# BAB V

# PENGUJIAN DAN IMPLEMENTASI SISTEM

## 5.1. Implementasi

Implementasi adalah suatu proses penerapan rancangan program yang telah dibuat ke dalam sebuah aplikasi pemrograman sesuai dengan tujuan yang diharapkan dari program aplikasi tersebut. Kegiatan implementasi dilakukan dengan dasar yang telah direncanakan dalam tahap perancangan.

### 5.1.1. Batasan Implementasi

Dalam pengimplementasian perangkat lunak sistem informasi pelayanan pasien ini terdapat beberapa hal yang menjadi batasan implementasi, yaitu :

1. Basis data yang digunakan dalam mengimplementasikan sistem informasi pelayanan pasien ini adalah My SQL.
2. Sistem informasi pelayanan pasien ini dapat dijalankan pada system operasi berbasis Windows dan pengimplementasian hanya dilakukan pada Microsoft Windows 7 Service Pack 1

### 5.1.2. Implementasi Perangkat Lunak

Untuk mendukung kelancaran pengembangan sistem informasi pelayanan pasien ini selain diperlukan perangkat keras juga diperlukan perangkat lunak. Perangkat lunak yang digunakan sebagai pendukung sistem informasi ini adalah Java Virtual Machine, Jasper Report, iReport dan Xampp.

### 5.1.3. Implementasi Perangkat Keras

Perangkat keras yang dibutuhkan yang harus terpenuhi untuk dapat menjalankan sistem aplikasi ini antara lain:

1. Komputer Server
    a. Prosesor yang digunakan dengan kecepatan 1,80 GHz
    b. Menggunakan RAM minimal 512MB
    c. Hardisk 180 GB dan disarankan minimal 80 GB
    d. Mouse, keyboard dan Monitor sebagai media peralatan antarmuka
2. Komputer Client
    a. Prosesor yang disarankan minimal berkecepatan 800 MHz
    b. RAM yang disarankan minimal 512MB
    c. Hardisk yang disarankan minimal 80 GB
    d. Mouse, keyboard, dam monitor sebagai media peralatan antarmuka
    e. Printer
3. Jaringan
    a. Wireless (Ad-Hoc)

### 5.1.4. Implementasi Basis Data

Untuk pengembangan basis data pada sistem informasi pelayanan pasien ini menggunakan My SQL. Adapun implementasi pembuatan basis data adalah sebagai berikut:

1. Pembuatan Database

```
CREATE DATABASE db_pelayanan;
```

2. Sintaks tabel tb_admin

```
CREATE TABLE IF NOT EXISTS `tb_admin` (
`username` varchar(30) NOT NULL,
`password` varchar(30) NOT NULL,
 `no` int(11) NOT NULL AUTO_INCREMENT,
PRIMARY KEY (`no`)
) ENGINE=InnoDB  DEFAULT CHARSET=latin1 AUTO_INCREMENT=2;
```

3. Sintaks tabel tb_anak

```
CREATE TABLE IF NOT EXISTS `tb_anak` (
  `no_register_anak` int(20) NOT NULL DEFAULT '0',
  `tanggal` varchar(14) DEFAULT NULL,
  `nama` varchar(30) DEFAULT NULL,
  `tgl_lahir` varchar(30) DEFAULT NULL,
  `j_kelamin` varchar(10) DEFAULT NULL,
  `j_persalinan` varchar(20) DEFAULT NULL,
  `p_persalinan` varchar(50) DEFAULT NULL,
  `t_persalinan` varchar(30) DEFAULT NULL,
```

```
  `asi` varchar(10) DEFAULT NULL,
  `bb` int(8) NOT NULL,
  `umur` varchar(6) NOT NULL,
  `no_register_ibu` int(16) NOT NULL,
  PRIMARY KEY (`no_register_anak`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4. Sintaks tabel tb_bidan

```
CREATE TABLE IF NOT EXISTS `tb_bidan` (
  `nama` varchar(30) NOT NULL,
  `alamat` varchar(30) NOT NULL,
  `kode` varchar(14) NOT NULL,
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

5. Sintaks tabel tb_ibu_hamil

```
CREATE TABLE IF NOT EXISTS `tb_ibu_hamil` (
  `no_register` int(10) NOT NULL DEFAULT '0',
  `tanggal` varchar(20) DEFAULT NULL,
  `no_kunjungan` int(16) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `hpht` varchar(16) NOT NULL,
  `hpt` varchar(16) NOT NULL,
  `G` int(3) NOT NULL,
  `P` int(3) NOT NULL,
```

```
`A` int(3) NOT NULL,
 `riwayat` varchar(50) NOT NULL,
 PRIMARY KEY (`no_kunjungan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

6. Sintaks tabel tb_kb

```
CREATE TABLE IF NOT EXISTS `tb_kb` (
 `no_kunjungan` int(16) NOT NULL,
 `tanggal` varchar(16) NOT NULL,
 `no_register` int(16) NOT NULL,
 `nama` varchar(30) NOT NULL,
 `macam_kb` varchar(30) NOT NULL,
 `kb_akhir` varchar(30) NOT NULL,
 `j_kb` varchar(30) NOT NULL,
 PRIMARY KEY (`no_kunjungan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

7. Sintaks tabel tb_kunjungan

```
CREATE TABLE IF NOT EXISTS `tb_kunjungan` (
 `no_kunjungan` int(16) NOT NULL,
 `no_register` int(16) NOT NULL,
 `kode_p` int(16) NOT NULL,
 `tanggal` varchar(16) NOT NULL,
 PRIMARY KEY (`no_kunjungan`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

8. Sintaks tabel tb_layanan

```
CREATE TABLE IF NOT EXISTS `tb_layanan` (
  `kode` int(16) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `tarif` int(12) NOT NULL,
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

9. Sintaks tabel tb_obat

```
CREATE TABLE IF NOT EXISTS `tb_obat` (
  `kode` int(16) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `harga_d` int(12) NOT NULL,
  `harga_j` int(12) NOT NULL,
  `kode_s` int(16) NOT NULL,
  `stok` int(10) NOT NULL,
  PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

10. Sintaks tabel tb_p_anak

```
CREATE TABLE IF NOT EXISTS `tb_p_anak` (
  `no` int(10) NOT NULL,
  `tanggal` varchar(10) NOT NULL,
  `umur` int(2) NOT NULL,
  `bb` int(2) NOT NULL,
```

```
  `makanan` varchar(30) NOT NULL,
  `gejala` varchar(30) NOT NULL,
  `nasehat` varchar(30) NOT NULL,
  `obat` varchar(30) NOT NULL,
  `no_register` int(6) NOT NULL,
  `kode_bidan` int(16) NOT NULL,
  PRIMARY KEY (`no`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

11. Sintaks tabel tb_p_ibu_hamil

```
CREATE TABLE IF NOT EXISTS `tb_p_ibu_hamil` (
  `no_register` int(16) NOT NULL,
  `tanggal` varchar(16) NOT NULL,
  `keluhan` varchar(50) NOT NULL,
  `darah` varchar(50) NOT NULL,
  `bb` varchar(3) NOT NULL,
  `u_hamil` varchar(50) NOT NULL,
  `t_fundus` varchar(50) NOT NULL,
  `l_janin` varchar(50) NOT NULL,
  `d_janin` varchar(50) NOT NULL,
  `lab` varchar(50) NOT NULL,
  `p_khusus` varchar(50) NOT NULL,
  `tindakan` varchar(50) NOT NULL,
  `no` int(5) NOT NULL,
```

```
  `kode_bidan` int(16) NOT NULL,
  PRIMARY KEY (`no`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

12. Sintaks tabel tb_p_kb

```
CREATE TABLE IF NOT EXISTS `tb_p_kb` (
  `no` int(16) NOT NULL,
  `no_register` int(16) NOT NULL,
  `tanggal` varchar(16) NOT NULL,
  `haid_akhir` varchar(16) NOT NULL,
  `bb` int(3) NOT NULL,
  `darah` varchar(10) NOT NULL,
  `keluhan` varchar(30) NOT NULL,
  `tindakan` varchar(30) NOT NULL,
  `kode_bidan` int(4) NOT NULL,
  PRIMARY KEY (`no`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

13. Sintaks tabel tb_partus

```
CREATE TABLE IF NOT EXISTS `tb_partus` (
  `tanggal` varchar(15) DEFAULT NULL,
  `lahir_jam` varchar(15) DEFAULT NULL,
  `anak_ke` int(3) DEFAULT NULL,
  `no_register` int(15) NOT NULL DEFAULT '0',
  `no_kunjungan` int(16) NOT NULL,
```

```
  `pb` varchar(16) NOT NULL,
  `kode_bidan` int(16) NOT NULL,
  `kode_anak` int(16) NOT NULL,
  PRIMARY KEY (`no_register`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

14. Sintaks tabel tb_pasien

```
CREATE TABLE IF NOT EXISTS `tb_pasien` (
  `no_register` int(16) NOT NULL,
  `tanggal` varchar(16) NOT NULL,
  `nama_pasien` varchar(30) NOT NULL,
  `alamat` varchar(50) NOT NULL,
  `no_telp` varchar(16) NOT NULL,
  `umur` int(3) NOT NULL,
  `pek_p` varchar(30) NOT NULL,
  `nama_s` varchar(30) NOT NULL,
  `pek_s` varchar(30) NOT NULL,
  `umur_s` int(3) NOT NULL,
  `riwayat_penyakit` varchar(30) NOT NULL,
  PRIMARY KEY (`no_register`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

15. Sintaks tabel tb_pembelian_obat

```
CREATE TABLE IF NOT EXISTS `tb_pembelian_obat` (
  `no_t` int(16) NOT NULL,
```

```
  `tanggal` varchar(16) NOT NULL,
  `kode_supplier` int(16) NOT NULL,
  `kode_obat` int(16) NOT NULL,
  `jumlah` int(16) NOT NULL,
  `total` bigint(16) NOT NULL,
  PRIMARY KEY (`no_t`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

16. Sintaks tabel tb_penjualan_obat

```
CREATE TABLE IF NOT EXISTS `tb_penjualan_obat` (
  `no_t` int(16) NOT NULL,
  `no_register` int(16) NOT NULL,
  `kode_obat` varchar(16) NOT NULL,
  `tanggal` varchar(16) NOT NULL,
  `jumlah` int(16) NOT NULL,
  `total` bigint(20) NOT NULL,
  PRIMARY KEY (`no_t`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

17. Sintaks tabel tb_supplier

```
CREATE TABLE IF NOT EXISTS `tb_supplier` (
  `kode` int(16) NOT NULL,
  `nama` varchar(30) NOT NULL,
  `alamat` varchar(30) NOT NULL,
  `no_telp` int(16) NOT NULL,
```

```
PRIMARY KEY (`kode`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

### 5.1.5. Implementasi Antar Muka

Implementasi antarmuka dilakukan dengan setiap tampilan program yang dibuat dan pengkodeannya dalam bentuk file program. Tahap implementasi sistem merupakan tahap penerapan sistem agar dapat dioperasikan secara optimal sesuai kebutuhan.

1. Form Login

Pada form login terdiri dari textbox username dan textbox password serta button login. Button login berfungsi sebagai button untuk masuk kedalam form menu utama.




Gambar 5.1 Form Login

2. Menu Admin

Form Menu Admin terdiri dari :

1. Master Data, terdiri dari master data pelayanan, master data obat, master data supplier dan master data bidan.
2. Pendaftaran terdiri dari input kunjungan, pendaftaran pasien ibu dan pendaftaran pasien anak.

3. Pembelian obat, input data pembelian obat.
4. Penjualan obat, input data penjualan obat.
5. Kadaluwarsa obat, mengetahui data obat yang sudah kadaluwarsa.
6. Laporan, untuk mencetak laporan.
7. Logout, untuk keluar dari menu utama ke form login.




Gambar 5.2 Form Menu Admin

3. Form Menu Bidan

Form Menu Bidan terdiri dari :

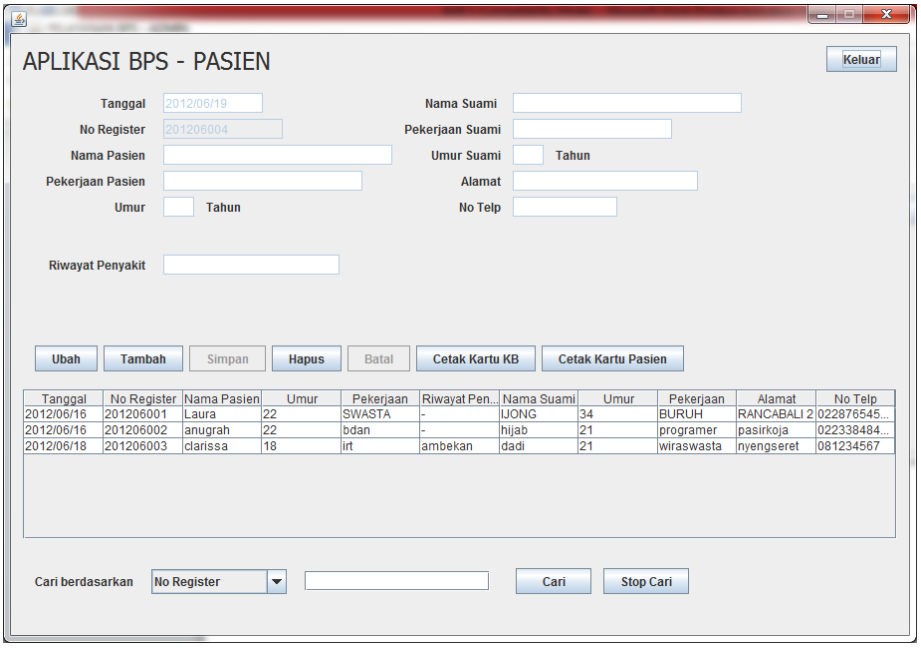
1. Pemeriksaan anak,input data pemeriksaan anak.
2. Pemeriksaan ibu hamil, input data pemeriksaan ibu hamil.
3. Pemeriksaan KB, input data pemeriksaan KB.
4. Pemeriksaan partus, input data pemeriksaan partus.
5. Penjualan Obat, input obat yang diperlukan untuk pasien.




Gambar 5.3 Form Menu Bidan

4. Form Input Data Pasien

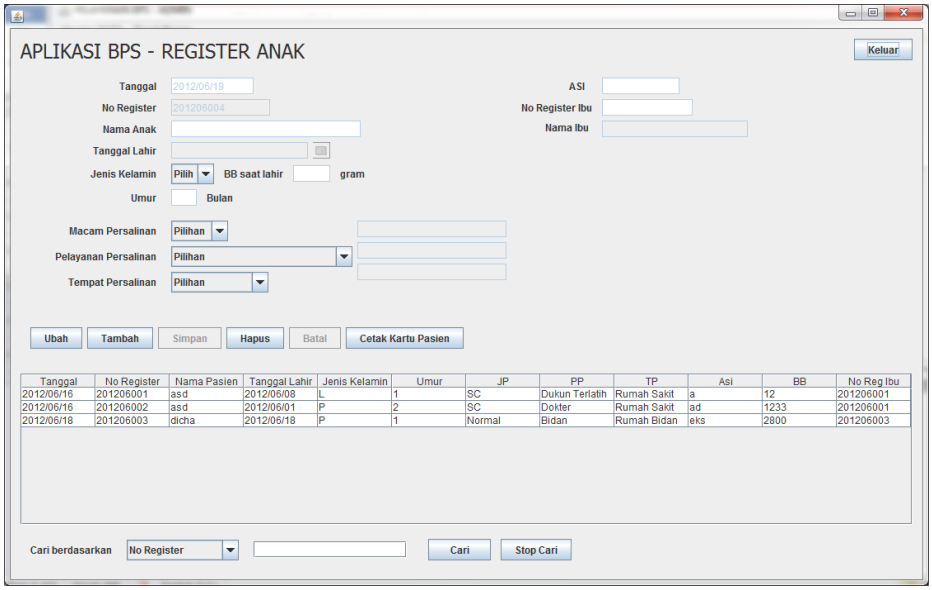
Form Input Data Pasien digunakan untuk input data pasien. Untuk menginput data pasien maka klik button tambah lalu form isian akan aktif. Untuk mengubah data, klik salah satu data pada tabel kemudian klik tombol edit dan diubah. Jika data telah diubah maka untuk menyimpan data yang telah diubah klik button simpan dan data akan tersipan didalam database. Untuk menghapus data, klik salah satu data pasien pada tabel kemudian klik tombol hapus dan data akan terhapus. Jika ingin keluar maka klik tombol dan akan kembali ke menu utama.




Gambar 5.4 Form Input Data Pasien

5. Form Input Data Pasien Anak

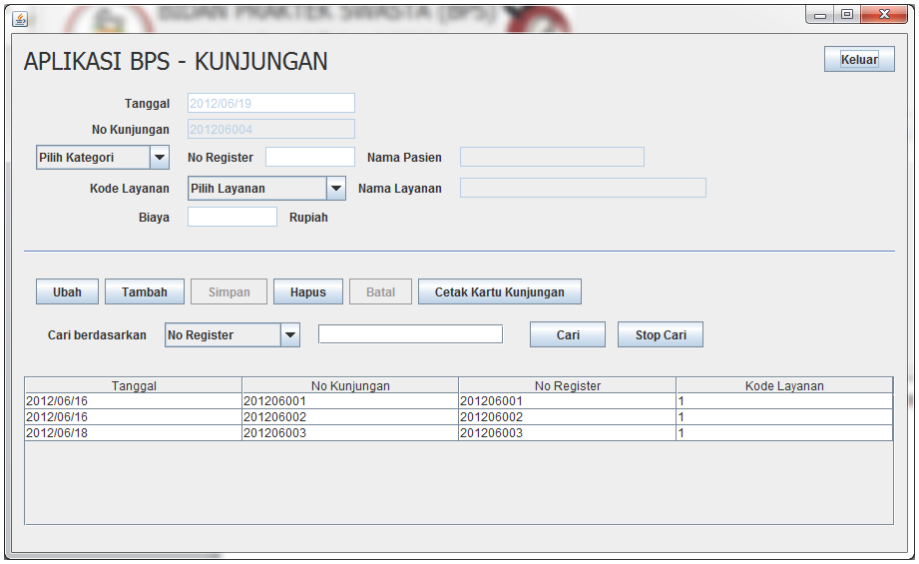
Form Input Data Pasien Anak digunakan untuk input data pasien anak. Untuk menginput data pasien anak maka klik button tambah lalu form isian akan aktif. Untuk mengubah data, klik salah satu data pada tabel kemudian klik tombol edit dan diubah. Jika data telah diubah maka untuk menyimpan data yang telah diubah klik button simpan dan data akan tersipan didalam database. Untuk menghapus data, klik salah satu data pasien pada tabel kemudian klik tombol hapus dan data akan terhapus. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.5 Form Input Data Pasien Anak

6. Form Input Kunjungan

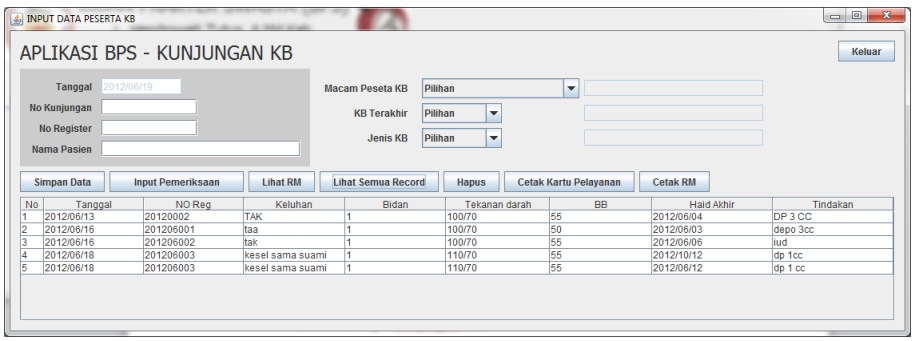
Form Input Kunjungan digunakan untuk input data kunjungan pasien. Untuk menginput data pasien anak maka klik button tambah lalu form isian akan aktif. Untuk mengubah data, klik salah satu data pada tabel kemudian klik tombol edit dan diubah. Jika data telah diubah maka untuk menyimpan data yang telah diubah klik button simpan dan data akan tersipan didalam database. Untuk menghapus data, klik salah satu data pasien pada tabel kemudian klik tombol hapus dan data akan terhapus. Untuk cetak kartu kunjungan, klik salah satu data pasien pada tabel lalu klik cetak kartu kunjungan. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.6 Form Input Kunjungan

7. Form Input Kunjungan KB

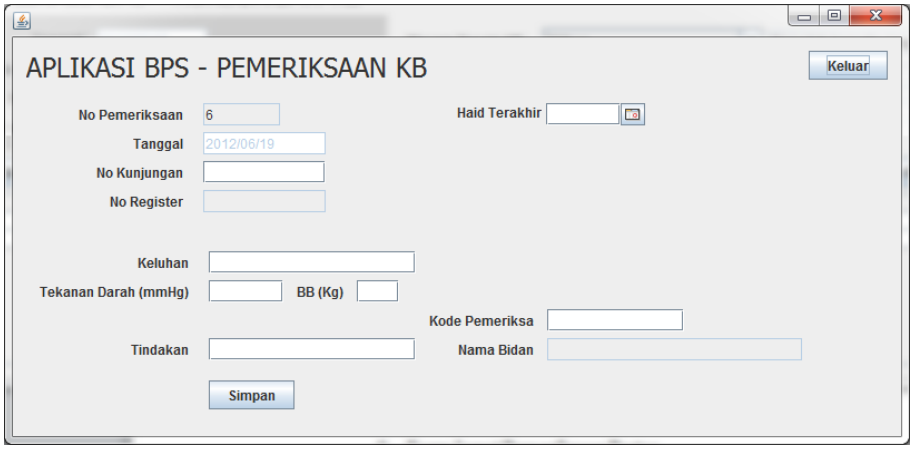
Form Input Kunjungan KB digunakan untuk input data kunjungan pasien KB. Untuk menginput data kunjungan, input nomor kunjungan, bila pasien telah mendaftar maka no register pasien dan nama akan langsung muncul, lalu bisa diteruskan input data, lalu klik button simpan data dan data akan tersipan didalam database. Klik tombol input pemeriksaan untuk menginputkan pemeriksaan KB. Untuk cetak kartu pelayanan dan rekam medik, pastikan semua field terisi.. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.7 Form Input Kunjungan KB

8. Form Input Pemeriksaan KB

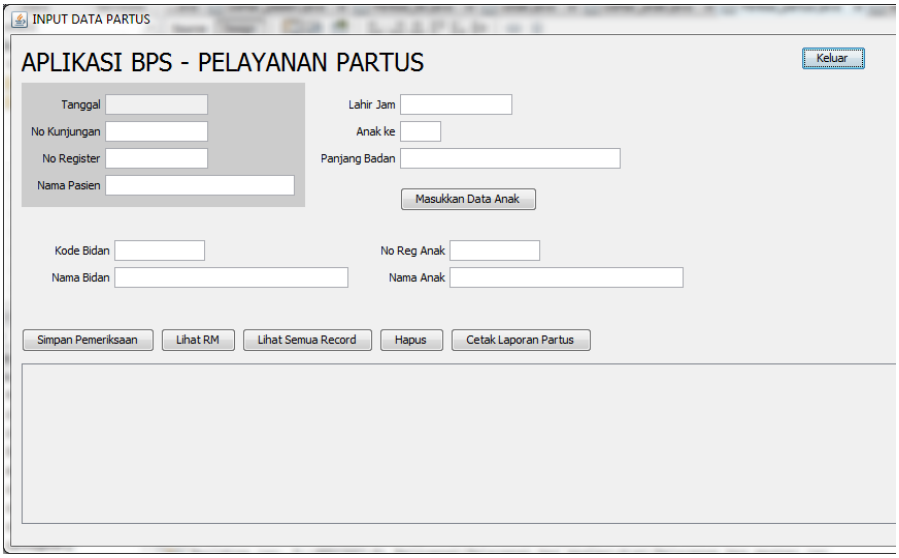
Form Input Pemeriksaan KB digunakan untuk input data pemeriksaan per-kunjungan pasien KB. Untuk menginput data, input nomor kunjungan, maka no register pasien dan nama akan langsung muncul, lalu bisa diteruskan input data, lalu klik button simpan data dan data akan tersipan didalam database. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.8 Form Input Pemeriksaan KB

9. Form Input Pemeriksaan Partus

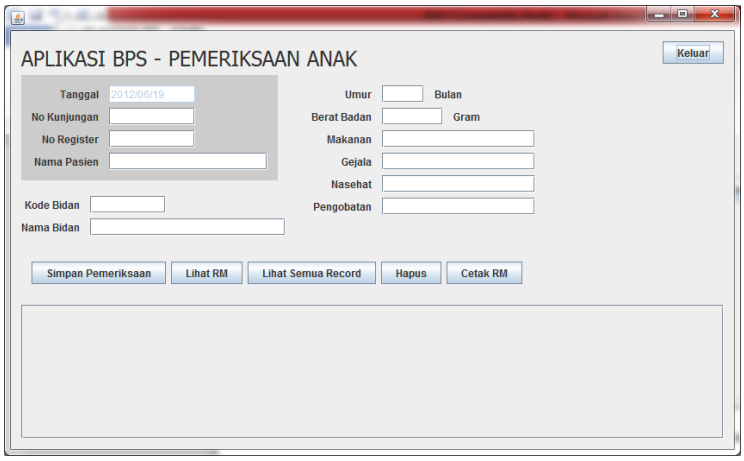
Form Input Pemeriksaan Partus digunakan untuk input data pemeriksaan partus. Untuk menginput data, input nomor kunjungan, bila pasien telah mendaftar maka no register pasien dan nama akan langsung muncul, lalu bisa diteruskan input data. Data anak di input melalui klik tombol masukkan data anak, lalu klik button simpan pemeriksaan dan data akan tersipan didalam database. Untuk cetak laporan partus, pastikan semua field terisi. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.9 Form Input Pemeriksaan Partus

10. Form Input Pemeriksaan Anak

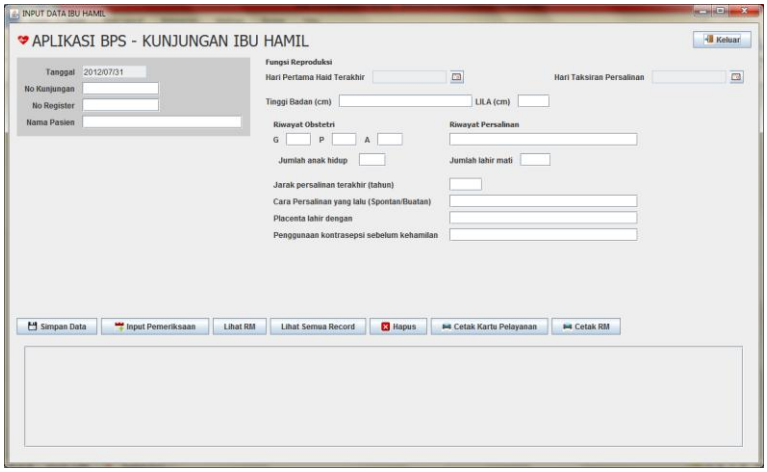
Form Input Pemeriksaan Anak digunakan untuk input data pemeriksaan anak. Untuk menginput data, input nomor kunjungan, bila pasien telah mendaftar maka no register pasien dan nama akan langsung muncul, lalu bisa diteruskan input data, lalu klik button simpan pemeriksaan dan data akan tersipan didalam database. Untuk cetak rekam medik, pastikan semua field terisi. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.10 Form Input Pemeriksaan Anak

11. Form Input Kunjungan Ibu Hamil

Form Input Kunjungan Ibu Hamil digunakan untuk input data kunjungan pasien ibu hamil. Untuk menginput data kunjungan, input nomor kunjungan, bila pasien telah mendaftar maka no register pasien dan nama akan langsung muncul, lalu bisa diteruskan input data, lalu klik button simpan data dan data akan tersipan didalam database. Klik tombol input pemeriksaan untuk menginputkan pemeriksaan ibu hamil. Untuk cetak kartu pelayanan dan rekam medik, pastikan semua field terisi.. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.11 Form Input Kunjungan Ibu Hamil

12. Form Input Pemeriksaan Ibu Hamil

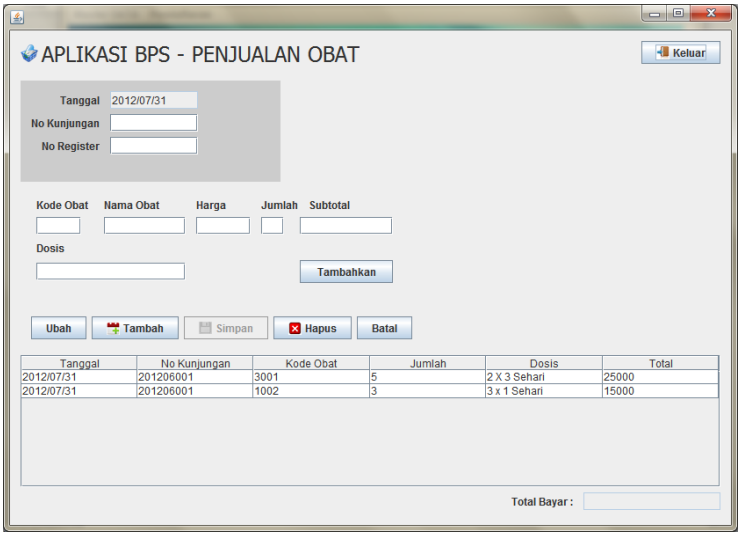
Form Input Pemeriksaan Ibu hamil digunakan untuk input data pemeriksaan per-kunjungan pasien ibu hamil. Untuk menginput data, input nomor kunjungan, maka no register pasien dan nama akan langsung muncul, lalu bisa diteruskan input data, lalu klik button simpan data dan data akan tersipan didalam database. Jika ingin keluar maka klik tombol keluar dan akan kembali ke menu utama.




Gambar 5.12 Form Input Pemeriksaan Ibu Hamil

13. Form Penjualan Obat

Form Penjualan Obat berfungsi untuk input data penjualan obat. Pilih kategori pasien, lalu input no register. Lalu masukkan kode obat dan jumlah, lalu klik subtotal untuk menghitung biaya. Klik simpan untuk selesai, bila masih ada yang dijual dengan no register yang sama, klik tombol tambahkan.




Gambar 5.13 Form Penjualan Obat

14. Form Pembelian Obat

Form Pembelian Obat berfungsi untuk input data pembelian obat.Ketik kode supplier. Lalu masukkan kode obat dan jumlah, lalu klik subtotal untuk menghitung biaya. Klik simpan untuk selesai, bila masih ada yang dijual dengan supplier yang sama, klik tombol tambahkan.




Gambar 5.14 Form Pembelian Obat

15. Form Cetak Laporan

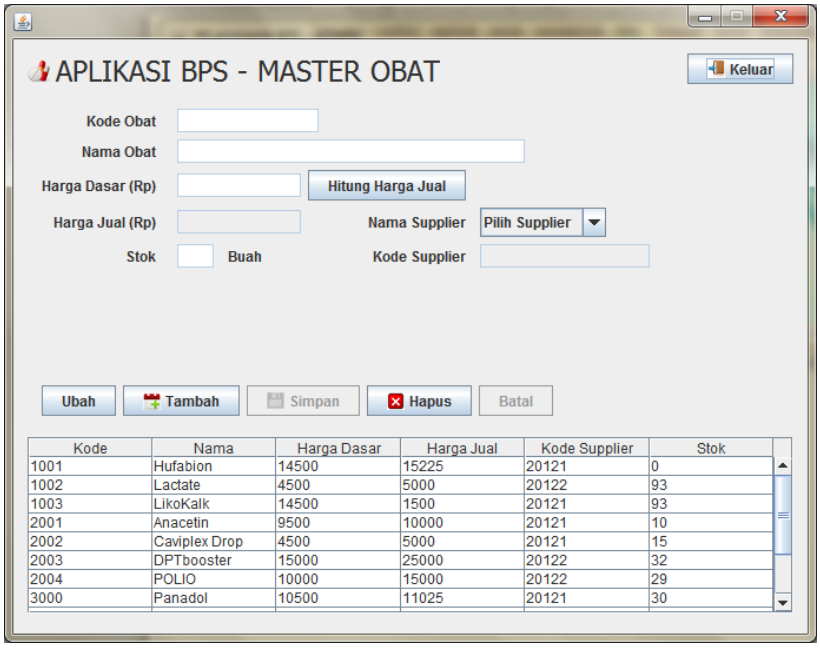
Form Cetak Laporan berfungsi untuk input tanggal mulai dan akhir dari laporan yang akan dibuat. Masukan dari tanggal dan sampai tanggal, lalu klik laporan yang akan dibuat. Untuk keluar dan ke menu utama klik tombol close.




Gambar 5.15 Form Cetak Laporan

16. Form Master Obat

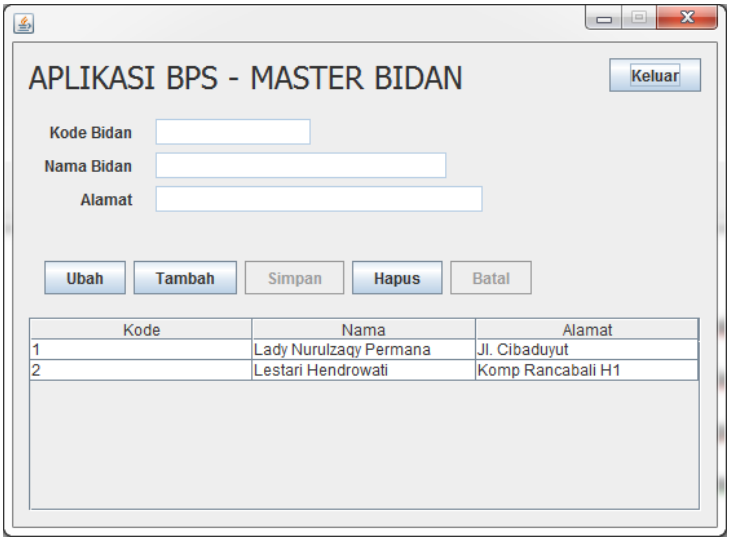
Form Master obat berfungsi untuk memanipulasi data master obat. Tersedia tombol tambah untuk menambah data. Simpan untuk menyimpan data yang sudah di input. Ubah untuk mengubah data. Hapus untuk mengubah data. Keluar untuk kembali ke menu utama.




Gambar 5.16 Form Master Obat

17. Form Master Bidan

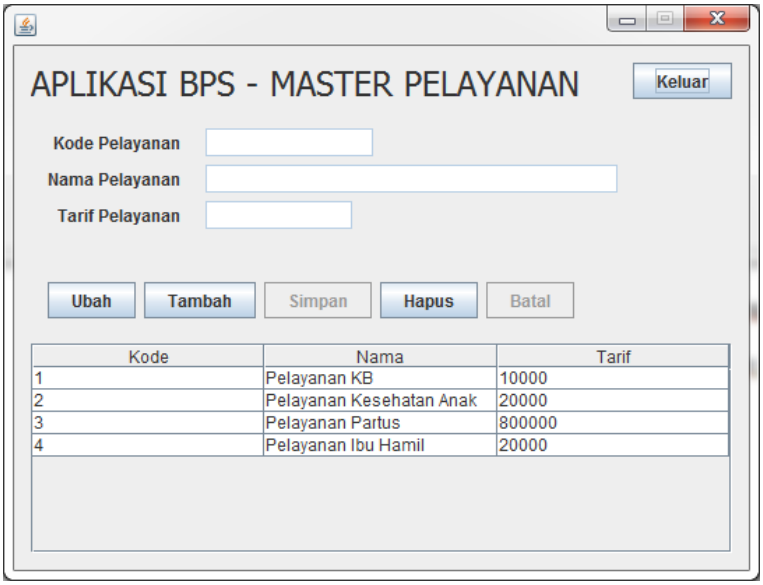
Form Master bidan berfungsi untuk memanipulasi data master bidan. Tersedia tombol tambah untuk menambah data. Simpan untuk menyimpan data yang sudah di input. Ubah untuk mengubah data. Hapus untuk mengubah data. Keluar untuk kembali ke menu utama.




Gambar 5.17 Form Master Bidan

18. Form Master Pelayanan

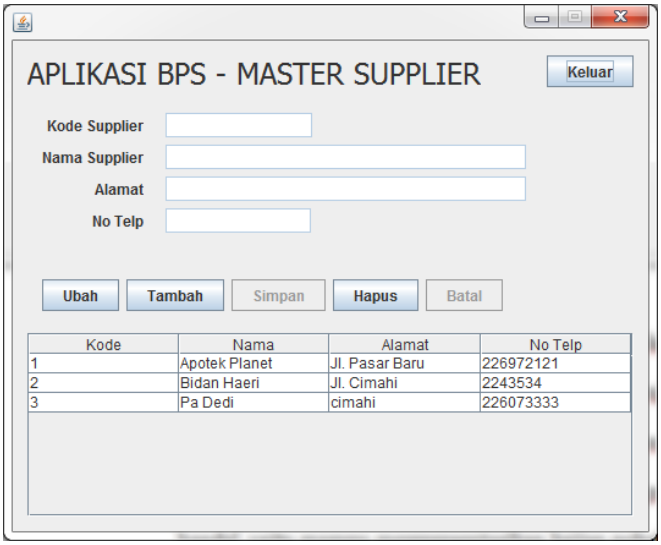
Form Master pelayanan berfungsi untuk memanipulasi data master pelayanan. Ada tombol tambah untuk menambah data. Simpan untuk menyimpan data yang sudah di input. Ubah untuk mengubah data. Hapus untuk mengubah data. Keluar untuk kembali ke menu utama.




Gambar 5.18 Form Master Pelayanan

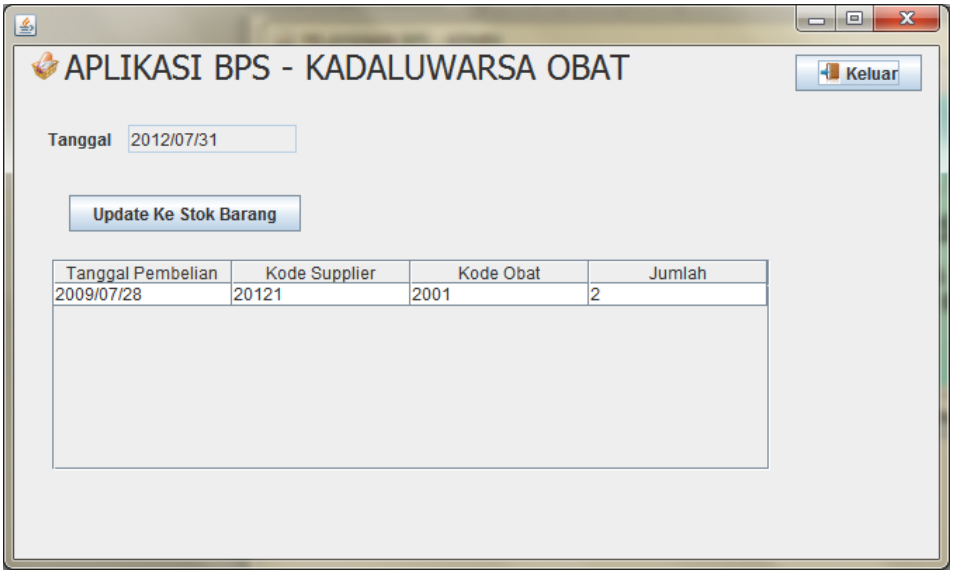
19. Form Master Supplier

Form Master supplier berfungsi untuk memanipulasi data master supplier. Ada tombol tambah untuk menambah data. Simpan untuk menyimpan data yang sudah di input. Ubah untuk mengubah data. Hapus untuk mengubah data. Keluar untuk kembali ke menu utama.




Gambar 5.19 Form Master Supplier

20. Form Kadaluwarsa Obat




Gambar 5.20 Form Kadaluwarsa Obat

21. Laporan Pembelian Obat




Gambar 5.21 Tampilan Laporan Pembelian Obat

22. Laporan Penjualan Obat



**Bidan Praktek Swasta (BPS)**

*L. Hendrowati Tutus, A.Md.Keb*

Jl. Batukarang H1 Komp. Rancabali I & II
Cimahi Utara - Cimahi Tlp. (022) 662 8170

Bidan Delima
PELAYANAN BERKUALITAS

LAPORAN PENJUALAN OBAT

| Kode Obat | Jumlah Obat | Subtotal |
|---|---|---|
| **Tanggal** 2012/06/13 | | |
| No Reg :20120001 | | |
| 1001 | 2 | Rp 30,000.00 |
| No Reg :20122001 | | |
| 2001 | 1 | Rp 10,000.00 |
| | **TOTAL** | **Rp 40,000.00** |
| **Tanggal** 2012/06/16 | | |
| No Reg :201206001 | | |
| 2001 | 2 | Rp 20,000.00 |
| 1001 | 2 | Rp 30,000.00 |
| | **TOTAL** | **Rp 50,000.00** |

Gambar 5.22 Tampilan Laporan Penjualan Obat

### 5.1.6. Implementasi Instalasi Program

Ada dua komponen utama dalam sistem ini, yaitu *Program* dan *database*. Untuk *program* bisa di copy di D:\SKRIPSI\Si_Pelayanan\Pelayanan_bps_master\

## 5.2. Pengujian

Pengujian merupakan bagian yang penting dalam siklus pembangunan perangkat lunak.Pengujian dilakukan untuk menjamin kualitas dan juga mengetahui kelemahan dari perangkat lunak.Tujuan dari pengujian ini adalah untuk menjamin bahwa perangkat lunak yang dibangun memiliki kualitas yang

handal, yaitu mampu merepresentasikan kajian pokok dari spesifikasi, analisis, perancangan dan pengkodean dari perangkat lunak itu sendiri.

Pada Pengujian perangkat lunak ini, penulis menggunakan suatu metode pengujian yang berfokus pada persyaratan fungsional perangkat lunak yang dibangun.Metode yang diambil adalah metode pengujian Black Box.Pengujian Black Box adalah pengujian yang sistemnya tanpa memperhatikan struktur logika internal perangkat lunak.Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar.

Pada metode ini data uji dibangkitkan, dieksekusi pada perangkat lunak dan kemudian keluaran dari perangkat lunak dicek apakah telah sesuai dengan yang diharapkan. Ada dua komponen yang harus diperhatikan dalam strategi pengujian, yaitu :

1. Faktor Pengujian yang merupakan hal-hal yang harus diperhatikan selama melakukan pengujian. Faktor pengujian ini dipilih sesuai dengan sistem yang akan diuji.
2. Tahapan pengujian yang merupakan langkah-langkah dalam melakukan pengujian.

### 5.2.1. Rencana Pengujian

Proses rencana pengujian meliputi *input/output.* Proses pengujian *input/output* adalah mencoba program dengan memasukan data ke dalam form-form masukan yang telah disediakan. Pengujian dalam penelitian ini dilakukan oleh pihak user atau pengguna yaitu petugas bagian Administrasi dan Bidan.

Berikut ini adalah rencana pengujian sistem informasi pelayanan pasien :

Tabel 5.1 Tabel Rencana Pengujian

| Kelas Uji | Butir Uji | Tingkat Pengujian | Jenis Pengujian |
|---|---|---|---|
| Login | Pengecekan Admin | Sistem | Black Box |
| | Pengecekan Bidan | Sistem | Black Box |
| Pengisian Data | Pengisian Data Pasien | Modul | Black Box |
| | Pengisian Data Bidan | Modul | Black Box |
| | Pengisian Data Obat | Modul | Black Box |
| | Pengisian Data Supplier | Modul | Black Box |
| | Pengisian Data Pelayanan | Modul | Black Box |

### 5.2.2 Kasus dan Hasil Pengujian

Setelah melaksanakan rencana pengujian maka dilanjutkan dengan kasus dan hasil pengujian. Berikut kasus dan hasil pengujian :

Tabel 5.2 Pengecekan Login Admin

| No | Kondisi | Hasil yang diharapkan | Hasil sebenarnya | Kesimpulan |
|---|---|---|---|---|
| 1 | Username: admin<br>Password: admin123 | Tampil menu admin | Tampil menu admin | |
| 2 | Username : “”<br>Password: “” | Tampil pesan peringatan | Tampil pesan peringatan | |
| 3 | Username dan password diisikan karakter acak | Tampil pesan peringatan | Tampil pesan peringatan | |

Tabel 5.3 Pengecekan Login Bidan

| No | Kondisi | Hasil yang diharapkan | Hasil sebenarnya | Kesimpulan |
|---|---|---|---|---|
| 1 | Username: bidan<br>Password: bidan123 | Tampil menu bidan | Tampil menu bidan | |
| 2 | Username : “”<br>Password: “” | Tampil pesan peringatan | Tampil pesan peringatan | |
| 3 | Username dan password diisikan karakter acak | Tampil pesan peringatan | Tampil pesan peringatan | |

Tabel 5.4 Pengecekan Pengisian Data Pasien

| No | Kondisi | Hasil yang diharapkan | Hasil sebenarnya | Kesimpulan |
|---|---|---|---|---|
| 1 | Klik tambah. Semua field diisi dengan data yang benar, klik simpan | Data tersimpan di dalam database | Data tersimpan di dalam database | |
| 2 | Klik tambah. Ada field yang kosong, klik simpan | Muncul window dialog peringatan. | Muncul window dialog peringatan | |

Tabel 5.5 Pengecekan Pengisian Data Bidan

| No | Kondisi | Hasil yang diharapkan | Hasil sebenarnya | Kesimpulan |
|---|---|---|---|---|
| 1 | Klik tambah. Semua field diisi dengan data yang benar, klik simpan | Data tersimpan di dalam database | Data tersimpan di dalam database | |
| 2 | Klik tambah. Ada field yang kosong, klik simpan | Muncul window dialog peringatan. | Muncul window dialog peringatan | |

Tabel 5.6 Pengecekan Pengisian Data Obat

| No | Kondisi | Hasil yang diharapkan | Hasil sebenarnya | Kesimpulan |
|---|---|---|---|---|
| 1 | Klik tambah. Semua field diisi dengan data yang benar, klik simpan | Data tersimpan di dalam database | Data tersimpan di dalam database | |
| 2 | Klik tambah. Ada field yang kosong, klik simpan | Muncul window dialog peringatan. | Muncul window dialog peringatan | |

Tabel 5.7 Pengecekan Pengisian Data Supplier

| No | Kondisi | Hasil yang diharapkan | Hasil sebenarnya | Kesimpulan |
|---|---|---|---|---|
| 1 | Klik tambah. Semua field diisi dengan data yang benar, klik simpan | Data tersimpan di dalam database | Data tersimpan di dalam database | |
| 2 | Klik tambah. Ada field yang kosong, klik simpan | Muncul window dialog peringatan. | Muncul window dialog peringatan | |

Tabel 5.8 Pengecekan Pengisian Data Pelayanan

| No | Kondisi | Hasil yang diharapkan | Hasil sebenarnya | Kesimpulan |
|---|---|---|---|---|
| 1 | Klik tambah. Semua field diisi dengan data yang benar, klik simpan | Data tersimpan di dalam database | Data tersimpan di dalam database | |
| 2 | Klik tambah. Ada field yang kosong, klik simpan | Muncul window dialog peringatan. | Muncul window dialog peringatan | |