Pengujian Beta

Pengujian beta yang dilakukan menggunakan kuisioner yang disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 10 responden yang terdiri dari pedagang. Dari hasil kuisioner tersebut akan dilakukan perhitungan agar dapat dapat diambil kesimpulan terhadap penilaian penerapan produk yang disediakan. Kuisioner ini terdiri dari 5 pertanyaan dengan menggunakan skala likert dengan skala 1 sampai 5, berikut ini daftar pertanyaan yang diberikan:

1. Apakah website ini dapat membantu dalam membangun aplikasi *e-commerce*?

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

2. Apakah Fitur-fitur aplikasi *e-commerce* ini sudah memenuhi standar aplikasi *e-commerce* yang ada?

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

3. Apakah website ini memberikan harga yang sesuai dengan paket yang disediakan ?

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

4. Apakah tampilan pada aplikasi *e-commerce* ini menarik?

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

5. Apakah aplikasi *e-commerce* ini mudah digunakan dan dapat dimengerti?

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

Berdasarkan data hasil kuesioner tersebut, dicari presentase masing-masing jawaban dengan menggunakan rumus :

$$P = \frac{S}{skorideal} \times 100$$

Keterangan :

P = Nilai presentase yang dicari

S = Jumlah frekuensi dikalikan dengan skor yang ditetapkan tiap jawaban

Skorideal = Skor tertinggi dikalikan dengan jumlah sampel

1. Apakah website ini dapat membantu dalam membangun aplikasi *e-commerce*?

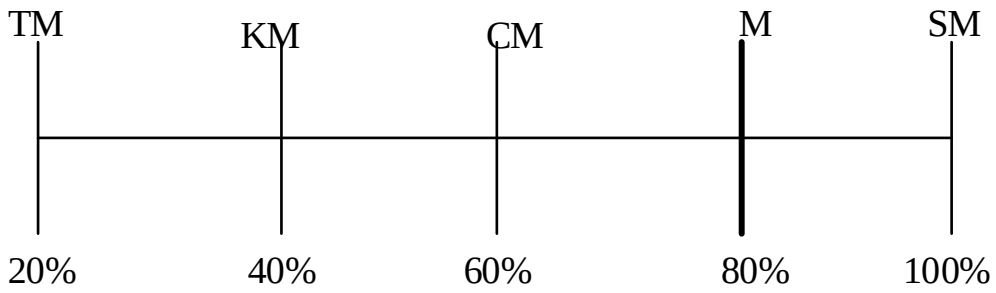
Tabel 4.54 Hasil Pengujian Kuesioner Soal Nomor 1

Pertanyaan	No.	Keterangan	Skor	Responden	S
1	1	Sangat Membantu	5	3	15
	2	Membantu	4	4	16
	3	Cukup Membantu	3	3	9
	4	Kurang Membantu	2	0	0
	5	Tidak Membantu	1	0	0
Jumlah				10	40

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 3 atau 30% menyatakan sangat membantu, 4 atau 40% menyatakan membantu, 3 atau 30% menyatakan cukup membantu.

$$P = \frac{40}{50} \times 100 = 80\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap pertanyaan "apakah website ini dapat membantu dalam membangun aplikasi e-commerce?" dengan presentase adalah 80% dari yang diharapkan 100%, atau bisa dikategorikan membantu. Berikut kriteria interpretasi skor :



Gambar 4.1 Interpretasi skor pertanyaan ke-1

2. Apakah Fitur-fitur aplikasi *e-commerce* ini sudah memenuhi standar aplikasi *e-commerce* yang ada?

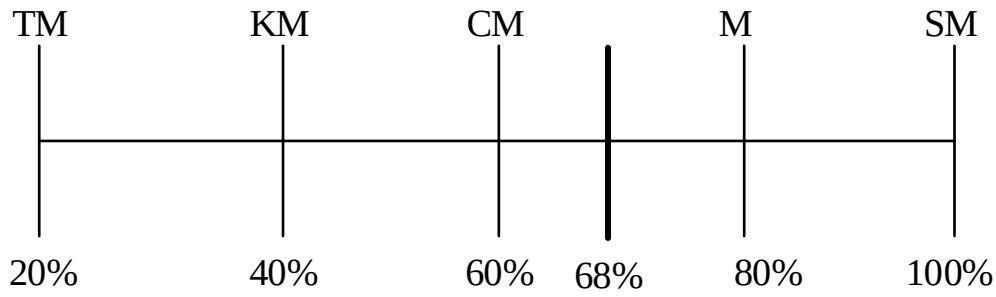
Tabel 4.55 Hasil Pengujian Kuesioner Soal Nomor 2

Pertanyaan	No.	Keterangan	Skor	Responden	S
2	1	Sangat memenuhi	5	0	0
	2	Memenuhi	4	4	16
	3	Cukup memenuhi	3	6	18
	4	Kurang memenuhi	2	0	0
	5	Tidak memenuhi	1	0	0
Jumlah				10	34

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 4 atau 40% menyatakan memenuhi, dan 6 atau 60% cukup memenuhi.

$$P = \frac{34}{50} \times 100 = 68\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap *e-commerce* yang ada dengan presentase adalah 68% dari yang diharapkan 100%, atau bisa dikategorikan sebagai memenuhi. Berikut kriteria interpretasi skor :



Gambar 4.2 Interpretasi skor pertanyaan ke-2

3. Apakah website ini memberikan harga yang sesuai dengan paket yang disediakan ?

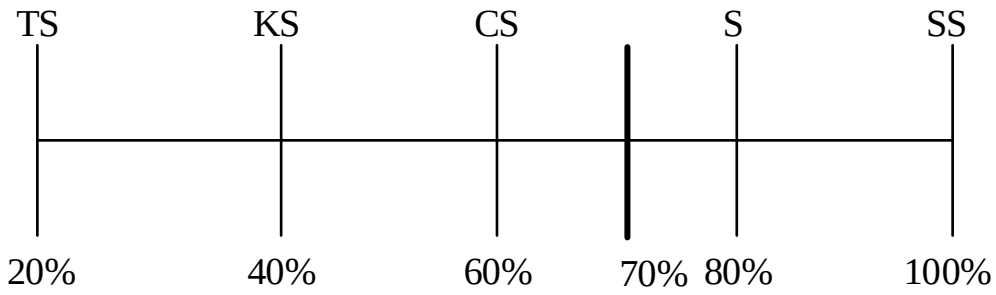
Tabel 4.56 Hasil Pengujian Kuesioner Soal Nomor 3

Pertanyaan	No.	Keterangan	Skor	Responden	S
3	1	Sangat sesuai	5	1	5
	2	Sesuai	4	4	16
	3	Cukup sesuai	3	4	12
	4	Kurang sesuai	2	1	2
	5	Tidak sesuai	1	0	0
Jumlah				10	35

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 1 atau 10% menyatakan sangat sesuai, 4 atau 40% menyatakan sesuai, 4 atau 40% menyatakan cukup sesuai dan 1 atau 10% menyatakan kurang sesuai dalam harga dengan paket yang disediakan.

$$P = \frac{35}{50} \times 100 = 70\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap kesesuaian harga dengan presentase adalah 70% dari yang diharapkan 100%, atau bisa dikategorikan sebagai sesuai. Berikut kriteria interpretasi skor :



Gambar 4.3 Interpretasi skor pertanyaan ke-3

4. Apakah tampilan pada aplikasi *e-commerce* ini menarik?

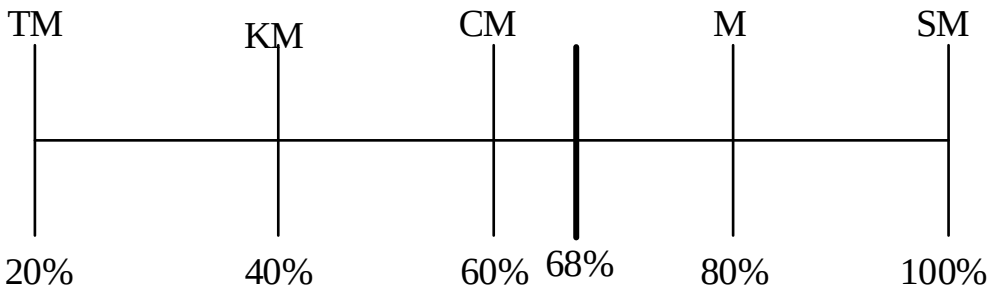
Tabel 4.57 Hasil Pengujian Kuesioner Soal Nomor 4

Pertanyaan	No.	Keterangan	Skor	Responden	S
4	1	Sangat menarik	5	1	5
	2	Menarik	4	3	12
	3	Cukup menarik	3	5	15
	4	Kurang menarik	2	1	2
	5	Tidak menarik	1	0	0
Jumlah				10	34

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 1 atau 10% menyatakan sangat menarik, dan 3 atau 30% menyatakan menarik dan 5 atau 50% menyatakan bahwa tampilan atau *interface* aplikasi *e-commerce* ini cukup menarik dan 1 atau 10% menyatakan kurang menarik.

$$P = \frac{34}{50} \times 100 = 68\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap tampilan *e-commerce* ini dengan jumlah presentase adalah 68% dari yang diharapkan 100%, atau bisa dikategorikan sebagai cukup menarik. Berikut kriteria interpretasi skor :



Gambar 4.4 Interpretasi skor pertanyaan ke-4

5. Apakah aplikasi *e-commerce* ini mudah digunakan dan dapat dimengerti?

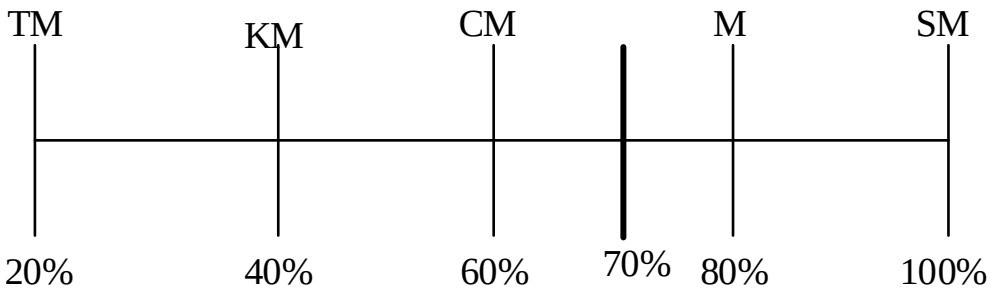
Tabel 4.58 Hasil Pengujian Kuesioner Soal Nomor 5

Pertanyaan	No.	Keterangan	Skor	Responden	S
5	1	Sangat Mengerti	5	0	0
	2	Mengerti	4	6	24
	3	Cukup Mengerti	3	3	9
	4	Kurang Mengerti	2	1	2
	5	Tidak Mengerti	1	0	0
Jumlah				10	35

Berdasarkan hasil persentase diatas maka dapat disimpulkan sebanyak 6 atau 60% menyatakan mengerti, dan 3 atau 30% menyatakan cukup mengerti dan 1 atau 10% menyatakan Tidak Mengerti.

$$P = \frac{35}{50} \times 100 = 70\%$$

Berdasarkan hasil persentase nilai diatas, maka dapat disimpulkan bahwa penilaian terhadap penggunaan aplikasi *e-commerce* ini dengan jumlah presentase adalah 70% dari yang diharapkan 100%, atau bisa dikategorikan sebagai mengerti. Berikut kriteria interpretasi skor :



Gambar 4.5 Interpretasi skor pertanyaan ke-5

4.5.2 Kesimpulan Pengujian Beta

Berdasarkan pengujian beta diatas, dapat diambil kesimpulan bahwa :

1. Fitur-fitur pada aplikasi e-commerce perlu ditambah dan dikembangkan lagi
2. Paket yang disediakan harus lebih variatif