BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi Sistem

Tahap implementasi 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 Impelementasi Perangkat Keras yang Digunakan

Spesifikasi perangkat keras yang digunakan dalam pembangunan sistem yang dibangun dapat dilihat pada tabel 4.1.

Tabel 4.1 Perangkat Keras yang Digunakan

No.	Perangkat Keras	Spesifikasi
1	Processor	*Speed minimal* 2 GHz
2	RAM	2048 Mb
3	Harddisk	200 GB
4	Monitor	17”
5	VGA	512 Mb

4.1.2 Implementasi Perangkat Lunak yang Digunakan

Spesifikasi perangkat lunak yang digunakan dalam aplikasi yang dibangun dapat dilihat pada tabel 4.2.

Tabel 4.2 Perangkat Lunak yang Digunakan

No.	Perangkat Lunak	Spesifikasi
1	Sistem Operasi	Windows 7 Ultimate
2	Bahasa Pemrograman	PHP
3	*DBMS*	MySQL 5.5.16
4	*Web Browser*	Mozilla Firefox dan Google Chorme
5	*Code Editor*	Adobe Dreamweaver CS 6
6	*Web Server*	XAMPP 2.5

4.2 Implementasi *Database*

Pembuatan *database* dilakukan dengan menggunakan aplikasi *DBMSMySQL*. Implementasi *database* dalam bahasa *SQL* adalah sebagai berikut.

1. Tabel Operator

```
CREATE TABLE IF NOT EXISTS `data_operator` (
  `kode_operator` int(11) NOT NULL AUTO_INCREMENT,
  `nama_operator` varchar(20) NOT NULL,
  `alamat` varchar(50) NOT NULL,
  `kode_bagian` int(15) NOT NULL,
  `user` varchar(20) NOT NULL,
  `no_telp` varchar(17) NOT NULL,
  `password` varchar(45) NOT NULL,
  PRIMARY KEY (`kode_operator`),
  UNIQUE KEY `user` (`user`),
  KEY `fk_databagian_operator_1` (`kode_bagian`)
ALTER TABLE `data_operator`
  ADD CONSTRAINT `data_operator_ibfk_1` FOREIGN KEY
  (`kode_bagian`) REFERENCES `data_bagian`
  (`kode_bagian`) ON DELETE CASCADE ON UPDATE
  CASCADE;
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=6 ;
```

2. Tabel Bahan

```
CREATE TABLE IF NOT EXISTS `data_bahan` (
   `kode_bahan` varchar(7) NOT NULL,
   `nama_bahan` varchar(20) NOT NULL,
   `jenis_bahan` varchar(15) NOT NULL,
   `satuan` varchar(45) NOT NULL,
   `warna` varchar(12) NOT NULL,
   `harga` decimal(10,2) NOT NULL,
   `jumlah` int(11) NOT NULL,
   `kode_supplier` varchar(4) NOT NULL,
   PRIMARY KEY (`kode_bahan`),
   KEY `kode_supplier` (`kode_supplier`)
ALTER TABLE `data_bahan`
   ADD CONSTRAINT `data_bahan_ibfk_1` FOREIGN KEY
    (`kode_supplier`) REFERENCES `data_supplier`
    (`kode_supplier`) ON DELETE CASCADE ON UPDATE
    CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

3. Tabel DEX

```
CREATE TABLE IF NOT EXISTS `data_dex` (
   `kode_dex` varchar(6) NOT NULL,
   `nama_dex` varchar(30) NOT NULL,
   `alamat` varchar(150) NOT NULL,
```

```
  `telp_dex` varchar(18) NOT NULL,
  PRIMARY KEY (`kode_dex`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;REFERENCES
 `obat` (`id_obat`) ON DELETE CASCADE ON UPDATE
 CASCADE)
ENGINE=InnoDB  DEFAULT CHARSET=latin1
 AUTO_INCREMENT=1 ;
```

4. Tabel Pemakaian_Bahan

```
CREATE TABLE IF NOT EXISTS `data_pemakaian_bahan` (
  `kode_pemakaian` int(11) NOT NULL AUTO_INCREMENT,
  `kode_barang` int(11) NOT NULL,
  `jumlah_barang` int(11) NOT NULL,
  `tgl_pemakaian` date NOT NULL,
  PRIMARY KEY (`kode_pemakaian`),
  KEY `fk_databarang_pemakaian` (`kode_barang`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=15 ;
```

5. Tabel Penerimaan

```
CREATE TABLE IF NOT EXISTS `data_penerimaan` (
`kode_penerimaan` int(11) NOT NULL AUTO_INCREMENT,
`kode_supplier` varchar(5) NOT NULL,
`tgl_penerimaan` date NOT NULL,
PRIMARY KEY (`kode_penerimaan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=65 ;
```

6. Tabel Pengiriman

```
CREATE TABLE IF NOT EXISTS `data_pengiriman` (
  `kode_pengiriman` int(11) NOT NULL
AUTO_INCREMENT,
  `kode_DEX` varchar(6) NOT NULL,
  `tgl_pengiriman` date NOT NULL,
  PRIMARY KEY (`kode_pengiriman`),
  KEY `fk_datadex_pengiriman_1` (`kode_DEX`)
ALTER TABLE `data_pengiriman`
  ADD CONSTRAINT `data_pengiriman_ibfk_1` FOREIGN
  KEY (`kode_DEX`) REFERENCES `data_dex`
  (`kode_dex`) ON DELETE CASCADE ON UPDATE CASCADE;
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
AUTO_INCREMENT=16 ;
```

7. Tabel Permintaan

```
CREATE TABLE IF NOT EXISTS `data_permintaan` (
`kode_permintaan` int(11) NOT NULL AUTO_INCREMENT,
`kode_dex` varchar(6) NOT NULL,
`kode_operator` int(11) NOT NULL,
`tgl_permintaan` date NOT NULL,
 PRIMARY KEY (`kode_permintaan`),
 KEY `fk_datadex_permintaan_1` (`kode_dex`),
 KEY `fk_dataopertor_permintaan_1`(`kode_operator`)
ALTER TABLE `data_permintaan`
  ADD CONSTRAINT `data_permintaan_ibfk_1` FOREIGN
  KEY (`kode_dex`) REFERENCES `data_dex`
  (`kode_dex`) ON DELETE CASCADE ON UPDATE CASCADE;
   ) ENGINE=InnoDB  DEFAULT CHARSET=latin1
  AUTO_INCREMENT=10 ;
```

8. Tabel *Supplier*

```
CREATE TABLE IF NOT EXISTS `data_supplier` (
 `kode_supplier` varchar(4) NOT NULL,
 `nama_supplier` varchar(30) NOT NULL,
 `alamat_supplier` varchar(45) NOT NULL,
 `no_telepon` varchar(18) NOT NULL,
 PRIMARY KEY (`kode_supplier`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

9. Tabel Detail_Barang

```
CREATE TABLE IF NOT EXISTS `detail_barang` (
  `kode_barang` int(11) NOT NULL,
  `kode_bahan` varchar(7) NOT NULL,
  `jumlah` int(11) NOT NULL,
  KEY `fk_databarang_detail_barang_1`
(`kode_barang`),
   KEY `fk_databahan_detailbarang_1` (`kode_bahan`)
ALTER TABLE `data_permintaan`
   ADD CONSTRAINT `data_permintaan_ibfk_1` FOREIGN
   KEY (`kode_dex`) REFERENCES `data_dex`
   (`kode_dex`) ON DELETE CASCADE ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

10. Tabel Detail_Pemakaian_Barang

```
CREATE TABLE IF NOT EXISTS `detail_pemakaian_bahan`
(
`kode_pemakaian` int(11) NOT NULL,
`kode_bahan` varchar(7) NOT NULL,
`jumlah` int(11) NOT NULL,
UNIQUE KEY `kode_pemakaian` (`kode_pemakaian`),
KEY `fk_databahan_pemakaian_1` (`kode_bahan`)
ALTER TABLE `detail_pemakaian_bahan`
  ADD CONSTRAINT `detail_pemakaian_bahan_ibfk_1`
FOREIGN KEY (`kode_pemakaian`) REFERENCES
`data_pemakaian_bahan` (`kode_pemakaian`) ON DELETE
CASCADE ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

11. Tabel Detail_pembelian

```
CREATE TABLE IF NOT EXISTS `detail_pembelian` (
 `kode_pembelian` int(11) NOT NULL,
 `kode_bahan` varchar(7) NOT NULL,
 `jumlah` int(11) NOT NULL,
 `harga` decimal(12,2) NOT NULL,
 `keterangan` varchar(45) NOT NULL,
 KEY `kode_pembelian` (`kode_pembelian`),
 KEY `fk_databahan_pembelian_1` (`kode_bahan`)
ALTER TABLE `detail_pembelian`
 ADD CONSTRAINT `detail_pembelian_ibfk_1` FOREIGN KEY
    (`kode_pembelian`) REFERENCES `data_pembelian`
    (`kode_pembelian`) ON DELETE CASCADE ON UPDATE
    CASCADE,
 ADD CONSTRAINT `detail_pembelian_ibfk_2` FOREIGN KEY
    (`kode_bahan`) REFERENCES `data_bahan` (`kode_bahan`) ON
    DELETE CASCADE ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

12. Tabel Detail_Penerimaan

```
CREATE TABLE IF NOT EXISTS `detail_penerimaan` (
 `kode_penerimaan` int(11) NOT NULL,
 `kode_bahan` varchar(7) NOT NULL,
 `jumlah` int(11) NOT NULL,
 `harga` decimal(12,2) NOT NULL,
 KEY `fk_detail_penerimaan_penerimaan_1`
 (`kode_penerimaan`)
ALTER TABLE `detail_penerimaan`
```

```
   ADD CONSTRAINT `detail_penerimaan_ibfk_1`
 FOREIGN KEY (`kode_penerimaan`) REFERENCES
 `data_penerimaan` (`kode_penerimaan`) ON DELETE
 CASCADE ON UPDATE CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13. Tabel Eoq

```
CREATE TABLE IF NOT EXISTS `eoq` (
 `nomor` int(11) NOT NULL AUTO_INCREMENT,
 `kode_bahan` varchar(7) NOT NULL,
 `tahun` year(4) NOT NULL,
 `frek_pesan` int(11) NOT NULL,
 `safety_stok` int(11) NOT NULL,
 `jum_pesan` int(11) NOT NULL,
 `rop` int(11) NOT NULL,
 `jarak_pesan` int(11) NOT NULL,
 `biaya_pesan` decimal(15,2) NOT NULL,
 `biaya_simpan` decimal(15,2) NOT NULL,
 `biaya_total` decimal(15,2) NOT NULL,
 `max_inventory` int(11) NOT NULL,
 PRIMARY KEY (`nomor`),
 KEY `fk_databahan_eoq_2` (`kode_bahan`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1
   AUTO_INCREMENT=35 ;
```

14. Tabel Detail_permintaan

```
CREATE TABLE IF NOT EXISTS `detail_permintaan` (
`kode_permintaan` int(11) NOT NULL,
`kode_barang` int(11) NOT NULL,
`jumlah` int(11) NOT NULL,
 KEY `kode_permintaan` (`kode_permintaan`),
 KEY `fk_databarang_permintaan_1` (`kode_barang`)
ALTER TABLE `detail_permintaan`
  ADD CONSTRAINT `detail_permintaan_ibfk_1` FOREIGN
KEY (`kode_permintaan`) REFERENCES
`data_permintaan` (`kode_permintaan`) ON DELETE
CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `detail_permintaan_ibfk_2` FOREIGN
KEY (`kode_barang`) REFERENCES `data_barang`
(`kode_barang`) ON DELETE CASCADE ON UPDATE
CASCADE;
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

15. Tabel Detail_pengiriman

```
CREATE TABLE IF NOT EXISTS `detail_pengiriman` (
  `kode_pengiriman` int(11) NOT NULL,
  `kode_barang` int(11) NOT NULL,
  `jumlah` int(11) NOT NULL,
  KEY `kode_pengiriman` (`kode_pengiriman`),
  KEY `fk_databarang_pengiriman_1` (`kode_barang`)
ALTER TABLE `detail_pengiriman`
  ADD CONSTRAINT `detail_pengiriman_ibfk_1` FOREIGN KEY
     (`kode_pengiriman`) REFERENCES `data_pengiriman`
     (`kode_pengiriman`) ON DELETE CASCADE ON UPDATE
     CASCADE;
 ) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

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 dapat dilihat pada Tabel 4.3

Tabel 4.3. Implementasi Antarmuka Untuk Admin

No.	Menu	Deskripsi	Nama File
1.	Beranda	Penyajian informasi data bahan baku, permintaan, dan pengiriman	index.php
2.	Tambah Data operator	Digunakan untuk menambah data operator	
3.	Data operator	Penyajian informasi data operator	
4.	Lihat Data	Digunakan untuk melihat data	
5.	Laporan	Digunakan untuk melihat data laporan	

Tabel 4.4 Implementasi Antarmuka Untuk Penyedia Bahan Baku

No.	Menu	Deskripsi	Nama File
1.	Beranda	Penyajian informasi data bahan baku, permintaan, dan pengiriman	index.php
2.	Tambah Data bahan baku	Digunakan untuk menambah data bahan baku	
3.	Data bahan baku	Digunakan untuk menambah data pemakaian obat	

No.	Menu	Deskripsi	Nama File
4.	Tambah Data penerimaan bahan baku	Digunakan untuk menghitung jumlah permintaan obat ke Dinkes	
5.	Data penerimaan bahan bakku	Digunakan untuk menambah data penerimaan obat	
6.	Tambah Data pembelian bahan baku	Penyajian informasi data pemakaian obat	
7.	Data Pembelian bahan bakku	Penyajian informasi data permintaan obat	datapermintaan.php
8.	Tambah Data *supplier*	Penyajian informasi data penerimaan obat	datapenerimaan.php
9.	Data *supplier*	Penyajian informasi cetak laporan pemakaian	lappemakaian.php

Tabel 4.5 Implementasi Antarmuka Untuk Penyedia Barang

No.	Menu	Deskripsi	Nama File
1.	Beranda	Penyajian informasi data bahan baku, permintaan, dan pengiriman	index.php
2.	Tambah Data barang	Digunakan untuk menambah data bahan baku	
3.	Data barang	Digunakan untuk menambah data pemakaian obat	
4.	Tambah Data pengiriman barang	Digunakan untuk menghitung jumlah permintaan obat ke Dinkes	
5.	Data pengiriman barang	Digunakan untuk menambah data penerimaan obat	

Tabel 4.6 Implementasi Antarmuka Untuk *Costumer Service*

No.	Menu	Deskripsi	Nama File
1.	Beranda	Penyajian informasi data bahan baku, permintaan, dan pengiriman	index.php
2.	Tambah Data *DEX*	Digunakan untuk menambah data bahan baku	
3.	Data DEX	Digunakan untuk menambah data pemakaian obat	
4.	Tambah Data permintaan barang	Digunakan untuk menghitung jumlah permintaan obat ke Dinkes	
5.	Data permintaan barang	Digunakan untuk menambah data penerimaan obat	
6.	Lihat data barang		

Tabel 4.7 Implementasi Antarmuka Untuk Produksi

No.	Menu	Deskripsi	Nama File
1.	Beranda	Penyajian informasi data bahan baku, permintaan, dan pengiriman	index.php
2.	Tambah Data detail barang	Digunakan untuk menambah data bahan baku	
3.	Data detail barang	Digunakan untuk menambah data pemakaian obat	
4.	Tambah Data pemakaian bahan	Digunakan untuk menghitung jumlah permintaan obat ke Dinkes	
5.	Data pemakaian bahan	Digunakan untuk menambah data penerimaan obat	

4.4 Pengujian Perangkat Lunak

Pengujian sistem merupakan hal terpenting yang bertujuan untukmenemukan 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 Skenario Pengujian *Black Box*

Pengujian *black box* menjelaskan skenario pengujian sistem dijelaskan pada tabel 4.5.

Tabel 4.5 Tabel Skenario Pengujian

Kasus Uji	Detail Uji	Jenis Pengujian
Login	Login pengguna	*Black Box*
Kelola Data Operator	Penambahan Pengubahan	*Black Box*
Kelola Data Bahan	Penambahan Pengubahan	*Black Box*
Kelola Data Barang	Penambahan Pengubahan	*Black Box*
Kelola Data DEX	Penambahan Pengubahan	*Black Box*
Kelola Data Pemakaian Bahan	Penambahan Pengubahan	*Black Box*
Kelola Data Pembelian	Penambahan	*Black Box*

	Pengubahan	
Kelola Data Penerimaan	Penambahan Pengubahan	*Black Box*
Kelola Data Pengiriman	Penambahan Pengubahan	*Black Box*
Kelola Data Permintaan	Penambahan Pengubahan	*Black Box*
Kelola Data Supplier	Penambahan Pengubahan	*Black Box*
Kelola Data Detail_Barang	Tampilkan	*Black Box*
Kelola Data Detail_Pemakaian_Bahan	Tampilkan	*Black Box*
Kelola Data Detail_Pembelian	Tampilkan	*Black Box*
Kelola Data Detail_Penerimaan	Tampilkan	*Black Box*
Kelola Data Detail_Pengiriman	Tampilkan	*Black Box*
Kelola Data Detail_Permintaan	Tampilkan	*Black Box*
Laporan Pemakaian	Menampilkan form Print	*Black Box*
Laporan Permintaan	Menampilkan form Print	*Black Box*
Laporan Pengiriman	Menampilkan form Print	*Black Box*
Laporan Penerimaan	Menampilkan form Print	*Black Box*

4.4.2 Kasus dan Hasil Pengujian *Black Box*

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

1. Pengujian Login

Login digunakan untuk penentuan hak akses pengguna sebagai admin, Costumer Service, Gudang bahan baku, Gudang Barang, Produksi . Dapat dilihat pengujian data benar login pada tabel 4.6

Tabel 4.6 Pengujian Login Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username **Password** **D4isi** Contoh Masukan 1. Data login Admin Username : arya Password : 123 D4isi : Admin 2. Data login Costumer Service Username : deni	Mengisikan data login yang sudah terdaftar.	Data login benar dan akan masuk ke masing-masing *interface* yang telah ditentukan sesuai dengn data login.	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Password : 123 D4isi : Costumer Service 3. Data login Penyedia Bahan Baku Username : arbi Password : 123 D4isi : Gudang bahan baku 4. Data login Penyedia Barang Username : ari Password : 123 D4isi : Gudang Barang 5.Data login Produksi Username : heri Password : 123 D4isi : Produksi			

Pengujian data salah login maka dapat dilihat pada tabel 4.7.

Tabel 4.7 Pengujian Login Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username **Password** **D4isi** Contoh Masukan 1. Data login Admin Username : arya Password : (kosong) D4isi : (kosong) 2. Data login Costumer Service Username : heri Password : (kosong) D4isi : (kosong) 3. Data login Penyedia Bahan Baku Username : arbi Password : (kosong) D4isi : (kosong)	Menampilkan pesan "Login Gagal ".	Dapat menampilkan pesan kesalahan " Login Gagal ".	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
4. Data login Penyedia Barang Username : ari Password : (kosong) D4isi : (kosong) 5.Data login Produksi Username : heri Password : (kosong) D4isi : (kosong)			
Username **Password** **D4isi** Contoh Masukan 1. Data login Admin Username : arya123 Password : 1234 D4isi : Admin 2. Data login Costumer Service Username : heri Password : 1234567 D4isi : Produksi 3. Data login Penyedia Bahan Baku Username : arbi Password : 123 D4isi : admin 4. Data login Penyedia Barang Username : ari123 Password : 123 D4isi : Gudang Barang 5.Data login Produksi Username : heri12 Password : 123 D4isi : admin	Menampilkan pesan " Login Gagal".	Dapat menampilkan pesan kesalahan " Login Gagal ".	[√]diterima [ ]ditolak

2. Pengujian Penambahan Data Operator

Pengujian penambahan data petugas dilakukan oleh Admin pada tabel 4.8

Tabel 4.8 Pengujian Penambahan Data Operator Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Operator Contoh Masukan : Nama : arya Alamat : Pcm cimahi Bagian : admin User : arya No.Telepon : 08567643876 Password: 123	Mengisikan data operator dan menyimpan data.	Data petugas dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data operator dapat dilihat pada tabel 4.9

Tabel 4.9 Pengujian Penambahan Data Operator Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Operator Contoh Masukan : Nama : (kosong) Alamat : (kosong) Bagian : (kosong) User : (kosong) No.Telepon : (kosong) Password: (kosong)	Menampilkan pesan “data tidak valid! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	Dapat menampilkan pesan kesalahan “Data tidak valid Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	[√]diterima [ ]ditolak
Data Tambah Operator	Menampilkan pesan “Maaf data gagal	Dapat menampilkan pesan kesalahan "	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Contoh Masukan : Nama : arya Alamat : Pcm cimahi Bagian : (kosong) User : (kosong) No.Telepon : 085785865277 Password: 123	ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	

3. Pengujian Penambahan Data Bahan

Pengujian penambahan data Bahan dilakukan oleh bagian penyedia bahan baku bisa dilihat pada tabel 4.10

Tabel 4.10 Pengujian Penambahan Data Bahan Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Bahan Contoh Masukan : Kode Bahan : bbk001 Nama Bahan : Kanvas Nama Supplier : Indahmas Jenis : Kain Satuan : Yard Warna : Putih Harga : 27.500 Jumlah : 1640	Mengisikan data bahan dan menyimpan data.	Data bahan dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data Bahan dapat dilihat pada tabel 4.11.

Tabel 4.11 Pengujian Penambahan Data Bahan Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Bahan Contoh Masukan : Kode Bahan : (kosong) Nama Bahan : (kosong) Nama Supplier : (kosong) Jenis : (kosong) Satuan : (kosong) Warna : (kosong) Harga : (kosong) Jumlah : (kosong)	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak
Data Tambah Bahan Contoh Masukan : Kode Bahan : bbk001 Nama Bahan : Kanvas Nama Supplier : Indahmas Jenis : Kain Satuan : Yard Warna : Putih Harga : 27.500rupiah Jumlah : 1aa	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak

4. Pengujian Penambahan Data Barang

Pengujian penambahan data Barang dilakukan oleh Gudang Barang pada tabel 4.12.

Tabel 4.12 Pengujian Penambahan Data Barang Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Barang Contoh Masukan : Nama Barang: Purple Green Nama Bahan : Kanvas Jumlah Bahan : 5	Mengisikan data barang dan menyimpan data.	Data barang dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data Barang dapat dilihat pada tabel 4.13.

Tabel 4.13 Pengujian Penambahan Data Barang Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Barang Contoh Masukan : Nama Barang: (kosong) Nama Bahan : (kosong) Jumlah Bahan : (kosong)	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak
Data Tambah Barang Contoh Masukan : Nama Barang: Purple Green Nama Bahan : Kanvas Jumlah Bahan : busa	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak

5. Pengujian Penambahan Data *DEX*

Pengujian penambahan data *DEX* dilakukan oleh *Costumer Service* pada tabel 4.14.

Tabel 4.14 Pengujian Penambahan Data *DEX* Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah DEX Contoh Masukan : Nama : Ibu Tanty Alamat : Jambi No. Telepon : 08574873645	Mengisikan data *DEX* dan menyimpan data.	Data *DEX* dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data puskesmas dapat dilihat pada tabel 4.15.

Tabel 4.15 Pengujian Penambahan Data *DEX* Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah *DEX* Contoh Masukan : Nama : (kosong) Alamat : (kosong) No. Telepon : (kosong)	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak
Data Tambah *DEX* Contoh Masukan : Nama : Ibu Tanty Alamat : Jambi No. Telepon : aaadd	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
		data masukan tidak valid.”	

6. Pengujian Penambahan Data Pemakaian Bahan

Pengujian penambahan data pemakaian bahan dilakukan produksi pada tabel 4.16.

Tabel 4.16 Pengujian Penambahan Data Pemakaian Bahan Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Pemakaian Bahan Contoh Masukan : Tanggal Pemakaian : 12-02-2012 Nama Barang : Purple red Nama Bahan : Kanvas Jumlah Bahan : 1	Mengisikan data pemakaian bahan dan menyimpan data.	Data pemakaian bahan dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data pemakaian dapat dilihat pada tabel 4.17.

Tabel 4.17 Pengujian Penambahan Data Pemakaian Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Pemakaian Bahan Contoh Masukan : Tanggal Pemakaian : (kosong) Nama Barang : (kosong) Nama Bahan : (kosong) Jumlah Bahan : (kosong)	Menampilkan pesan “Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	Dapat menampilkan pesan kesalahan “Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Pemakaian Bahan Contoh Masukan : Tanggal Pemakaian : 12-02-2012 Nama Barang : Purple red Nama Bahan : Kanvas Jumlah Bahan : ada	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak

7. Pengujian Penambahan Data Pembelian

Pengujian penambahan data pembelian dilakukan oleh gudang bahan baku pada tabel 4.18.

Tabel 4.18 Pengujian Penambahan Data Pembelian Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Pembelian Contoh Masukan : Tanggal Pembeliann : 12-12-2012 Nama Supplier : Laris Nama Bahan : Kanvas Harga : 27.500 Jumlah : 100	Menghitung jumlah pembelian bahan dan menyimpan data.	Peramalan pembelian bahan dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data pembeliaan dapat dilihat pada tabel 4.19.

Tabel 4.19 Pengujian Penambahan Data Pembelian Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Pembelian Contoh Masukan : Tanggal Pembeliann : (kosong) Nama Supplier : (kosong) Nama Bahan : (kosong) Harga : (kosong) Jumlah : (kosong)	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak
Data Tambah Pembelian Contoh Masukan : Tanggal Pembeliann : 12-12-2012 Nama Supplier : Laris Nama Bahan : Kanvas Harga : 27.500rupiah Jumlah : ada	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak

8. Pengujian Penambahan Data Penerimaan

Pengujian penambahan data penerimaan dilakukan oleh gudang bahan baku pada tabel 4.20.

Tabel 4.20 Pengujian Penambahan Data Penerimaan Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Penerimaan Contoh Masukan :	Mengisikan data penerimaan dan menyimpan data.	Data penerimaan akaian dapat tersimpan.	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tanggal Penerimaan : 10-02-2012 Nama Supplier : Laris Nama Bahan : kanvas Jumlah : 347			

Pengujian data salah penambahan data pengiriman dapat dilihat pada tabel 4.21.

Tabel 4.21 Pengujian Penambahan Data Pengiriman Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Pengiriman Contoh Masukan : Tanggal Penerimaan : (kosong) Nama Supplier : (kosong) Nama Bahan : (kosong) Jumlah : (kosong)	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak
Data Tambah Pengiriman Contoh Masukan : Tanggal Penerimaan : 10-02-2012 Nama Supplier : Laris Nama Bahan : kanvas Harga : 27.500rp Jumlah : 3ada	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak

9. Pengujian Penambahan Data Pengiriman

Pengujian penambahan data pengiriman dilakukan oleh gudang barang pada tabel 4.22.

Tabel 4.22 Pengujian Penambahan Data Pengiriman Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Pengiriman Contoh Masukan : Tanggal Pengiriman : 16-11-2012 Nama *DEX* : Ibu ayi Nama Barang : Kanvas Jumlah : 500	Mengisikan data pengiriman dan menyimpan data.	Data pengiriman dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data penerimaan dapat dilihat pada tabel 4.23.

Tabel 4.23 Pengujian Penambahan Data Penerimaan Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Penerimaan Contoh Masukan : Tanggal Pengiriman : (kosong) Nama *DEX* : (kosong) Nama Barang : (kosong) Jumlah : (kosong)	Menampilkan pesan “Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	Dapat menampilkan pesan kesalahan “Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	[√]diterima [ ]ditolak
Data Tambah Penerimaan Contoh Masukan :	Menampilkan pesan “Maaf data gagal ditambahkan! Data Masukkan tidak	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Tanggal Pengiriman : 16-11-2012 Nama *DEX* : Ibu ayi Nama Barang : Kanvas Jumlah : ada 4	boleh kosong atau data masukan tidak valid.”	Masukkan tidak boleh kosong atau data masukan tidak valid.”	

10. Pengujian Penambahan Data Permintaan

Pengujian penambahan data permintaan dilakukan oleh gudang bahan baku pada tabel 4.24

Tabel 4.24 Pengujian Penambahan Data Permintaan Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Permintaan Contoh Masukan : Tanggal Permintaan : 12/02/2012 Nama *DEX* : Ibu sri amd Nama Barang : Purple red Jumlah : 27	Mengisikan data permintaan dan menyimpan data.	Data permintaan dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data operator dapat dilihat pada tabel 4.25

Tabel 4.25 Pengujian Penambahan Data permintaan Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah Permintaan Contoh Masukan : Tanggal Permintaan :	Menampilkan pesan “data tidak valid! Data Masukkan tidak boleh kosong atau	Dapat menampilkan pesan kesalahan “Data tidak valid Data Masukkan tidak	[√]diterima [ ]ditolak

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
(kosong) Nama *DEX* : (kosong) Nama Barang : (kosong) Jumlah : (kosong)	data masukan tidak valid.”	boleh kosong atau data masukan tidak valid.”	
Data Tambah Permintaan Contoh Masukan : Tanggal Permintaan : 12/02/2012 Nama *DEX* : Ibu sri amd Nama Barang : Purple red Jumlah : gakada	Menampilkan pesan “Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid.”	[√]diterima [ ]ditolak

11. Pengujian Penambahan Data *Supplier*

Pengujian penambahan data *Supplier* dilakukan oleh gudang bahan baku pada tabel 4.26

Tabel 4.26 Pengujian Penambahan Data *Supplier* Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah *Supplier* Contoh Masukan : Nama : Indramayutex Alamat : Jl babakan hilir No. Telepon : 086477388294	Mengisikan data *Supplier* dan menyimpan data.	Data Supplier dapat tersimpan.	[√]diterima [ ]ditolak

Pengujian data salah penambahan data operator dapat dilihat pada tabel 4.27

Tabel 4.27 Pengujian Penambahan Data *Supplier* Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Data Tambah *Supplier* Contoh Masukan : Nama : (kosong) Alamat : (kosong) No. Telepon : (kosong)	Menampilkan pesan "data tidak valid! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan "Data tidak valid Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak
Data Tambah *Supplier* Contoh Masukan : Nama : Indramayutex Alamat : Jl babakan hilir No. Telepon : adddada	Menampilkan pesan "Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	Dapat menampilkan pesan kesalahan " Maaf data gagal ditambahkan! Data Masukkan tidak boleh kosong atau data masukan tidak valid."	[√]diterima [ ]ditolak

12. Pengujian Cetak Laporan

Pengujian Cetak Laporan dilakukan oleh admin, pada tabel 4.28

Tabel 4.28 Pengujian Cetak Laporan Data Benar

Kasus dan hasil uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Laporan Contoh Masukan : Dari Tanggal : 12/12/2012 Sampai Tanggal : 12/12/2012	Menampilkan form laporan Print	Menampilkan form Laporan Print	[√]diterima [ ]ditolak

Pengujian cetak laporan data salah dapat dilihat pada tabel 4.29.

Tabel 4.29 Pengujian Cetak Laporan Data Salah

Kasus dan hasil uji (Data salah)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Laporan Contoh Masukan : Dari Tanggal : (kosong) Sampai Tanggal : (kosong)	Menampilkan pesan "Tanggal Tidak Boleh Kosong"	Dapat menampilkan pesan kesalahan "Tanggal Tidak Boleh Kosong"	[√]diterima [ ]ditolak

4.4.3. Kesimpulan Pengujian *Black Box*

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 langsung ke lapangan, dengan mengadakan wawancara mengenai kepuasan *user* terhadap sistem yang telah dibangun, dengan kandungan point syarat *user friendly* serta apakah sesuai dengan yang diharapkan.

4.5.1. Skenario Pengujian Beta

Pengujian *betha* merupakan pengujian yang dilakukan dengan tujuan untuk mengetahui sejauh mana kualitas sistem, apakah sudah memenuhi harapan atau belum. Untuk itu dalam pengujian *betha* dilakukan penelitian terhadap calon pengguna sistem dengan melakukan pengumpulan data menggunakan metodologi wawancara.

Pada penelitian ini pengujian *betha* dilakukan dengan menggunakan metodologi wawancara yang diberikan kepada sebagian pengguna dengan mengambil sampel calon pengguna yaitu bagian gudang bahan baku, dan *Costumer*

Service. Hasil wancara tersebut bisa diambil kesimpulan terhadap penilaian terhadap sistem yang baru.

A. Wawancara dengan bagian gudang bahan baku di Maika-Etnik

1. Apakah sistem ini membantu dalam memberikan informasi tentang ketersedian bahan baku?

   Jawaban : Sistem ini sangat membantu bagian penyedia bahan baku, bagian produksi, dan bagian gudang barang karena kita tidak harus menghitung ulang bahan baku dan bahan produksi ataupun barang di gudang barang

2. Apakah sistem ini membantu dalam perhitungan pembelian bahan baku ?

   Jawaban : Sistem ini sangat membantu dalam perhitungan pembelian bahan baku, karena sistem ini menentukan bahan baku yang harus dipesan, menentukan titik pemesanan kembali sehingga stok bahan yang ada di gudang tidak menumpuk maupun kekurangan.

3. Apakah sistem ini membantu perhitungan nilai ekonomis pemesanan bahan baku ?

   Jawaban : Sistem ini sangat membantu bagian bahan baku dalam perhitungan dalam pemesanan bahan baku, karena ini pada dasarnya sama seperti perhitungan yang kita lakukan, perhitungan berapa jumlah bahan baku yang harus dipesan tiap periodenya, EOQ ini cocok.

4. Apakah sistem ini mudah digunakan ?

   Jawaban : Sistem ini mudah digunakan karena menunya jelas, kita login juga sesuai bagian dan pekerjaan kita

B. Wawancara dengan bagian costumer service di Maika-Etnik

1. Apakah sistem ini membantu dalam Memberikan informasi tentang ketersedian bahan baku, ketersediaan barang, pengiriman ke setiap bagiannya ?

   Jawaban : Sistem ini sangat membantu bagian costumer service, karena kita tidak harus menghitung ketersediaan barang di gudang barang karena sudah terintegrasi.

2. Apakah sistem ini membantu proses pembuatan laporan permintaan dan pengiriman?

Jawaban : sistem ini sangat membantu, biasanya kita harus membuat dan print laporan dengan jumlah yang banyak, sekarang kita tinggal masukkan data permintaan dan pengiriman yang terkirim.

C. Wawancara dengan bagian gudang barang di Maika-Etnik

2. Apakah sistem ini membantu dalam memberikan informasi tentang ketersedian barang?

   Jawaban : Sistem ini lumayan membantu bagian penyedia gudang barang, karena pada saat ada permintaan kita dapat mengetahui ketersediaan barang secara cepat.

2. Apakah sistem ini membantu dalam pengelolaan ketersediaan barang ?

   Jawaban : Sistem ini sangat membantu dalam penggelolaan ketersediian barang karena mudah untuk update stok dan mengetahui barang apa saja yang harus di produksi.

D. Wawancara dengan bagian produksi di Maika-Etnik

1. Apakah sistem ini membantu dalam Memberikan informasi tentang proses produksi ke setiap bagiannya ?

   Jawaban : Sistem ini sangat membantu bagian produksi, karena kita tidak harus mengunakan perkiraan pada saat proses produksi. Kita hanya melihat ketersedian barang.

2. Apakah sistem ini membantu proses pembuatan produksi?

   Jawaban : sistem ini sangat membantu, karena kita langsung bisa mengetahui bahan baku yang tersedia di gudang bahan baku.

4.5.2 Kesimpulan Hasil Pengujian *Betha*

Dari hasil pengujian *betha* dapat disimpulkan bahwa Pembangunan sistem ini membantu Maika-Etnik dalam mengolah data disetiap bagiannya, nilai ekonomis pemesanan yang ada dalam sistem ini juga membantu menghitung nilai akurat pembelian bahan baku sehingga tidak terjadinya pemborosanyang sangat besar pada saat pembelian bahan baku dan tidak juga kekurangan bahan baku.