BAB 4

IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini akan dilakukan implementasi dan pengujian terhadap sistem informasi kearsipan yang dibangun. Tahapan ini dilakukan setelah perancangan selesai dilakukan dan selanjutnya akan di implementasikan ke dalam bahasa pemrograman. Setelah implementasi maka dilakukan pengujian terhadap sistem yang baru akan dilihat kekurangan-kekurangan pada aplikasi yang baru untuk pengembangan sistem selanjutnya.

4.1 Implementasi Sistem

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

Untuk mendukung aplikasi yang diterapkan pada toko Gideon Musik, maka dalam hal ini menggunakan perangkat keras dan perangkat lunak yang menunjang dalam pembangunan aplikasi *e-commerce* pada toko Gideon Musik.

4.1.1 Perangkat keras yang digunakan

Spesifikasi perangkat keras yang digunakan dalam pembangunan website e-commerce pada toko Gideon Musik aadalah tercantum dalam tabel 4.1.

Tabel 4.1 Perangkat Keras yang Digunakan

No	Perangkat Keras	Spesifikasi
1	Processor	Core i3 2,4 GHz
2	RAM	1 GB DDR3
3	Hardisk	320 GB
4	Monitor	HD LED LCD 14”
5	VGA	ATI Radeon HD 5470 512 MB
6	LAN Card	10/100 Mbps
7	Koneksi Internet	600 kbps
8	Printer	PIXMA iP1980

4.1.2 Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak yang digunakan dalam pembangunan website e-commerce pada toko Gideon Musik adalah tercantum pada tabel 4.2 sebagai berikut:

Tabel 4.2 Perangkat Lunak yang Digunakan

No	Perangkat Lunak	Keterangan
1	Sistem Operasi	Windows XP SP2
2	Web Browser	Google Chrome
3	Bahasa Pemrograman	PHP
4	DBMS	Mysql 5.0.27

5	Code Editor	Adobe Dreamweaver CS5, Notepad ++
6	DFD Modeler	Microsoft Visio 2007
7	Web Server	XAMPP 2.5.9

4.1.3 Implementasi Database

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

4.1.3.1 Implementasi Tabel Produk

Implementasi dari tabel produk dapat dilihat pada tabel 4.3 :

Tabel 4.3 Tabel Produk

Field	Type	Collation	Attributes	Null	Default	Extra
id_produk	int(11)			No	*None*	AUTO_INCREMENT
nama_produk	varchar(30)	latin1_swedish_ci		No	*None*	
deskripsi	text	latin1_swedish_ci		No	*None*	
gambar	varchar(50)	latin1_swedish_ci		No	*None*	
harga	decimal(10,0)			No	*None*	
diskon	decimal(10,0)			No	*None*	
stok	int(11)			No	*None*	
id_kategori	int(11)			No	*None*	
id_merek	int(5)			No	*None*	
berat	float			No	*None*	
volume	varchar(15)	latin1_swedish_ci		No	*None*	

```
CREATE TABLE IF NOT EXISTS `produk` (
  `id_produk` int(11) NOT NULL AUTO_INCREMENT,
  `nama_produk` varchar(30) NOT NULL,
  `deskripsi` text NOT NULL,
  `gambar` varchar(50) NOT NULL,
  `harga` decimal(10,0) NOT NULL,
  `diskon` decimal(10,0) NOT NULL,
  `stok` int(11) NOT NULL,
  `id_kategori` int(11) NOT NULL,
  `id_merek` int(5) NOT NULL,
  `berat` float NOT NULL,
  `volume` varchar(15) NOT NULL,
  PRIMARY KEY (`id_produk`),
  KEY `id_ kategori ` (`id_ kategori `),
  KEY ` merek ` (`id_ merek `)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=10 ;


ALTER TABLE 'produk'
ADD CONSTRAINT 'produk_fk1' FOREIGN KEY
   ('id_ kategori')REFERENCES'kategori' ('id_ kategori'),
ADD CONSTRAINT 'produk_fk2' FOREIGN KEY
   ('id_ merek')REFERENCES'merek' ('id_ merek');
```

4.1.3.2 Implementasi Tabel Merek

Implementasi dari tabel merek dapat dilihat pada tabel 4.4.

Tabel 4.4 tabel merek

Field	Type	Collation	Attributes	Null	Default	Extra
id_merek	int(5)			No		auto_increment
merek	varchar(15)	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `merek` (
  `id_merek` int(5) NOT NULL auto_increment,
  `merek` varchar(15) NOT NULL,
  PRIMARY KEY  (`id_merek`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=37 ;
```

4.1.3.3 Implementasi Tabel Kategori

Implementasi dari tabel kategori dapat dilihat pada tabel 4.5.

Tabel 4.5 tabel kategori

Field	Type	Collation	Attributes	Null	Default	Extra
id_kategori	int(11)			No		auto_increment
nama_kategori	varchar(20)	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `kategori` (
  `id_kategori` int(11) NOT NULL auto_increment,
  `nama_kategori` varchar(20) NOT NULL,
  PRIMARY KEY  (`id_kategori`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=37 ;
```

4.1.3.4 Implementasi Tabel Provinsi

Implementasi dari tabel kategori dapat dilihat pada tabel 4.6.

Tabel 4.6 tabel provinsi

Field	Type	Collation	Attributes	Null	Default	Extra
id_provinsi	int(3)			No		auto_increment
nama_provinsi	varchar(20)	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `provinsi` (
  `id_provinsi` int(3) NOT NULL auto_increment,
  `nama_provinsi` varchar(20) NOT NULL,
  PRIMARY KEY  (`id_provinsi`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=28 ;
```

4.1.3.5 Implementasi Tabel Kota

Implementasi dari tabel kota dapat dilihat pada tabel 4.7.

Tabel 4.7 tabel kota

Field	Type	Collation	Attributes	Null	Default	Extra
id_kota	int(3)			No		auto_increment
nama_kota	varchar(20)	latin1_swedish_ci		No		
id_provinsi	int(3)			No		

```
CREATE TABLE IF NOT EXISTS `kota` (
  `id_kota` int(3) NOT NULL auto_increment,
  `nama_kota` varchar(20) NOT NULL,
  `id_provinsi` int(3) NOT NULL,
  PRIMARY KEY  (`id_kota`),
  KEY `id_ provinsi ` (`id_ provinsi `)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=69 ;


ALTER TABLE 'kota'
ADD CONSTRAINT 'kota_fk' FOREIGN KEY
   ('id_ provinsi')REFERENCES'provinsi'('id_ provinsi');
```

4.1.3.6 Implementasi Tabel Jasa Pengiriman

Implementasi dari tabel jasa pengiriman dapat dilihat pada tabel 4.8.

Tabel 4.8 tabel jasa pengiriman

Field	Type	Collation	Attributes	Null	Default	Extra
id_jasapengiriman	int(3)			No		auto_increment
nama_jasapengiriman	varchar(35)	latin1_swedish_ci		No		
link_tracking	varchar(50)	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `jasa_pengiriman` (
  `id_jasapengiriman` int(2) NOT NULL AUTO_INCREMENT,
  `nama_jasapengiriman` varchar(20) NOT NULL,
  `link_tracking` varchar(150) NOT NULL,
  PRIMARY KEY (`id_jasapengiriman`),
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

4.1.3.7 Implementasi Tabel Paket Pengiriman

Implementasi dari tabel paket pengiriman dapat dilihat pada tabel 4.9.

Tabel 4.9 Tabel Paket Pengiriman

Field	Type	Collation	Attributes	Null	Default	Extra
id_paket	int(2)			No		auto_increment
id_jasapengiriman	int(2)			No		
nama_paket	varchar(15)	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `paket_pengiriman` (
  `id_paket` int(2) NOT NULL auto_increment,
  `id_jasapengiriman` int(2) NOT NULL,
  `nama_paket` Varchar(15) NOT NULL,
    PRIMARY KEY  (`id_paket`),
   KEY `id_ jasapengiriman’ (`id_ jasapengiriman`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;


ALTER TABLE ‘paket_pengiriman’
ADD CONSTRAINT ‘paket_pengiriman_fk’ FOREIGN KEY
   (‘id_jasapengiriman’)REFERENCES‘jasapengiriman’(‘id_
   jasapengiriman’);
```

4.1.3.8 Implementasi Tabel Ongkos Kirim

Implementasi dari tabel ongkos kirim dapat dilihat pada tabel 4.10.

Tabel 4.10 Tabel ongkos kirim

Field	Type	Collation	Attributes	Null	Default	Extra
id_ongkoskirim	int(11)			No	*None*	AUTO_INCREMENT
id_jasapengiriman	int(2)			No	*None*	
paket_kiriman	int(2)			No	*None*	
id_kota	int(5)			No	*None*	
biaya	decimal(10,0)			No	*None*	

```
CREATE TABLE IF NOT EXISTS `ongkos_kirim` (
  `id_ongkoskirim` int(11) NOT NULL AUTO_INCREMENT,
  `id_jasapengiriman` int(2) NOT NULL,
  `paket_kiriman` int(2) NOT NULL,
  `id_kota` int(5) NOT NULL,
  `biaya` decimal(10,0) NOT NULL,
  PRIMARY KEY (`id_ongkoskirim`),
  UNIQUE KEY ` id_ongkoskirim` (`id_ongkoskirim`),
  KEY `id_jasapengiriman’ (`id_jasapengiriman`),
  KEY `id_kota’ (`id_kota`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=75 ;
ALTER TABLE ‘ongkos_kirim’
ADD CONSTRAINT ‘ongkos_kirim_fk1’ FOREIGN KEY
   (‘id_jasapengiriman’)REFERENCES‘jasapengiriman’(‘id_
   jasapengiriman’),
ADD CONSTRAINT ‘ongkos_kirim_fk2’ FOREIGN KEY
   (‘id_kota’)REFERENCES‘kota’(‘id_ kota’);
```

4.1.3.9 Implementasi Tabel Informasi

Implementasi dari tabel informasi dapat dilihat pada tabel 4.11.

Tabel 4.11 tabel informasi

Field	Type	Collation	Attributes	Null	Default	Extra
id_informasi	int(3)			No		auto_increment
judul	varchar(100)	latin1_swedish_ci		No		
deskripsi	text	latin1_swedish_ci		No		
tgl_buat	date			No		
tgl_ubah	date			No		
username	varchar(35)	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `informasi` (
  `id_informasi` int(3) NOT NULL auto_increment,
  `judul` varchar(100) NOT NULL,
  `deskripsi` text NOT NULL,
  `tgl_buat` date NOT NULL,
  `tgl_ubah` date NOT NULL,
  `username` varchar(35) NOT NULL,
  PRIMARY KEY  (`id_informasi`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=8 ;
```

4.1.3.10 Implementasi Tabel *Member*

Implementasi dari tabel *member* dapat dilihat pada tabel 4.12.

Tabel 4.12 tabel *member*

Field	Type	Collation	Attributes	Null	Default	Extra
id_member	int(3)			No		auto_increment
nama_depan	varchar(50)	latin1_swedish_ci		No		
nama_belakang	varchar(50)	latin1_swedish_ci		No		
alamat	text	latin1_swedish_ci		No		
id_kota	int(3)			No		
kode_pos	varchar(5)	latin1_swedish_ci		No		
email	varchar(35)	latin1_swedish_ci		No		
password	varchar(35)	latin1_swedish_ci		No		
telephone	varchar(15)	latin1_swedish_ci		No		
kode_pendaftaran	varchar(32)	latin1_swedish_ci		No		
kode_resetpassword	varchar(32)	latin1_swedish_ci		No		
status	tinyint(2)			No		
tgl_daftar	date			No		
tgl_ubah	date			No		

```
CREATE TABLE IF NOT EXISTS `member` (
  `id_member` int(3) NOT NULL auto_increment,
  `nama_depan` varchar(50) NOT NULL,
  `nama_belakang` varchar(50) NOT NULL,
  `alamat` text NOT NULL,
  `id_kota` int(3) NOT NULL,
  `kode_pos` varchar(5) NOT NULL,
  `email` varchar(35) NOT NULL,
  `password` varchar(35) NOT NULL,
  `telephone` varchar(15) NOT NULL,
```

```
  `kode_pendaftaran` varchar(32) NOT NULL,
  `kode_resetpassword` varchar(32) NOT NULL,
  `status` tinyint(2) NOT NULL,
  `tgl_daftar` date NOT NULL,
  `tgl_ubah` date NOT NULL,
  PRIMARY KEY  (`id_member`),
  KEY `id_kota' (`id_kota`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=67 ;
ALTER TABLE 'member'
ADD CONSTRAINT 'member_fk1' FOREIGN KEY
   ('id_kota')REFERENCES'kota'('id_ kota');
```

4.1.3.11 Implementasi Tabel Admin

Implementasi dari tabel admin dapat dilihat pada tabel 4.13.

Tabel 4.13 Tabel admin

Field	Type	Collation	Attributes	Null	Default	Extra
id_admin	int(2)			No		auto_increment
username	varchar(35)	latin1_swedish_ci		No		
password	varchar(35)	latin1_swedish_ci		No		
email	varchar(35)	latin1_swedish_ci		No		
Level	enum('0', '1')	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `admin` (
  `id_admin` int(2) NOT NULL auto_increment,
  `username` varchar(35) NOT NULL,
  `password` varchar(35) NOT NULL,
  `email` varchar(35) NOT NULL,
```

```
  `Level` enum('0','1') NOT NULL COMMENT
'0:operator,1:adminnistrator',
  PRIMARY KEY  (`id_admin`),
  UNIQUE KEY `username` (`username`),
  UNIQUE KEY `email` (`email`)
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=11 ;
```

4.1.3.12 Implementasi Tabel Pesanan

Implementasi dari tabel pesanan dapat dilihat pada tabel 4.14.

Tabel 4.14 Tabel Pesanan

Field	Type	Collation	Attributes	Null	Default	Extra
id_pesanan	varchar(15)	latin1_swedish_ci		No	*None*	
tgl_pesanan	date			No	*None*	
tgl_batasbayar	date			No	*None*	
tgl_batasretur	date			No	*None*	
id_member	varchar(30)	latin1_swedish_ci		No	*None*	
total_pesanan	decimal(10,0)			No	*None*	
status	varchar(100)	latin1_swedish_ci		No	*None*	
izin_retur	enum('1','0')	latin1_swedish_ci		No	*None*	

```
CREATE TABLE IF NOT EXISTS `pesanan` (
  `id_pesanan` varchar(15) NOT NULL,
  `tgl_pesanan` date NOT NULL,
  `tgl_batasbayar` date NOT NULL,
  `id_member` int(3) NOT NULL,
  `total_pesanan` decimal(10,0) NOT NULL,
  `status` varchar(10) NOT NULL,
  PRIMARY KEY  (`id_pesanan`),
```

```
KEY `id_member’ (`id_member`)

) ENGINE=MyISAM DEFAULT CHARSET=latin1;

ALTER TABLE ‘pesanan’

ADD CONSTRAINT ‘pesanan_fk’ FOREIGN KEY

(‘id_member’)REFERENCES‘member’(‘id_ member’);
```

4.1.3.13 Implementasi Tabel Detail Pesanan

Implementasi dari tabel detail pesanan dapat dilihat pada tabel 4.15.

Tabel 4.15 Tabel Detail Pesanan

Field	Type	Collation	Attributes	Null	Default	Extra
id_detailpesanan	int(3)			No		auto_increment
id_pesanan	varchar(15)	latin1_swedish_ci		No		
id_produk	int(5)			No		
discount	float			No		
quantity	int(3)			No		

```
CREATE TABLE IF NOT EXISTS `detail_pesanan` (

  `id_detailpesanan` int(3) NOT NULL auto_increment,

  `id_pesanan` varchar(15) NOT NULL,

  `id_produk` int(5) NOT NULL,

  `discount` float NOT NULL,

  `quantity` int(3) NOT NULL,

  PRIMARY KEY  (`id_detailpesanan`),

  KEY `id_pesanan’ (`id_pesanan`),

  KEY `id_produk’ (`id_produk`)

) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=236 ;

ALTER TABLE ‘detail_pesanan’
```

```
ADD CONSTRAINT ‘detail_pesanan_fk1’ FOREIGN KEY
(‘id_pesanan’)REFERENCES‘pesanan’(‘id_pesanan’),
ADD CONSTRAINT ‘detail_pesanan_fk2’ FOREIGN KEY
(‘id_produk’)REFERENCES‘produk’(‘id_produk’);
```

4.1.3.14 Implementasi Tabel Pembayaran

Implementasi dari tabel pembayaran dapat dilihat pada tabel 4.16.

Tabel 4.16 Tabel Pembayaran

Field	Type	Collation	Attributes	Null	Default	Extra
id_pembayaran	int(3)			No		auto_increment
tgl_pembayaran	date			No		
tipe_pembayaran	varchar(10)	latin1_swedish_ci		No		
bank_pembayar	varchar(30)	latin1_swedish_ci		No		
keterangan	text	latin1_swedish_ci		No		
nomor_rekeningpembayar	varchar(45)	latin1_swedish_ci		No		
nama_pembayar	varchar(35)	latin1_swedish_ci		No		
rekening_penerima	varchar(35)	latin1_swedish_ci		No		
mata_uang	varchar(5)	latin1_swedish_ci		No		
kurs	double			No		
id_pesanan	varchar(15)	latin1_swedish_ci		No		
total_pembayaran	decimal(10,0)			No		

```
CREATE TABLE IF NOT EXISTS `pembayaran` (
  `id_pembayaran` int(3) NOT NULL auto_increment,
  `tgl_pembayaran` date NOT NULL,
  `tipe_pembayaran` varchar(10) NOT NULL,
  `bank_pembayar` varchar(30) NOT NULL,
  `keterangan` text NOT NULL,
  `nomor_rekeningpembayar` varchar(45) NOT NULL,
  `nama_pembayar` varchar(35) NOT NULL,
  `rekening_penerima` varchar(35) NOT NULL,
```

```
  `mata_uang` varchar(5) NOT NULL,
  `kurs` double NOT NULL,
  `id_pesanan` varchar(15) NOT NULL,
  `total_pembayaran` decimal(10,0) NOT NULL,
  PRIMARY KEY  (`id_pembayaran`),
  KEY `id_pesanan' (`id_pesanan`),
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=204 ;
ALTER TABLE 'pembayaran'
ADD CONSTRAINT 'pembayaran_fk' FOREIGN KEY
('id_pesanan')REFERENCES'pesanan'('id_pesanan');
```

4.1.3.15 Implementasi Tabel Retur

Implementasi dari tabel retur dapat dilihat pada tabel 4.17.

Tabel 4.17 Tabel Retur

Field	Type	Collation	Attributes	Null	Default	Extra
id_retur	int(3)			No		auto_increment
id_produk	int(5)			No		
quantity	int(11)			No		
comment	text	latin1_swedish_ci		No		
tanggapan	text	latin1_swedish_ci		No		
id_pesanan	varchar(15)	latin1_swedish_ci		No		
tgl_retur	date			No		

```
CREATE TABLE IF NOT EXISTS `retur` (
  `id_retur` int(3) NOT NULL AUTO_INCREMENT,
  `id_produk` int(5) NOT NULL,
  `member` varchar(30) NOT NULL,
  `quantity` int(11) NOT NULL,
```

```
  `status_retur` varchar(10) NOT NULL,
  `comment` text NOT NULL,
  `tanggapan` text NOT NULL,
  `id_pesanan` varchar(15) NOT NULL,
  `tgl_retur` date NOT NULL,
  PRIMARY KEY (`id_retur`),
 KEY `id_produk' (`id_produk`),
 KEY `id_pesanan' (`id_pesanan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=22 ;
ALTER TABLE 'retur'
ADD CONSTRAINT 'retur_fk1' FOREIGN KEY
('id_produk')REFERENCES'produk'('id_produk'),
ADD CONSTRAINT 'retur_fk2' FOREIGN KEY
('id_pesanan')REFERENCES'pesanan'('id_pesanan');
```

4.1.3.16 Implementasi Tabel Detail retur

Implementasi dari tabel detail retur dapat dilihat pada tabel 4.18.

Tabel 4.18 Tabel Detail Retur

Field	Type	Collation	Attributes	Null	Default	Extra
id_detailretur	int(3)			No		auto_increment
id_retur	int(3)			No		
id_detailpesanan	int(3)			No		
jumlah	int(2)			No		

```
CREATE TABLE IF NOT EXISTS `detail_retur` (
  `id_detailretur` int(3) NOT NULL auto_increment,
  `id_retur` int(3) NOT NULL,
```

```
  `id_detailpesanan` int(3) NOT NULL,
  `jumlah` int(2) NOT NULL,
  PRIMARY KEY  (`id_detailretur`),
  KEY `id_retur' (`id_retur`),
  KEY `id_detailpesanan' (`id_detailpesanan`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
ALTER TABLE 'detail_retur'
ADD CONSTRAINT 'detail_retur_fk1' FOREIGN KEY
('id_retur')REFERENCES'retur'('id_retur'),
ADD CONSTRAINT 'detail_retur_fk2' FOREIGN KEY
('id_detailpesanan')REFERENCES'detail_pesanan'('id_detailpes
anan');
```

4.1.3.17 Implementasi Tabel Pengiriman

Implementasi dari tabel pengiriman dapat dilihat pada tabel 4.19.

Tabel 4.19 Tabel Pengiriman

Field	Type	Collation	Attributes	Null	Default	Extra
id_pengiriman	int(11)			No	*None*	AUTO_INCREMENT
id_pesanan	varchar(50)	latin1_swedish_ci		No	*None*	
atas_nama	varchar(35)	latin1_swedish_ci		No	*None*	
alamat	text	latin1_swedish_ci		No	*None*	
provinsi	varchar(35)	latin1_swedish_ci		No	*None*	
id_kota	varchar(35)	latin1_swedish_ci		No	*None*	
kode_pos	varchar(7)	latin1_swedish_ci		No	*None*	
telepone	varchar(15)	latin1_swedish_ci		No	*None*	
resi_kirim	varchar(15)	latin1_swedish_ci		No	*None*	
tgl_kirim	varchar(50)	latin1_swedish_ci		No	*None*	
penerima	varchar(100)	latin1_swedish_ci		No	*None*	
tgl_terima	varchar(100)	latin1_swedish_ci		No	*None*	
id_ongkos	int(11)			No	*None*	
ongkos_kirim	decimal(10,0)			No	*None*	

```
CREATE TABLE IF NOT EXISTS `pengiriman` (
  `id_pengiriman` int(11) NOT NULL AUTO_INCREMENT,
  `id_pesanan` varchar(50) NOT NULL,
  `atas_nama` varchar(35) NOT NULL,
  `alamat` text NOT NULL,
  `provinsi` varchar(35) NOT NULL,
  `id_kota` varchar(35) NOT NULL,
  `kode_pos` varchar(7) NOT NULL,
  `telepone` varchar(15) NOT NULL,
  `resi_kirim` varchar(15) NOT NULL,
  `tgl_kirim` varchar(50) NOT NULL,
  `penerima` varchar(100) NOT NULL,
  `tgl_terima` varchar(100) NOT NULL,
  `id_ongkos` int(11) NOT NULL,
  `ongkos_kirim` decimal(10,0) NOT NULL,
  PRIMARY KEY (`id_pengiriman`),
  KEY `id_pesanan' (`id_pesanan`),
  KEY `id_ongkos' (`id_ongkos`),
) ENGINE=MyISAM  DEFAULT CHARSET=latin1 AUTO_INCREMENT=336 ;
ALTER TABLE 'pengiriman'
ADD CONSTRAINT 'pengiriman_fk1' FOREIGN KEY
('id_pesanan')REFERENCES'pesanan'('id_pesanan'),
ADD CONSTRAINT 'pengiriman_fk2' FOREIGN KEY
('id_ongkos')REFERENCES'ongkos_kirim'('id_ongkos');
```

4.1.3.18 Implementasi Tabel matauang

Implementasi dari tabel matauang dapat dilihat pada tabel 4.20.

Tabel 4.20 tabel matauang

Field	Type	Collation	Attributes	Null	Default	Extra
id_matauang	char(3)	latin1_swedish_ci		No		
kurs	float			No		
username	varchar(35)	latin1_swedish_ci		No		

```
CREATE TABLE IF NOT EXISTS `matauang` (
  `id_matauang` char(3) NOT NULL,
  `kurs` float NOT NULL,
  `username` varchar(35) NOT NULL,
  PRIMARY KEY  (`id_matauang`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

4.1.4 Implementasi Antar Muka

Implementasi antar muka dilakukan dengan setiap tampilan program yang dibangun dan pengkodeannya dalam bentuk *file* program. Berikut ini adalah implementasi antarmuka untuk pengunjung dan member pada Tabel 4.21, operator pada Tabel 4.22, dan administrator pada Tabel 4.23. Adapun untuk tampilan implementasi pembangunan aplikasi *e-commerce* di toko Gideon Musik akan dilampirkan pada halaman lampiran.

Tabel 4.21 Implementasi Antar Muka Pengunjung dan Member

No	Menu	Deskripsi	Nama File
1.	Daftar Member	Digunakan oleh pengunjung	Form_pendaftaran.php

		untuk melakukan pendaftaran menjadi member baru	
2.	Login Member	Digunakan oleh member untuk melakukan login member	Member.php
3.	Lupa Password	Digunakan jika member lupa dengan *password*	Lupa_password.php
4.	Alamat Pengiriman	Digunakan untuk mengganti alamat member	Pengiriman.php
5.	Akun Member	Digunakan untuk mengganti kontak member	Akun_member.php
6.	Ganti Password	Digunakan untuk mengganti password member	gantipassword.php
7.	Catatan pesanan	Digunakan untuk melihat history pesanan yang pernah dilakukan	Catatanpesanan.php
8.	Keranjang	Digunakan untuk melihat,	Keranjang.php

	belanja	merubah, menghapus daftar pesanan yang akan dilakukan	
9.	Kategori	Digunakan untuk melihat barang perkategori	Kategori.php
10.	Logout	Digunakan member untuk keluar dari akunya	Member.php
11.	Pencarian barang	Digunakan member atau pengunjung untuk pencarian barang	Produk.php

Tabel 4.22 Implementasi Antar Muka Operator

No	Menu	Deskripsi	Nama File
1.	Login operator	Digunakan sebagai halaman *login* operator	Login.php
2.	Data kategori	Digunakan untuk melihat, menambah, dan merubah data kategori	Kategori.php
3.	Data produk	Digunakan untuk melihat,	Produk.php

		menambah, dan merubah data produk	
4.	Data provinsi	Digunakan untuk melihat, menambah dan merubah data provinsi	Provinsi.php
5.	Data kota	Digunakan untuk melihat, menambah dan merubah data kota	Kota.php
6.	Data ongkos kirim	Digunakan untuk melihat, menambah dan merubah data ongkos kirim	Ongkos_kirim.php
7.	*logout*	Digunakan operator untuk dapat keluar dari halaman operator	Login.php

Tabel 4.23 Implementasi Antar Muka Administrator

No	Menu	Deskripsi	Nama File
1.	*Login* Administrator	Digunakan sebagai halaman *login* administrator	Login.php
2.	Data admin	Digunakan untuk melihat, menambah, dan merubah data	Admin.php

		petugas	
3.	Data *Member*	Digunakan untuk melihat data member	Lihat_Member.php
4.	Informasi web	Digunakan untuk melihat,dan mengubah data informasi web	Informasi.php
5.	*Logout*	Digunakan administrator untuk keluar dari halaman administrator	Login.php

4.2 Pengujian Perangkat Lunak

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

4.2.1 Pengujian Alpha

Pengujian alpha dilakukan menggunakan metode *black box*. Untuk menentukan pengujian alpha ini dibagi menjadi tiga, yaitu skenario pengujian halaman member, skenario pengujian halaman administrator dan skenario pengujian aplikasi operator.

4.2.1.1 Skenario Pengujian Alpha

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi member maka dapat dilihat pada Tabel 4.24 sebagai berikut:

Tabel 4.24 Skenario Pengujian Aplikasi Member

Kelas Uji	Butir Uji	Jenis Pengujian
Lupa *password*	Menampilkan *form* lupa *password*	*Black Box*
Daftar Member	Menampilkan *form* daftar member	*Black Box*
	Isi *form* daftar member	*Black Box*
	Simpan data member baru	*Black Box*
Login	Isi data *login*	*Black Box*
	Verifikasi *e-mail*	*Black Box*
	Verifikasi *password*	*Black Box*
Ganti Profil	Menampilkan *form* ganti profil	*Black Box*
	Ubah data profil	*Black Box*
	Simpan perubahan	*Black Box*
Ganti *password*	Menampilkan form ganti *password*	*Black Box*
	Isi *form* ganti *password*	*Black Box*

	Simpan data baru	*Black Box*
Keranjang belanja	Menampilkan data pembelian	*Black Box*
	Mengubah data pembelian	*Black Box*
	Menghapus data pembelian	*Black Box*
	Menampilkan konfirmasi transaksi	*Black Box*
Isi *form* alamat tujuan	Menampilkan *form* alamat tujuan	*Black Box*
Isi *form* pembayaran	Menampilkan *form p*embayaran	*Black Box*
Pesanan	Menampilkan daftar pemesanan yang telah Dilakukan	*Black Box*
Logout	Menghapus *session* member	*Black Box*

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi operator maka dapat dilihat pada Tabel 4.25 sebagai berikut :

Tabel 4.25 Skenario Pengujian Aplikasi Operator

Kelas Uji	Butir Uji	Jenis Pengujian
Login	Isi data *login*	*Black Box*
	Verifikasi *e-mail*	*Black Box*
	Verifikasi *password*	*Black Box*
Data kategori	Menampilkan data kategori	*Black Box*
	Tambah data kategori	*Black Box*
	Ubah data kategori	*Black Box*
Data produk	Menampilkan data produk	*Black Box*
	Tambah data produk	*Black Box*
	Ubah data produk	*Black Box*
Laporan penjualan	Menampilkan data laporan penjualan	*Black Box*
	Cari data laporan Penjualan	*Black Box*
	Cetak data laporan Penjualan	*Black Box*
Data provinsi	Menampilkan data provinsi	*Black Box*
	Tambah data provinsi	*Black Box*
	Ubah data provinsi	*Black Box*
Data kota	Menampilkan data kota	*Black Box*
	Tambah data kota	*Black Box*

	Ubah data kota	*Black Box*
Data jasa pengiriman	Tambah data jasa pengiriman	*Black Box*
	ubah data jasa pengiriman	*Black Box*
	Hapus data jasa pengiriman	*Black Box*

Untuk penjelasan lebih lanjut tentang skenario pengujian aplikasi administrator maka dapat dilihat pada Tabel 4.26 sebagai berikut :

Tabel 4.26 Skenario Pengujian Aplikasi Administrator

Kelas Uji	**Butir Uji**	**Jenis Pengujian**
Login	Isi data *login*	*Black Box*
	Verifikasi e-mail	*Black Box*
	Verifikasi *password*	*Black Box*
Data admin	Menampilkan data admin	*Black Box*
	Tambah data admin	*Black Box*
	Ubah data admin	*Black Box*
Data member	Menampilkan data member	*Black Box*
	Ubah data member	*Black Box*
Pengaturan web	Menampilkan *form* pengaturan web	*Black Box*
	Ubah *form* pengaturan web	*Black Box*

Ganti *password*	Menampilkan *form* ganti *password*	*Black Box*
	Isi *form* ganti *password*	*Black Box*
	Simpan data baru	*Black Box*
Logout	Menghapus *session* administrator	*Black Box*

4.2.1.2 Kasus dan Hasil Pengujian

Berdasarkan rencana pengujian yang disusun, maka dilakukan pengujian sebagai berikut.

1. Pengujian Lupa Password member

Untuk penjelasan lebih lanjut tentang pengujian data benar lupa password member maka dapat dilihat pada Tabel 4.27 sebagai berikut :

Tabel 4.27 Pengujian Data Benar Lupa Password Member

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi e-mail dengan benar	Tercantum pada text box pada *field* e-mail	Dapat tercantum pada text box pada *field* email	[√] Diterima [ ] Ditolak
Email : Hery_yud@yahoo	Menampilkan data masukkan pada	Dapat menampilkan data masukkan pada	[√] Diterima [ ] Ditolak

.com	*field* email	*field* email	

Untuk penjelasan lebih lanjut tentang pengujian data salah lupa password member maka dapat dilihat pada Tabel 4.28 sebagai berikut :

Tabel 4.28 Pengujian Data Salah Lupa Password Member

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Format email tidak valid, contoh : Email : Hery.yud.yahoo.com	Menampilkan pesan kesalahan "e-mail tidak valid"	Dapat menampilkan pesan kesalahan "email tidak valid"	[√] Diterima [ ] Ditolak
Email belum terdaftar di dalam database	Menampilkan pesan kesalahan "ERROR : E-mail belum terdaftar!"	Dapat menampilkan pesan kesalahan "ERROR : E-mail belum terdaftar!"	[√] Diterima [ ] Ditolak

2. Pengujian Daftar Member

Untuk penjelasan lebih lanjut tentang pengujian data benar daftar member maka dapat dilihat pada Tabel 4.29 sebagai berikut :

Tabel 4.29 Pengujian Data Benar Daftar Member

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data daftar member, contoh : Email baru : Hery_yud@yahoo.com Password baru : 123456 Konfirmasi password : 123456 Nama Lengkap : Ade Heri Wahyudi Alamat : Jl. Kubang Sari No. 31 Provinsi : Jawa Barat	Jika semua *field* terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses registrasi	Form registrasi terisi atau benar dalam pengisian, maka akan dapat melanjutkan proses registrasi	[√] Diterima [ ] Ditolak

Kota : Bandung Kode Pos : 40132			

Untuk penjelasan lebih lanjut tentang pengujian data salah daftar member maka dapat dilihat pada Tabel 4.30 sebagai berikut :

Tabel 4.30 Pengujian Data Salah Daftar *Member*

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi email yang sudah terdaftar	Tampilkan pesan kesalahan "Email sudah terdaftar!	Dapat menampilkan pesan kesalahan "Email sudah terdaftar!	[√] Diterima [ ] Ditolak
Mengisi konfirmasi password yang berbeda dengan konfirmasi password, contoh : Password Baru : 222222 Konfirmasi Pasword : 111111	Tampilkan pesan Kesalahan "Konfirmasi password" harus sama dengan "Password"	Dapat menampilkan pesan kesalahan "Konfirmasi password" harus sama dengan "Password Baru"	[√] Diterima [ ] Ditolak
Memasukkan data	Menampilkan pesan	Dapat menampilkan	[√] Diterima

kosong	Kesalahan "E-mail dibutuhkan", "Password Baru dibutuhkan", Konfirmasi Pasworddibutuhkan", "Nama dibutuhkan", "Alamat dibutuhkan", "Provinsi dibutuhkan", "Kota dibutuhkan", "Kode Pos dibutuhkan", "Perhitungan dibutuhkan"	pesan kesalahan "E-mail dibutuhkan", "Password ibutuhkan", "Konfirmasi Password dibutuhkan", "Nama dibutuhkan", "Alamat dibutuhkan", "Provinsi dibutuhkan", "Kota dibutuhkan", "Kode Pos dibutuhkan", "Perhitungan dibutuhkan"	[ ] Ditolak

3. Pengujian *Login Member*

Untuk penjelasan lebih lanjut tentang pengujian data benar *login member* maka dapat dilihat pada Tabel 4.31 sebagai berikut :

Tabel 4.31 Pengujian Data Benar *Login member*

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang**	**Pengamatan**	**Kesimpulan**

	diharapkan		
Memasukan alamat *email*: hery_yud.@yahoo.com	Mengisikan data alamat *email* pada *field* alamat email	Dapat mengisi *field* alamat *email* pada *field* alamat *email*	[√] Diterima [ ] Ditolak
Memasukan *Password* : 123456	Mengisikan data *password* pada field *password*	Mengisikan data *password* pada field *password*	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah login *member* maka dapat dilihat pada Tabel 4.32 sebagai berikut :

Tabel 4.32 Pengujian Data Salah Login Member

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data *email* atau *password* yang tidak terdaftar	Tampilkan pesan Kesalahan "ERROR E-mail atau password salah"	Dapat menampilkan pesan kesalahan " ERROR E-mail atau password salah"	[√] Diterima [ ] Ditolak

Mengosongkan data *email* atau *password*	Tampilkan pesan kesalahan "Email dibutuhkan", "password dibutuhkan"	Dapat menampilkan pesan kesalahan "Email dibutuhkan", "pasword dibutukan"	[√] Diterima [ ] Ditolak

4. Pengujian Pencarian Produk

Untuk penjelasan lebih lanjut tentang pengujian data benar pencarian produk maka dapat dilihat pada Tabel 4.33 sebagai berikut :

Tabel 4.33 Pengujian Data Benar Pencarian Produk

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi data pada form menu pencarian, contoh : Nama produk : Mapec	Menampilkan barang sesuai dengan pencarian produk yang dimasukkan	Dapat menampilkan produk sesuai dengan pencarian produk yang dimasukkan	[√] Diterima [ ] Ditolak

5. Pengujian Keranjang Belanja

Untuk penjelasan lebih lanjut tentang pengujian data benar keranjang belanja maka dapat dilihat pada Tabel 4.34 sebagai berikut :

Tabel 4.34 Pengujian Data Benar Keranjang Belanja

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukkan data jumlah produk pada *record* produk yang dipilih. Misal : 3	Tercantum pada text box jumlah pada *record* produk yang dipilih	Dapat tercantum pada text box pada *record* produk yang dipilih	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah keranjang belanja maka dapat dilihat pada Tabel 4.35 sebagai berikut :

Tabel 4.35 Pengujian Data Salah Keranjang Belanja

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Jumlah produk yang dimasukkan ke keranjang	Mengambil stok maksimal pada stok produk dan	Dapat mengambil stok maksimal pada stok produk dan	[√] Diterima [ ] Ditolak

belanja melebihi stok produk. Misal : 10 (Stok produk yang tersedia 8 produk)	menampilkan pesan kesalahan ”Maaf, produk dengan kode 5 hanya memiliki sisa stok 8”	dapat menampilkan pesan kesalahan ”Maaf, produk dengan kode 5 hanya memiliki sisa stok 8”	
Mengisi jumlah nol atau minus pada *record* produk yang dipilih	Menampilkan pesan error: “masukkan tidak boleh bernilai nol atau minus”	Dapat Menampilkan pesan error: “masukkan tidak boleh bernilai nol atau minus”	[√] Diterima [ ] Ditolak

6. Pengujian Form Alamat Pengiriman

Untuk penjelasan lebih lanjut tentang pengujian data benar form alamat pengiriman maka dapat dilihat pada Tabel 4.36 sebagai berikut :

Tabel 4.36 Pengujian Data Benar Form Alamat Pengiriman

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Nama : Ade Heri Wahyudi	Menampilkan data masukkan pada	Dapat menampilkan data	[√] Diterima [ ] Ditolak

Alamat : Jl. Kubang sari No 31 Provinsi : Jawa Barat Kota : Bandung Kode pos : 40134	setiap *field* masing-masing	masukkan pada setiap *field* masing-masing	

Untuk penjelasan lebih lanjut tentang pengujian data salah form alamat tujuan pengiriman maka dapat dilihat pada Tabel 4.37 sebagai berikut:

Tabel 4.37 Pengujian Data Salah Form Alamat Tujuan Pengiriman

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Format kode pos tidak sesuai, contoh : Kode Pos : abcde	Menampilkan pesan kesalahan "Kode Pos tidak valid"	Dapat menampilkan pesan kesalahan "Kode Pos tidak valid"	[√] Diterima [ ] Ditolak
Memasukkan data kosong	Menampilkan pesan kesalahan "Nama dibutuhkan", "Alamat	Dapat menampilkan pesan kesalahan "Nama dibutuhkan", "Alamat dibutuhkan",	[√] Diterima [ ] Ditolak

	dibutuhkan”, “Provinsi dibutuhkan”, “Kota dibutuhkan”, “Kode Pos dibutuhkan”.	“Provinsi dibutuhkan”, “Kota dibutuhkan”, “Kode Pos dibutuhkan”.	

7. Pengujian Form Konfirmasi Pembayaran

Untuk penjelasan lebih lanjut tentang pengujian data benar form konfirmasi pembayaran maka dapat dilihat pada Tabel 4.38 sebagai berikut :

Tabel 4.38 Pengujian Data Benar Form Formulir Pembayaran

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi form pembayaran via transfer bank, contoh : Nama bank pengirim : BNI No rekening asal :	Mengisi form pembayaran via transfer bank dengan benar	Dapat mengisi form pembayaran via transfer bank dengan benar	[√] Diterima [ ] Ditolak

0131446036 Nama pemilik rekening asal : Ade Heri Wahyudi Nama bank penerima : BNI a/n M. Saeful			

Untuk penjelasan lebih lanjut tentang pengujian data salah form konfirmasi pembayaran maka dapat dilihat pada Tabel 4.39 sebagai berikut :

Tabel 4.39 Pengujian Data Salah Form Formulir Pembayaran

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Format No.Rekening Pengirim tidak sesuai, contoh : No.Rekening Pengirim: abcdefgh	Menampilkan pesan kesalahan "no rekening pengirim tidak valid"	Dapat menampilkan pesan kesalahan "no rekening pengirim tidak valid"	[√] Diterima [ ] Ditolak
Format Nama	Menampilkan	Dapat menampilkan	[√] Diterima

Pemilik Rekening Pengirim tidak sesuai, contoh : Nama Pemilik Rekening pengirim: 123456	pesan kesalahan “nama pemilik rekening pengirim tidak valid”	pesan kesalahan “nama pemilik rekening pengirim tidak valid”	[ ] Ditolak
Tidak memasukkan no rekening pengirim	Menampilkan pesan kesalahan “no rekening asal dibutuhkan”	Dapat menampilkan pesan kesalahan ”no rekening asal dibutuhkan”	[√] Diterima [ ] Ditolak
Tidak memasukkan nama pemilik rekening pengirim	Menampilkan pesan kesalahan “nama pemilik rekening asal dibutuhkan”	Dapat menampilkan pesan kesalahan ”nama pemilik rekening asal dibutuhkan”	[√] Diterima [ ] Ditolak

8. Pengujian Ganti *Password Member*

Untuk penjelasan lebih lanjut tentang pengujian data benar ganti *Password Member* maka dapat dilihat pada Tabel.4.40 sebagai berikut :

Tabel 4.40 Pengujian Data Benar Ganti Password Member

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Mengisi *field* dengan benar, contoh : Password sekarang : 123456 Password baru : 654321 Konfirmasi password : 654321	Mengisi *field* dengan benar pada data yang akan dirubah	Dapat mengisi *field* dengan benar pada data yang akan dirubah	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah ganti *password member* maka dapat dilihat pada Tabel 4.41 sebagai berikut :

Tabel 4.41 Pengujian Data Salah Ganti *Password Member*

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Data password sekarang tidak sesuai dengan *database*	Menampilkan pesan kesalahan "password tidak *valid*"	Dapat menampilkan pesan kesalahan "password tidak *valid*"	[√] Diterima [ ] Ditolak

Data *password* baru dan data konfirmasi *password* tidak sama	Menampilkan pesan kesalahan "konfirmasi *password* harus sama dengan *password* baru"	Dapat menampilkan pesan kesalahan "konfirmasi *password* harus sama dengan *password* baru"	[√] Diterima [ ] Ditolak
Tidak mengisi password sekarang	Menampilkan pesan kesalahan "*password* dibutuhkan"	Dapat menampilkan pesan kesalahan "*password* dibutuhkan"	[√] Diterima [ ] Ditolak
Tidak mengisi *password* baru	Menampilkan pesan kesalahan "*password* baru dibutuhkan"	Dapat menampilkan pesan kesalahan "*password* baru dibutuhkan"	[√] Diterima [ ] Ditolak
Tidak mengisi konfirmasi *password*	Menampilkan pesan kesalahan "konfirmasi *password* dibutuhkan"	menampilkan pesan kesalahan "konfirmasi *password* dibutuhkan"	[√] Diterima [ ] Ditolak
Tidak ada data yang dimasukan	Menampilkan pesan kesalahan	Dapat menampilkan pesan kesalahan	[√] Diterima [ ] Ditolak

pada semua field	"*password* dibutuhkan", "*password* baru dibutuhkan", "konfirmasi *password* dibutuhkan"	"*password* dibutuhkan", "*password* baru dibutuhkan", "konfirmasi *password* dibutuhkan"	

9. Pengujian Login Administrator dan Operator

Untuk penjelasan lebih lanjut tentang pengujian data benar login administrator dan operator maka dapat dilihat pada Tabel 4.42 sebagai berikut:

Tabel 4.42 Pengujian Data Benar *Login* Administrator dan Operator

Kasus dan Hasil Uji (Data Benar)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukkan data *login*, contoh : Username: administrator *Password* : 123456	Memasukkan data *login* dengan benar	Dapat memasukkan data *login* dengan benar	[√] Diterima [ ] Ditolak

Untuk penjelasan lebih lanjut tentang pengujian data salah login administrator dan operator maka dapat dilihat pada Tabel 4.43 sebagai berikut:

Tabel 4.43 Pengujian Data Salah *Login* Administrator dan Operator

Kasus dan Hasil Uji (Data salah)			
Data masukan	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan data u*sername* dan *password* yang tidak sesuai	Tampilkan pesan kesalahan ERROR username atau password salah"	Dapat menampilkan pesan kesalahan "ERROR username atau password salah"	[√] Diterima [ ] Ditolak
Memasukan data *username* atau *password* yang tidak terdaftar	Tampilkan pesan kesalahan ERROR username atau *password* salah"	Dapat menampilkan pesan kesalahan "*ERROR* username atau *password* salah"	[√] Diterima [ ] Ditolak
Mengosongkan data *username* atau *password*	Tampilkan pesan Kesalahan "username dibutuhkan", "*password* dibutuhkan"	Dapat menampilkan pesan kesalahan "username dibutuhkan", "*password* dibutuhkan"	[√] Diterima [ ] Ditolak

4.2.1.3 Kesimpulan Pengujian Alpha

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

4.2.2 Pengujian Beta

Pengujian beta merupakan pengujian yang dilakukan secara objektif dimana diuji secara langsung ke lapangan yaitu instansi yang bersangkutan mengenai kepuasan user dengan kandungan poin yaitu pemenuhan kebutuhan dari tujuan awal pembangunan *website e-commerce* dan tampilan antarmuka dari *website e-commerce* tersebut. Pengujian beta dilakukan melalui kuesioner.

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

4.2.2.1 Kuesioner Pengguna

Kuesioner disebarkan menggunakan teknik sampling yaitu *Simple Random Sampling* yang disebarkan kepada 10 pengguna. Dari hasil kuesioner tersebut akan dilakukan perhitungan agar dapat diambil kesimpulan terhadap penilaian penerapan sistem yang baru. Kuisioner ini terdiri dari 5 pertanyaan (contoh

kuisioner dapat dilihat pada lampiran) dengan menggunakan skala likert dengan skala 1 sampai 5, dengan ketentuan skala untuk tiap pertanyaan sebagai berikut :

SS = Sangat Setuju diberi skor = 5

ST = Setuju diberi skor = 4

CS = Cukup Setuju diberi skor = 3

TS = Tidak Setuju diberi skor = 2

STS= Sangat Tidak Setuju diberi skor = 1

4.2.2.2 Kuesioner Pengujian Beta Untuk Pengguna

1. Menurut anda, apakah *website e-commerce* toko Gideon Musik ini dapat memenuhi keinginan pelanggan dalam pelayanan transaksi jual-beli secara *online*?

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

2. Apakah anda setuju dengan adanya *website e-commerce* ini pemilik dapat memperluas cakupan pemasarannya ?

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

3. Apakah anda setuju dengan adanya *website e-commerce* ini dapat memudahkan anda untuk melakukan pembelian barang tanpa harus datang langsung ke toko Gideon Musik?

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

4. Apakah anda setuju bahwa *website e-commerce* ini dapat mempromosikan produk-produk terbaru yang ada pada toko Gideon Musik?

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

5. Apakah anda setuju bahwa tampilan *website e-commerce* toko Gideon Musik cukup menarik dan mudah dipahami / *User friendly* ?

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

Berdasarkan data hasil kuisioner tersebut, dapat dicari prosentase masing-masing jawaban dengan menggunakan rumus : $P = \frac{s}{Skorideal} \times 100\%.$

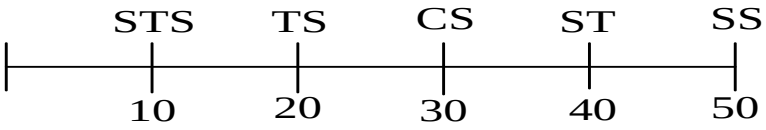
Keterangan :

P = Nilai persentasi yang dicari.

S = Jumlah frekuensi dikalikan dengan skor yang ditetapkan tiap jawaban

Skorideal = Skor tertinggi dikalikan dengan jumlah sampel

Untuk melihat hasil persetujuan dapat digambarkan sebagai berikut:



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

1. Menurut anda, apakah *website e-commerce* toko Gideon Musik ini dapat memenuhi keinginan member dalam pelayanan transaksi jual-beli secara *online*?

Tabel 4.44 Hasil Pengujian Kuisioner Soal Nomor 1

Pertanyaan	No	Keterangan	Responden	Skor
1	1	Sangat Setuju	3	15
	2	Setuju	6	24
	3	Cukup Setuju	1	3
	4	Tidak Setuju	0	0
	5	Sangat Tidak Setuju	0	0
Jumlah			10	42

Berdasarkan hasil persentase pada Tabel 4.44 maka dapat disimpulkan sebanyak 3×5=15 menyatakan sangat setuju (SS), 6×4=24 menyatakan setuju (ST) dan 1×3=3 menyatakan cukup setuju (CS). Jumlah skor yang diperoleh dari penelitian adalah 42. Jadi berdasarkan data maka tingkat persetujuan = (42 : 50) × 100% = 84% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju.

2. Apakah anda setuju dengan adanya *website e-commerce* ini pemilik dapat memperluas cakupan pemasarannya ?

Tabel 4.45 Hasil Pengujian Kuisioner Soal Nomor 2

Pertanyaan	No	Keterangan	Responden	Skor
1	1	Sangat Setuju	3	15
	2	Setuju	6	24
	3	Cukup Setuju	1	3
	4	Kurang Setuju	0	0
	5	Tidak Setuju	0	0
Jumlah			10	42

Berdasarkan hasil persentase pada Tabel 4.45 maka dapat disimpulkan sebanyak 3×5=15 menyatakan sangat setuju (SS), 6×4=24 menyatakan setuju (ST) dan 1×3=3 menyatakan cukup setuju (CS). Jumlah skor yang diperoleh dari penelitian adalah 42. Jadi berdasarkan data maka tingkat persetujuan = (42 : 50) × 100% = 84% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju.

3. Apakah anda setuju dengan adanya *website e-commerce* ini dapat memudahkan anda untuk melakukan pembelian barang tanpa harus datang langsung ke toko Gideon Musik?

Tabel 4.46 Hasil Pengujian Kuisioner Soal Nomor 3

Pertanyaan	No	Keterangan	Responden	Skor
1	1	Sangat Setuju	2	10
	2	Setuju	6	24
	3	Cukup Setuju	2	6
	4	Kurang Setuju	0	0
	5	Tidak Setuju	0	0
Jumlah			10	40

Berdasarkan hasil persentase pada Tabel 4.46 maka dapat disimpulkan sebanyak 2×5=10 menyatakan sangat setuju (SS), 6×4=24 menyatakan setuju (ST) dan 2×3=6 menyatakan cukup setuju (CS). Jumlah skor yang diperoleh dari penelitian adalah 40. Jadi berdasarkan data maka tingkat persetujuan = (40 : 50) × 100% = 80% dari yang diharapkan 100% atau bisa dikategorikan sebagai setuju.

4. Apakah anda setuju bahwa *website e-commerce* ini dapat mempromosikan produk-produk terbaru yang ada pada toko Gideon Musik?

Tabel 4.47 Hasil Pengujian Kuisioner Soal Nomor 4

Pertanyaan	No	Keterangan	Responden	Skor
1	1	Sangat Setuju	4	20
	2	Setuju	6	24
	3	Cukup Setuju	0	0
	4	Kurang Setuju	0	0

	5	Tidak Setuju	0	0
Jumlah			10	44

Berdasarkan hasil persentase pada Tabel 4.47 maka dapat disimpulkan sebanyak 4×5=20 menyatakan sangat setuju (SS) dan 6×4=24 menyatakan setuju (ST). Jumlah skor yang diperoleh dari penelitian adalah 44. Jadi berdasarkan data maka tingkat persetujuan = (44 : 50) × 100% = 88% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju.

5. Apakah anda setuju bahwa tampilan *website e-commerce* toko Gideon Musik cukup menarik dan mudah dipahami / *User friendly* ?

Tabel 4.48 Hasil Pengujian Kuisioner Soal Nomor 5

Pertanyaan	No	Keterangan	Responden	Skor
1	1	Sangat Setuju	3	15
	2	Setuju	7	28
	3	Cukup Setuju	0	0
	4	Kurang Setuju	0	0
	5	Tidak Setuju	0	0
Jumlah			10	43

Berdasarkan hasil persentase pada Tabel 4.48 maka dapat disimpulkan sebanyak 3×5=15 menyatakan sangat setuju (SS) dan 7×4=28 menyatakan setuju (ST). Jumlah skor yang diperoleh dari penelitian adalah 43. Jadi berdasarkan data

maka tingkat persetujuan = (43 : 50) × 100% = 86% dari yang diharapkan 100% atau bisa dikategorikan sebagai sangat setuju.

4.2.2.3 Kesimpulan Pengujian Beta

Berdasarkan hasil wawancara dan kuesioner yang didapatkan dari pengujian beta pengguna administrator, operator dan pengguna, maka sistem penerapan e-commerce untuk sistem penjualan ini sudah sesuai dengan tujuan yaitu untuk memenuhi tuntutan ekonomi digital sehingga dapat meningkatkan pelayanan kepada konsumen, memudahkan pemasaran toko Gideon Musik, memudahkan konsumen untuk melakukan pembelian barang tanpa harus datang langsung ke toko Gideon Musik dan mempermudah toko Gideon Musik dalam mempromosikan produk-produk kepada konsumen.