BAB IV

HASIL DAN PEMBAHASAN

4.1. Perancangan Sistem

Pada tahap perancangan sistem ini akan dijelaskan mengenai perancangan pada objek yang digunakan, perancangan arsitektur program yang akan dibuat, perancangan tampilan dan perancangan menu pada *form* yang dibutuhkan.

Tujuan dari perancangan sistem ini adalah untuk memberikan gambaran secara umum kepada *user* tentang sistem yang baru.

4.1.1. Tujuan Perancangan Sistem

Perancangan sistem merupakan suatu proses perencanaan dalam pembuatan sistem baru dan mengacu pada kebutuhan yang sudah dianalisis sebelumnya dengan tujuan untuk menghasilkan sistem baru yang sesuai dengan kebutuhan dan diharapkan dapat membantu proses pengelolaan rekam medis di Puskesmas Karangsembung menjadi lebih baik, efektif dan efisien.

4.1.2. Gambaran Umum Sistem Yang Diusulkan

Adapun gambaran umum dari sistem yang diusulkan adalah penulis akan merancang sebuah sistem terkomputerisasi dan otomatisasi yang sebelumnya tidak ada, sehingga proses pengelolaan rekam medis bisa dilakukan lebih mudah.

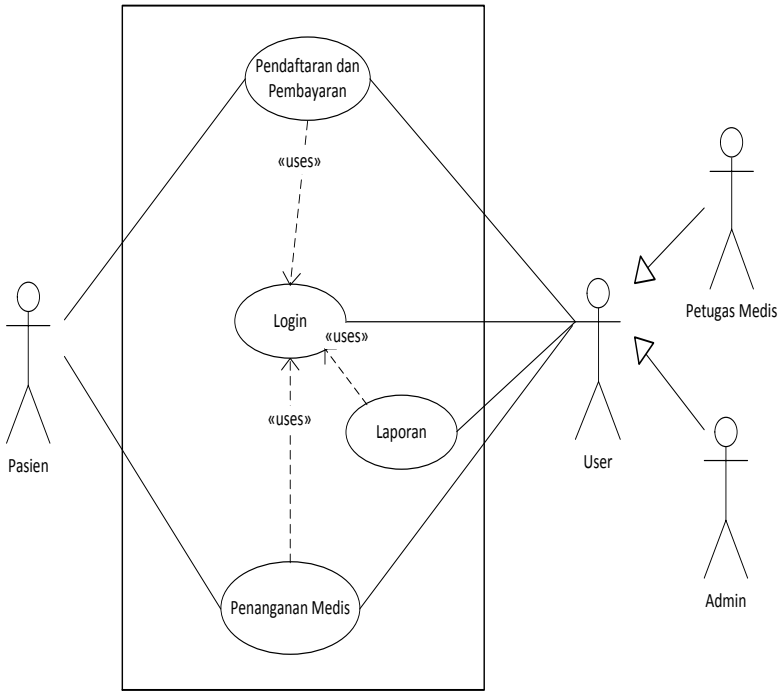
4.1.3. Perancangan Prosedur Yang Diusulkan

Perancangan prosedur merupakan awal dari pembuatan sistem yang akan dibuat. Sedangkan perancangan prosedur yang diusulkan merupakan tahap untuk

memperbaiki atau meningkatkan efisiensi kerja. Tahapan perancangan prosedur ini akan dijelaskan dengan menggunakan pemodelan sistem informasi berorientasi objek dengan UML *(Unified Modelling Language).*

4.1.3.1. Use Case Diagram

Berikut adalah gambar diagram *use case* yang diusulkan :



Gambar 4.1 *Use Case* Diagram Sistem Yang Diusulkan.

4.1.3.2. Skenario Use Case

Berikut adalah skenario *use case* yang diusulkan :

1. Nama Use Case : Login

   Aktor : User.

   Tujuan : Mengakses menu utama program.

Tabel 4.1 Tabel Skenario *Use Case Login*

No	User	Sistem
1	Memasukan Username dan Password	
2		Validasi Username dan Password
3		Menampilkan pesan login tidak valid
4	Memasukan kembali Username dan Password	
5		Validasi Username dan Password
6		Masuk ke menu utama

2. Nama Use Case : Pendaftaran

   Aktor : Pasien, User.

   Tujuan : Melakukan Pendaftaran

Tabel 4.2 Tabel Skenario *Use Case* Pendaftaran

No	Aktor	Sistem
1	Melakukan pendaftaran	
2	Input data pasien	
3		Menyimpan data pendaftaran pasien

4		Mencetak kartu pasien
5		Mencetak no. antrian

3. Nama Use Case : Penanganan Medis

   Aktor : Pasien, User.

   Tujuan : Melakukan penanganan medis.

Tabel 4.3 Tabel Skenario *Use Case* Penanganan Medis

No	Aktor	Sistem
1	Pasien menyampaikan keluhan	
2	User melakukan diagnosa dan penanganan medis	
3	UserMenginput hasil diagnosa	
4		Menyimpan data.

4. Nama Use Case : Bayar

   Aktor : Pasien.

   Tujuan : Membayar biaya pendaftaran

Tabel 4.4 Tabel Skenario *Use Case* Bayar

No	Aktor	Sistem
1	Membayar biaya pendaftaran	

2		Menyimpan data pembayaran
3		Mencetak bukti pembayaran

5. Nama Use Case : Laporan

   Aktor : Admin, Petugas Medis.

   Tujuan : Mencetak laporan.

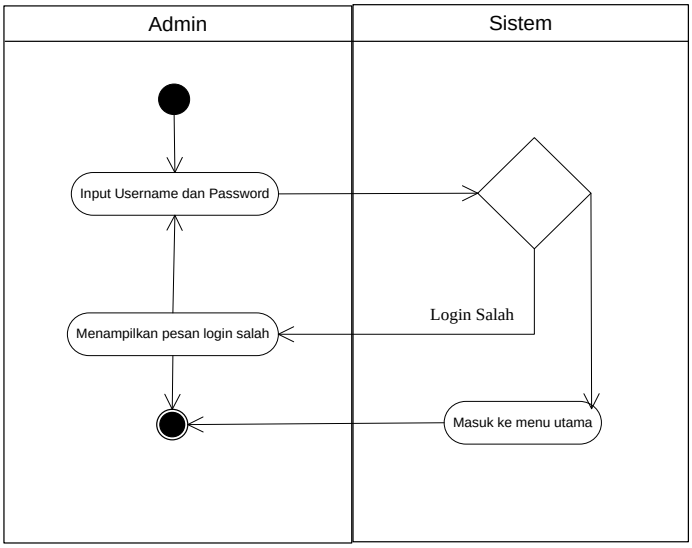
Tabel 4.5 Tabel Skenario *Use Case* Laporan

No	Aktor	Sistem
1	Memilih jenis laporan yang akan dibuat	
2		Menampilkan form dari jenis yang dipilih
3	Mengisi laporan sesuai dengan jenis yang dipilih	
4		Mencetak laporan

4.1.3.3. Activity Diagram

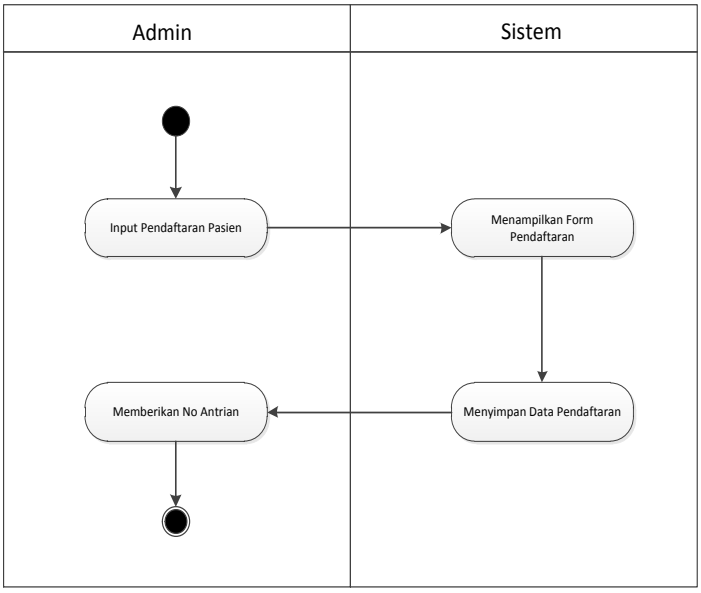
Berikut adalah *activity diagram* yang diusulkan :

1. Activity Diagram Login



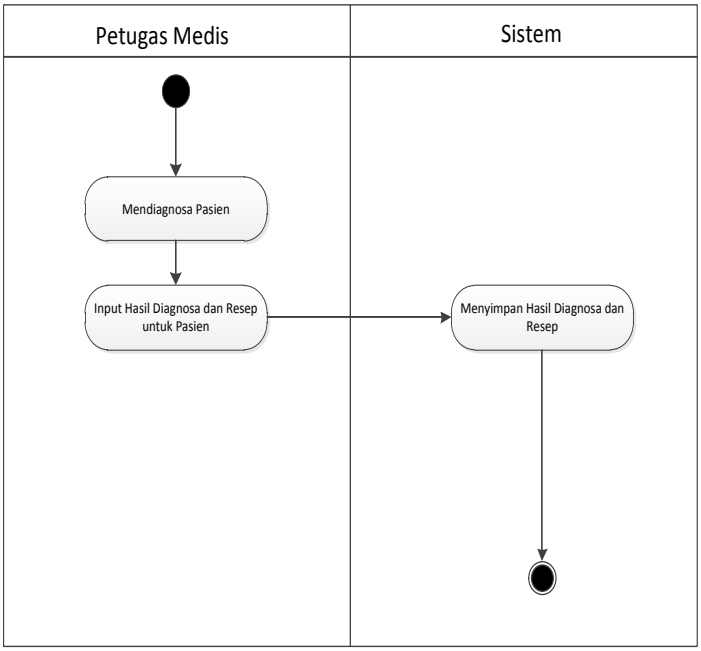
Gambar 4.2 *Activity Diagram Login*.

2. Activity Diagram Pendaftaran



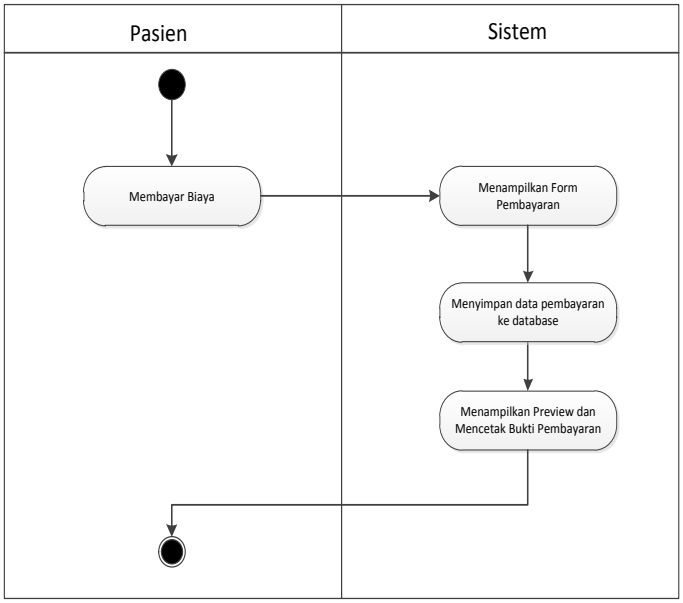
Gambar 4.3 *Activity Diagram* Pendaftaran.

3. Activity Diagram Penanganan Medis



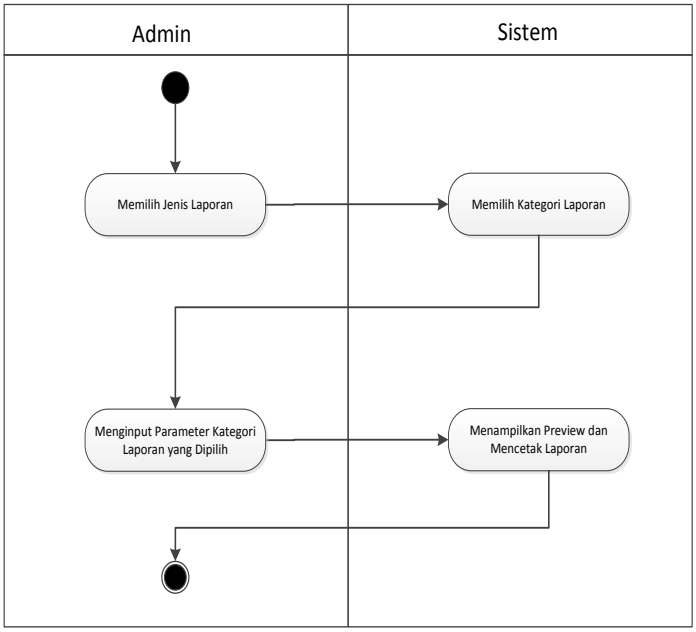
Gambar 4.4 *Activity Diagram* Penanganan Medis.

4. Activity Diagram Bayar



Gambar 4.5 *Activity Diagram* Bayar.

5. Activity Diagram Laporan

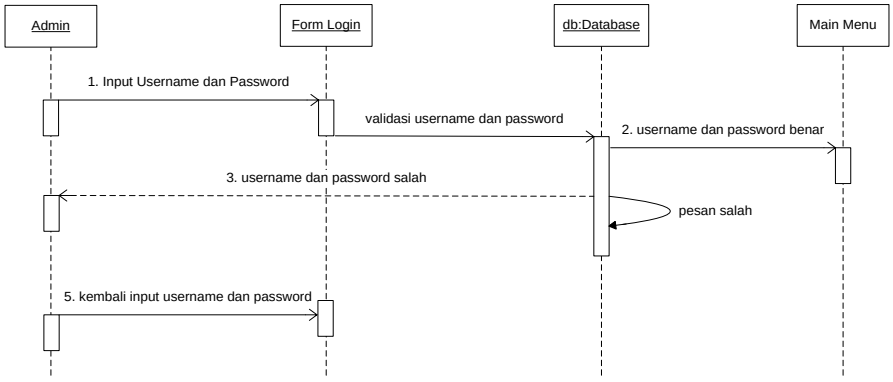


Gambar 4.6 *Activity Diagram* Laporan.

4.1.3.4. Sequence Diagram

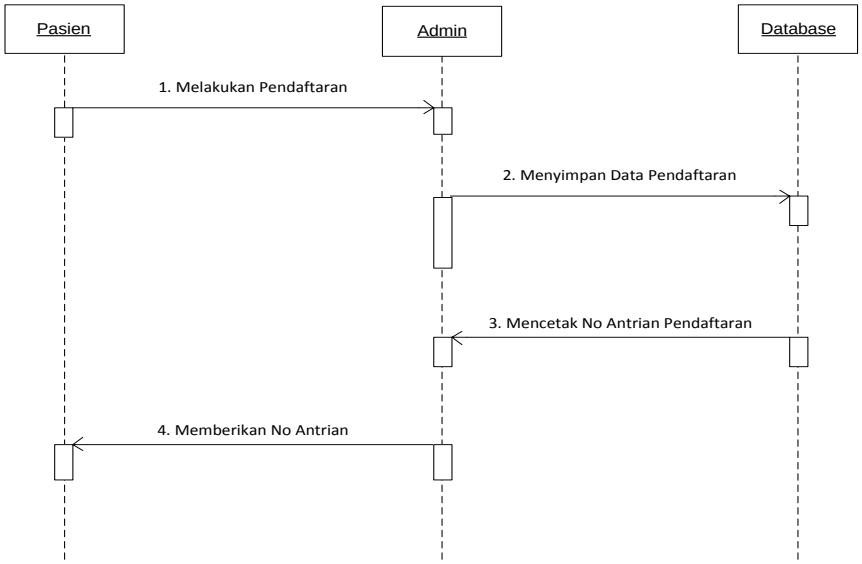
Berikut adalah *sequence diagram* yang diusulkan :

1. Sequence Diagram Login



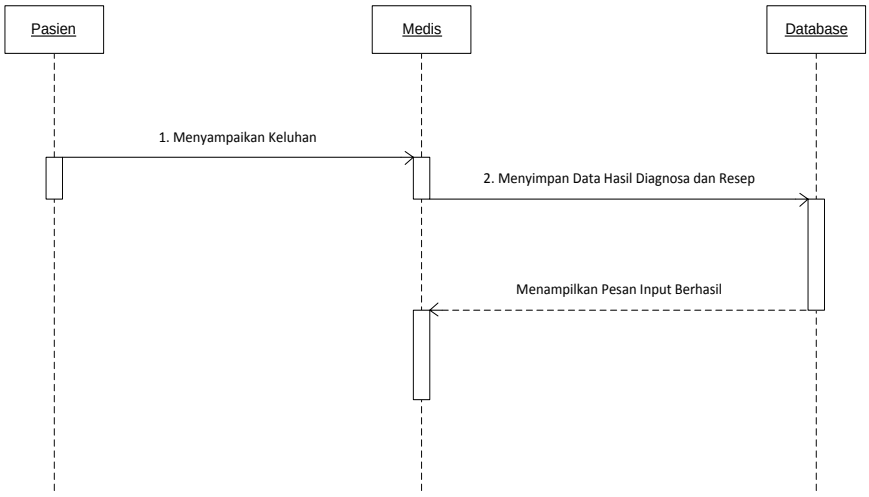
Gambar 4.7 *Sequence Diagram* Login.

2. Sequence Diagram Pendaftaran



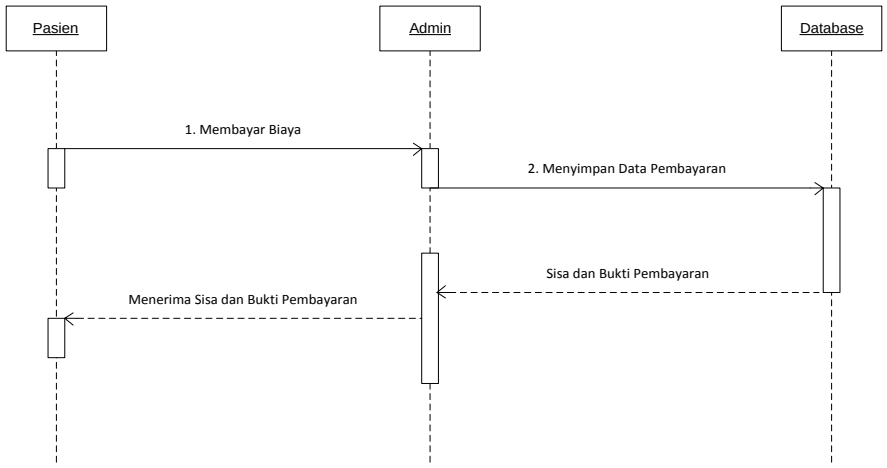
Gambar 4.8 *Sequence Diagram* Pendaftaran.

3. Sequence Diagram Penanganan Medis



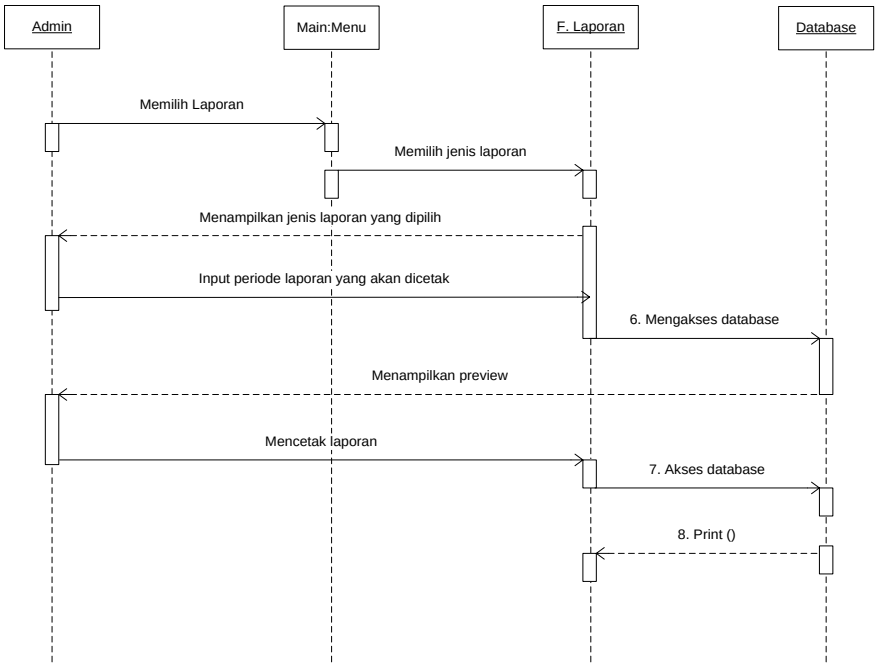
Gambar 4.9 *Sequence Diagram* Penanganan Medis.

4. Sequence Diagram Bayar



Gambar 4.10 *Sequence Diagram* Bayar.

5. Sequence Diagram Laporan

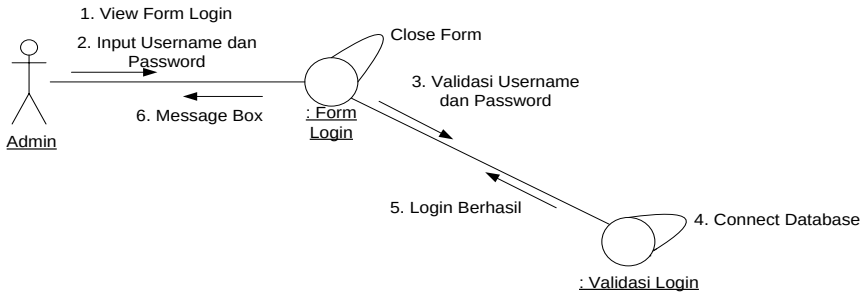


Gambar 4.11 *Sequence Diagram* Laporan.

4.1.3.5. Collaboration Diagram

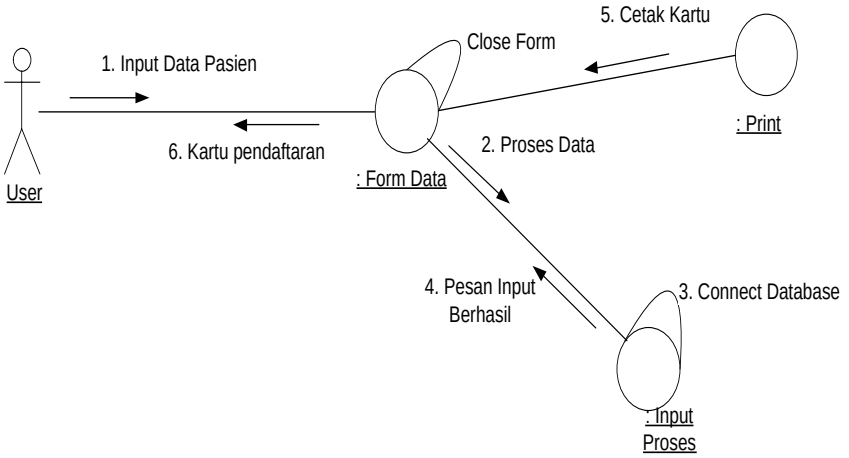
Berikut adalah *collaboration diagram* dari sistem yang diusulkan :

1. *Collaboration Diagram* Login



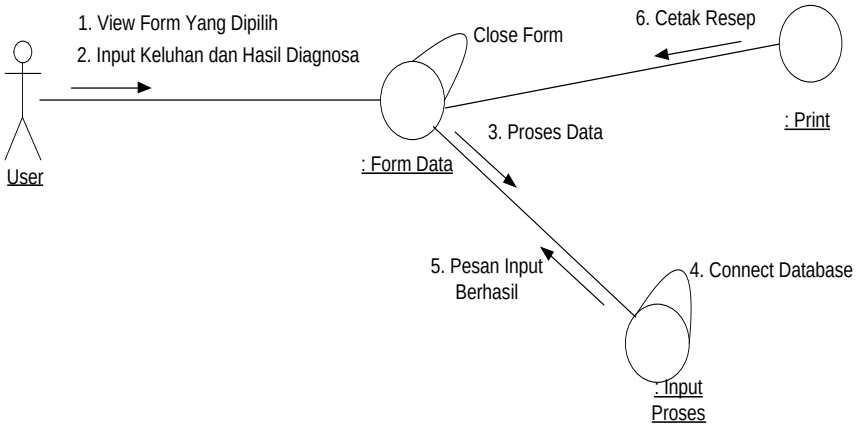
Gambar 4.12 *Collaboration Diagram Login*.

2. *Collaboration Diagram* Pendaftaran



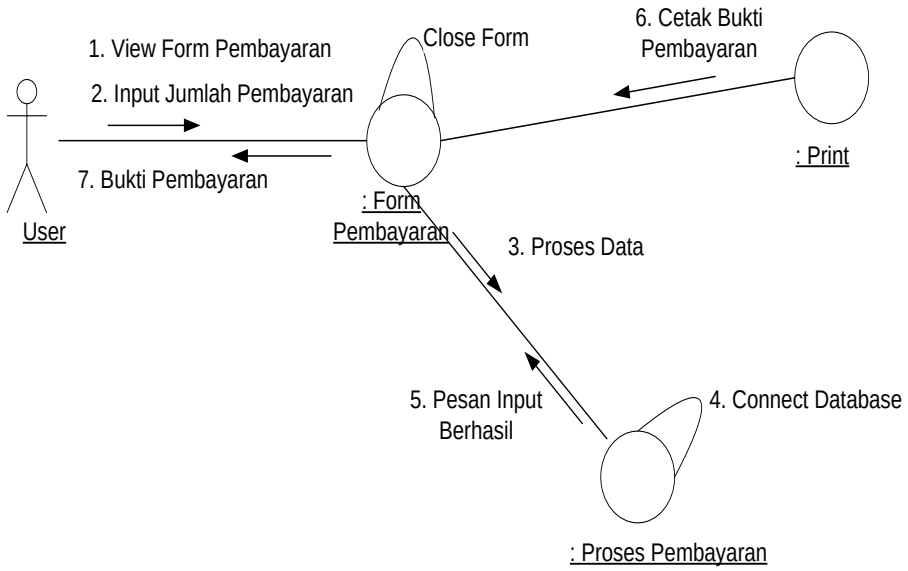
Gambar 4.13 *Collaboration Diagram* Pendaftaran.

3. *Collaboration Diagram* Penanganan Medis



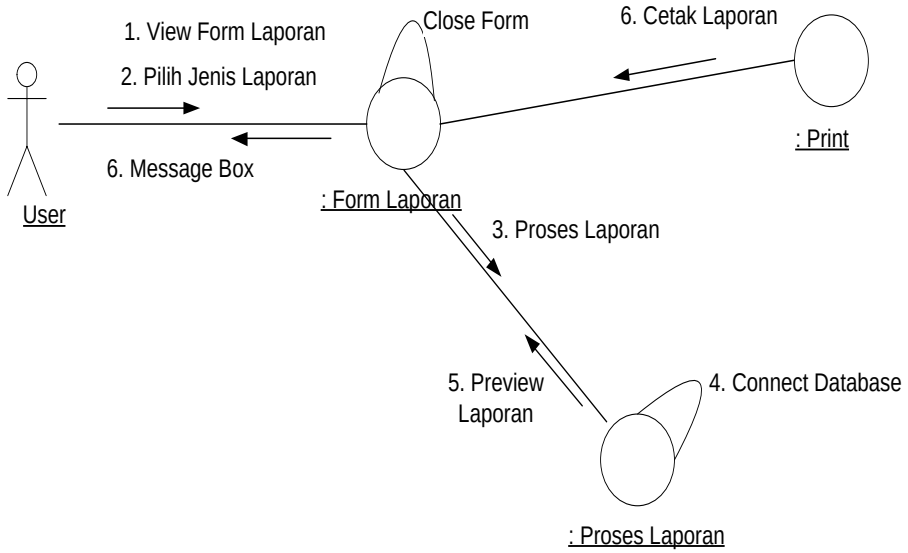
Gambar 4.14 *Collaboration Diagram* Penanganan Medis.

4. *Collaboration Diagram* Bayar



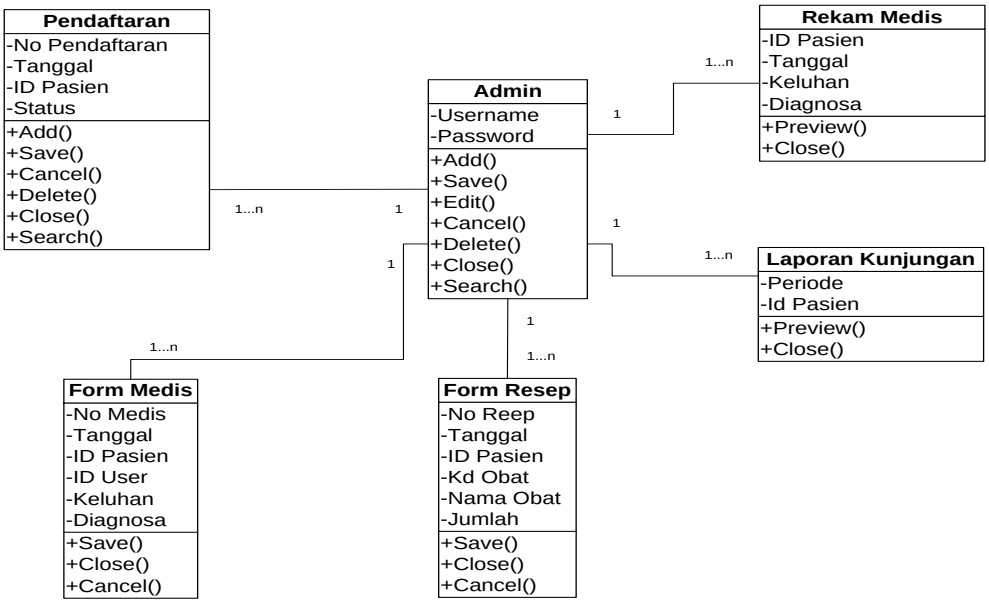
Gambar 4.15 *Collaboration Diagram* Bayar.

5. *Collaboration Diagram* Laporan



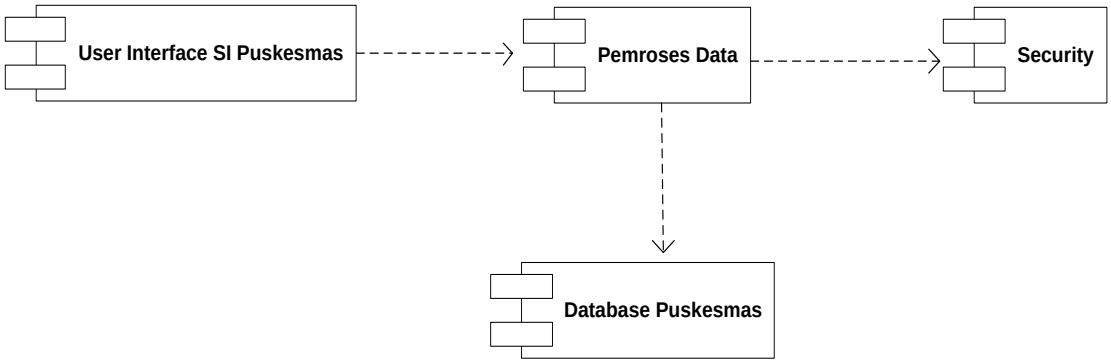
Gambar 4.16 *Collaboration Diagram* Laporan.

4.1.3.6. Class Diagram



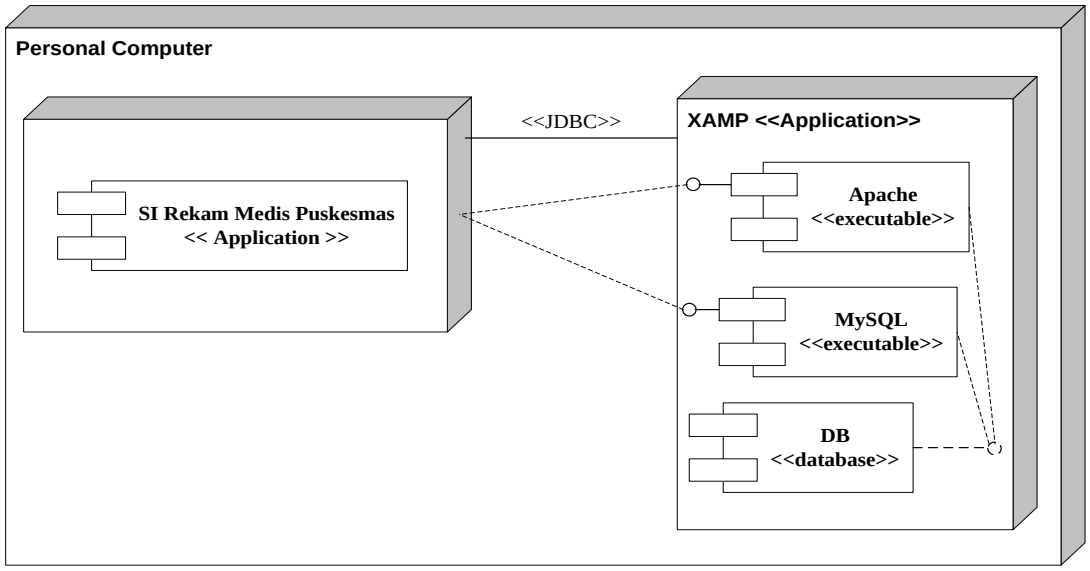
Gambar 4.17 *Class Diagram*.

4.1.3.7. Component Diagram



Gambar 4.18 *Component Diagram.*

4.1.3.8. Deployment Diagram

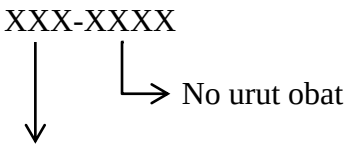


Gambar 4.19 *Deployment Diagram.*

4.1.3.9. Kodifikasi

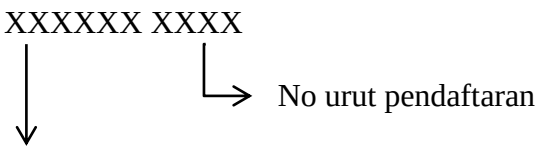
Kodifikasi atau pengkodean digunakan untuk menjabarkan item-item data yang bersifat unik. Dalam perancangan ini penulis melakukan pengkodean sebagai berikut :

1. Kode Obat



Contoh : OBB-0001

2. No Pendaftaran



Contoh : 1404230001

4.2. Perancangan Antar Muka

Perancangan antar muka (*interface*) merupakan suatu tahapan agar aplikasi yang dibuat lebih menarik dan tentunya lebih mudah digunakan. Tujuan dari perancangan antar muka pada adalah untuk mendapatkan suatu kriteria yang sangat penting dalam pengoprasian sebuah program aplikasi, yaitu aspek ramah dengan pengguna (*user friendly*). Menurut Ian Sommerville (2003:312), perancangan antar muka memiliki prinsip-prinsip sebagai berikut:

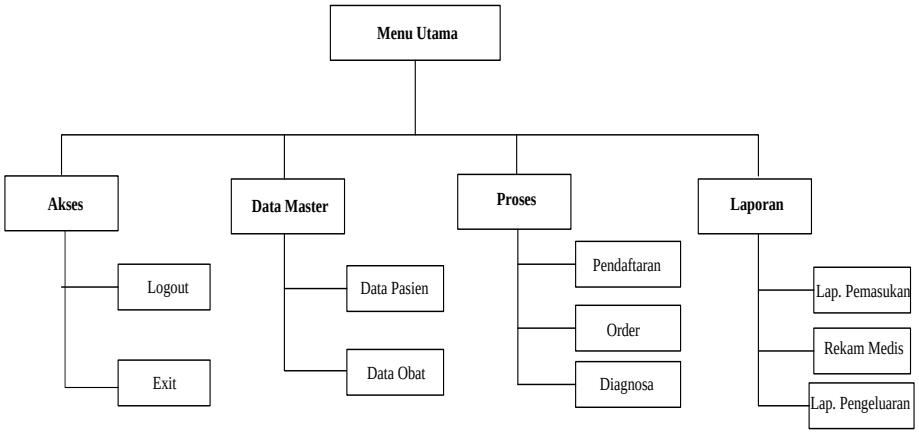
Tabel 4.6 Prinsip-Prinsip Perancangan Antar Muka

Prinsip	Keterangan
Kebiasaan User	Interface harus menggunakan istilah dan konsep yang diambil dari pengalaman orang-orang yang akan paling sering menggunakan sistem tersebut
Konsistensi	Inteface harus konsisten dalam hal, jika dimungkinkan, operasi-operasi yang sama harus diaktifkan dengan cara yang sama.
Kejutan Minimal	User harus tidak dikejutkan oleh perilaku sistem
Kemampuan Pemulihan (Recoverability)	*Interface* harus mencakup mekanisme untuk memungkinkan *user* pulih dari *error*
Panduan User	*Interface* harus menyediakan umpan balik yang berarti ketika terjadi kesalahan dan menyediakan fasilitas bantuan yang *context-sensitive*
Keragaman User	Interface harus menyediakan fasilitas interaksi yang sesuai untuk berbagai tife user sistem

[Sumber: Ian Summerville (2003:312)]

Perancangan antar muka yang ada dalam sistem berdasarkan diagram-diagram UML yang telah dibuat sebelumnya. Adapun perancangan antar muka ini terdiri dari struktur menu, perancangan *input* dan perancangan *output*.

4.2.1. Struktur Menu

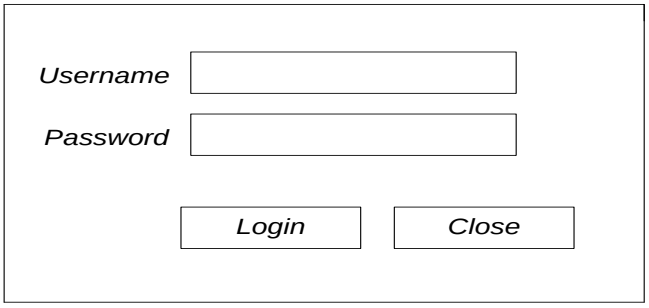


Gambar 4.20 Struktur Menu Utama Program

4.2.2. Perancangan Input

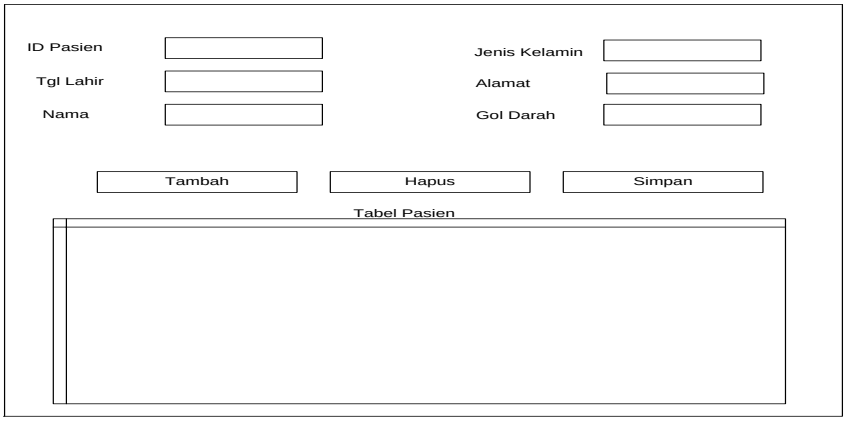
Perancangan *input* merupakan *form interface* tentang sistem masukan suatu proses dari pengguna sistem yang nantinya akan dijadikan sebagai laporan untuk admin. Adapun perancangan *input* sistem informasi ini adalah sebagai berikut:

1. Form Login



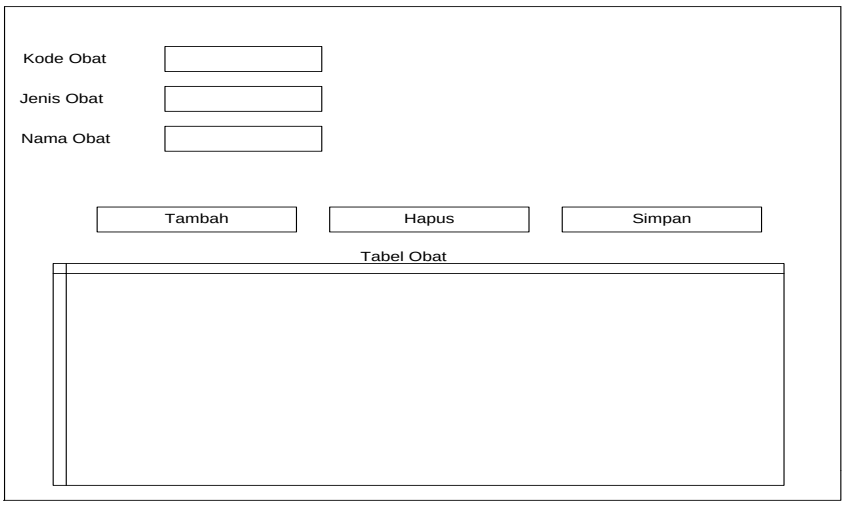
Gambar 4.21 Perancangan *Form Login*.

2. Form Data Pasien



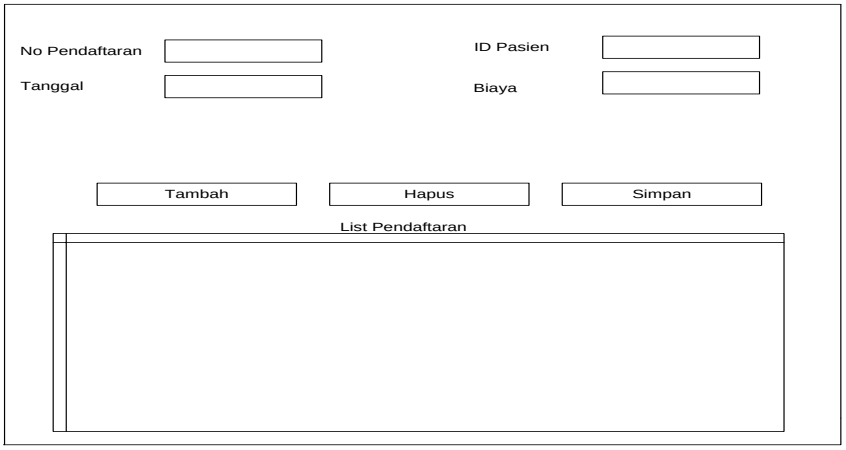
Gambar 4.22 Perancangan Data Pasien.

3. Form Data Obat



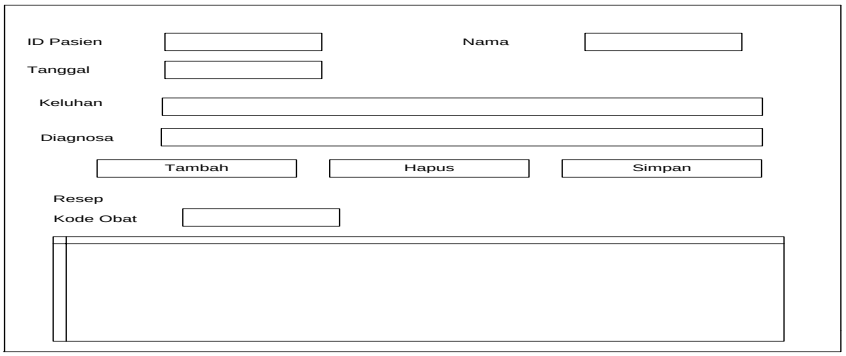
Gambar 4.23 Perancangan Data Obat.

4. Form Pendaftaran



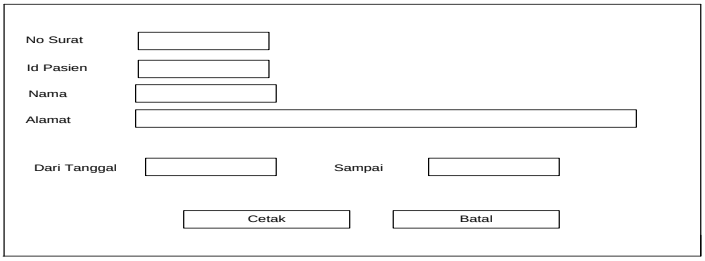
Gambar 4.24 Perancangan *Form* Pendaftaran.

5. Form Diagnosa



Gambar 4.25 Perancangan Form Diagnosa.

6. Form Surat Sakit



Gambar 4.26 Perancangan Form Surat Sakit.

4.2.3. Perancangan Output

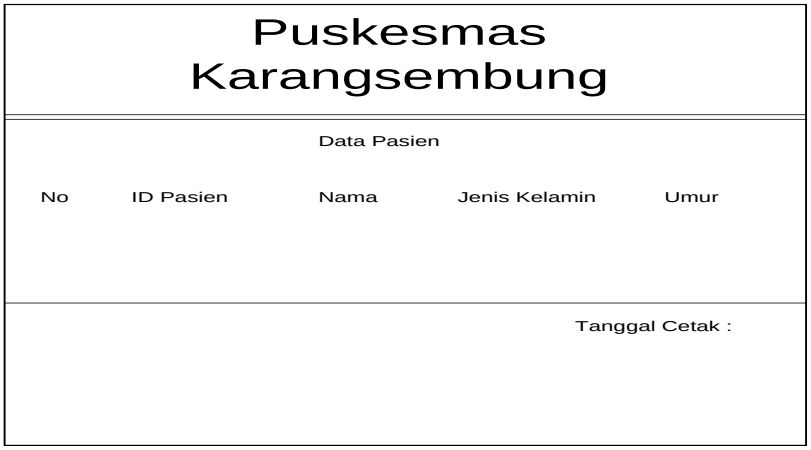
Perancangan *output* adalah informasi yang dihasilkan oleh sistem berupa laporan hasil dari proses *input*-an yang diterima oleh sistem. Adapun perancangan *output* dari sistem informasi ini adalah sebagai berikut:

1. Laporan Rekam Medis



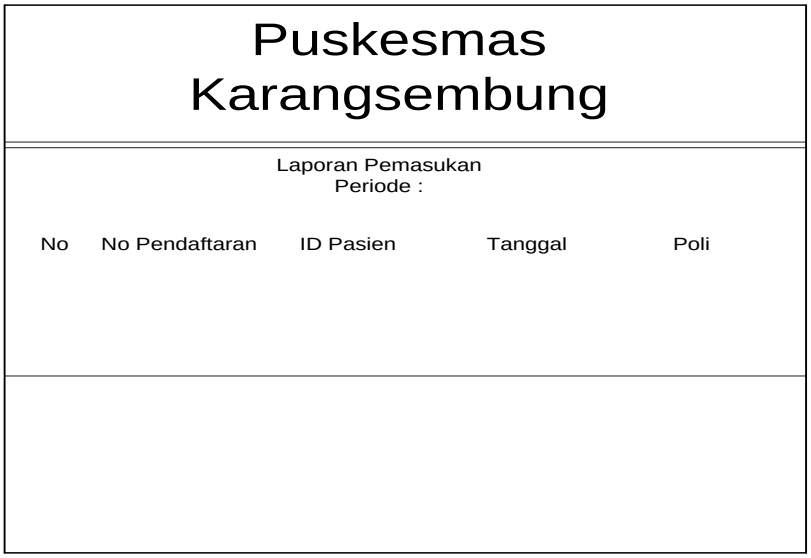
Gambar 4.27 Tampilan Laporan Rekam Medis.

2. Laporan Data Pasien



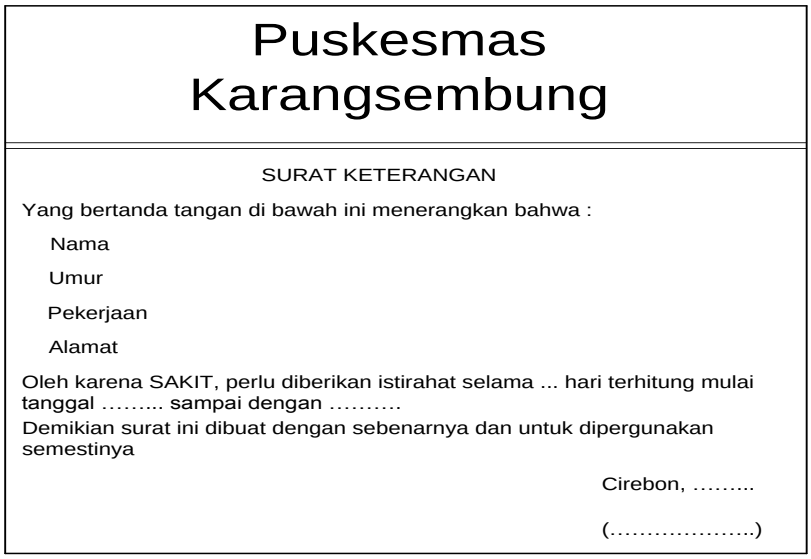
Gambar 4.28 Tampilan Laporan Data Pasien.

3. Laporan Kunjungan



Gambar 4.29 Tampilan Laporan Kunjungan.

4. Surat Sakit

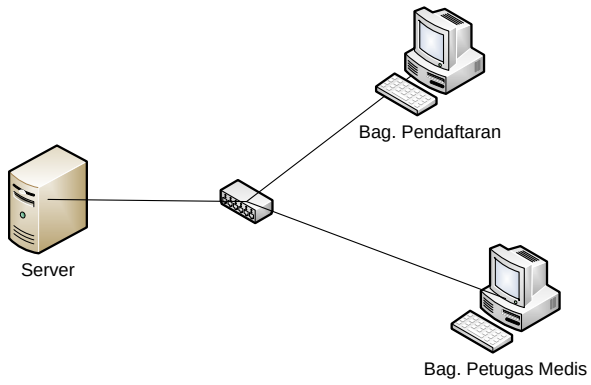


Gambar 4.30 Tampilan Surat Sakit.

4.3. Perancangan Arsitektur Jaringan

Perancangan arsitektur jaringan adalah bentuk umum dari suatu rancangan program untuk memudahkan pemakai dalam menjalankan program komputer secara terintegrasi. Pada perancangan arsitektur jaringan harus disesuaikan dengan fungsi bagi penggunanya agar tidak mengalami kesulitan dalam memilih menu-menu yang diinginkan.

Adapun tipe jaringan yang digunakan adalah tipe jaringan LAN *(local area network*), yakni tipe jaringan untuk area kecil yang cocok digunakan dalam perancangan sistem informasi puskesmas Karangsembung. Dan model hubungan yang digunakan adalah *client server*, dimana model *client server* ini memungkinkan jaringan untuk mensentralisasi fungsi dan aplikasi kepada satu atau dua *dedicated file server*. Sebuah *file server* menjadi jantung dari keseluruhan sistem yang bisa mengakses sumber daya dan menyediakan keamanan. Model hubungan ini menyediakan mekanisme untuk mengintegrasikan seluruh komponen yang ada di jaringan dan memungkinkan banyak pengguna secara bersama-sama memakai sumber daya pada *file server*.



Gambar 4.31. Perancangan Arsitektur Jaringan.

[Sumber : Budhi Irawan [2005:30]]

4.4. Implementasi

Tahap Selanjutnya dalam membangun sebuah perangkat lunak adalah implementasi. Implementasi merupakan kumpulan dari elemen-elemen yang telah didesain ke dalam bentuk pemrograman untuk menghasilkan suatu tujuan yang dibuat berdasarkan kebutuhan yang telah dianalisis sebelumnya. Langkah-langkah dalam tahap implementasi ini adalah urutan kegiatan awal sampai akhir yang harus dilakukan untuk mewujudkan sistem yang telah dirancang.

4.4.1. Batasan Implementasi

Dalam mengimplementasikan perangkat lunak Sistem Informasi Pengolahan Data Pasien ini terdapat beberapa hal yang menjadi batasan implementasi, yaitu :

1. Database yang di gunakan dalam implementasi ini adalah MySQL
2. Sistem Informasi ini tidak dapat memenuhi semua kebutuhan pengolahan data yang ada di Puskesmas Karangsembung Cirebon, melainkan hanya pada sub system yang meliputi pembahasan pokok saja.

4.4.2. Implementasi Perangkat Lunak

Perangkat lunak yang dibutuhkan dan digunakan dalam pembuatan aplikasi sistem informasi ini adalah sebagai berikut:

1. Windows 7 (*seven)*, sebagai sistem operasi yang dipakai oleh penulis.
2. Netbeans IDE 7.4, sebagai *software* yang penulis gunakan dalam membangun sistem informasi pengelolaan data tamu dan pengunjung ini.

3. iReport 4.7.1, sebagai aplikasi untuk pembuatan laporan.
4. MySQL, sebagai pengembang basis data dan perangkat lunak lainnya menggunakan XAMPP.

4.4.3. Implementasi Perangkat Keras

Adapun perangkat keras yang digunakan dalam pembuatan aplikasi sistem informasi ini adalah sebagai berikut :

1. Proccesor : Intel(R) Core(TM) i3 CPU 540 @ 3.10GHz.
2. Memory : 4096MB RAM.
3. Printer : Standar.
4. Hard Disk : 500GB
5. Mouse dan Keyboard.

4.4.4. Implementasi Basis Data

```
-- phpMyAdmin SQL Dump
-- version 3.5.2.2
-- http://www.phpmyadmin.net
--
-- Host: 127.0.0.1
-- Generation Time: May 04, 2014 at 12:26 AM
-- Server version: 5.5.27
-- PHP Version: 5.4.7
--
-- Database: `puskesmas`
```

```
--

-- --------------------------------------------------------

--
-- Table structure for table `daftar`
--

CREATE TABLE IF NOT EXISTS `daftar` (
  `no_daftar` varchar(15) NOT NULL,
  `tanggal` date NOT NULL,
  `id_pasien` varchar(15) NOT NULL,
  `kd_user` varchar(15) NOT NULL,
  `biaya` double NOT NULL,
  `status` int(1) NOT NULL,
  PRIMARY KEY (`no_daftar`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `dt_medis`
--
```

```
CREATE TABLE IF NOT EXISTS `dt_medis` (
  `no_medis` varchar(15) NOT NULL,
  `keluhan` text NOT NULL,
  `diagnosa` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `dt_orders`
--

CREATE TABLE IF NOT EXISTS `dt_orders` (
  `noPO` varchar(15) NOT NULL,
  `kd_obat` varchar(15) NOT NULL,
  `jns_obat` varchar(100) NOT NULL,
  `nm_obat` varchar(300) NOT NULL,
  `qty` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------
```

```
--
-- Table structure for table `dt_resep`
--

CREATE TABLE IF NOT EXISTS `dt_resep` (
  `no_resep` varchar(15) NOT NULL,
  `kd_obat` varchar(15) NOT NULL,
  `nm_obat` varchar(300) NOT NULL,
  `qty` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


-- --------------------------------------------------------

--
-- Table structure for table `income`
--

CREATE TABLE IF NOT EXISTS `income` (
  `no_resep` varchar(15) NOT NULL,
  `no_daftar` varchar(15) NOT NULL,
  `tanggal` date NOT NULL,
  `total` double NOT NULL,
```

```
  `bayar` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `login`
--

CREATE TABLE IF NOT EXISTS `login` (
  `kd_user` varchar(15) NOT NULL,
  `nama` varchar(200) NOT NULL,
  `bagian` varchar(100) NOT NULL,
  `log` time NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `medis`
--
```

```
CREATE TABLE IF NOT EXISTS `medis` (
  `no_medis` varchar(15) NOT NULL,
  `no_daftar` varchar(15) NOT NULL,
  `tanggal` date NOT NULL,
  `kd_user` varchar(15) NOT NULL,
  `id_pasien` varchar(15) NOT NULL,
  PRIMARY KEY (`no_medis`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `obat`
--

CREATE TABLE IF NOT EXISTS `obat` (
  `kd_obat` varchar(15) NOT NULL,
  `jns_obat` varchar(100) NOT NULL,
  `nm_obat` varchar(300) NOT NULL,
  `stok` int(11) NOT NULL,
  `stok_min` int(11) NOT NULL,
  `harga` double NOT NULL,
  `hrg_jual` double NOT NULL,
```

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

-- --------------------------------------------------------

--
-- Table structure for table `orders`
--

CREATE TABLE IF NOT EXISTS `orders` (
  `noPO` varchar(15) NOT NULL,
  `tanggal` date NOT NULL,
  `supplier` varchar(250) NOT NULL,
  `kd_user` varchar(15) NOT NULL,
  `status` int(1) NOT NULL COMMENT '0=Belum, 1=Sudah',
  PRIMARY KEY (`noPO`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `outcome`
--
```

```
CREATE TABLE IF NOT EXISTS `outcome` (
  `noPO` varchar(15) NOT NULL,
  `tanggal` date NOT NULL,
  `total` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `pasien`
--

CREATE TABLE IF NOT EXISTS `pasien` (
  `id_pasien` varchar(18) NOT NULL,
  `nama` varchar(100) NOT NULL,
  `jk` text NOT NULL,
  `asal` text NOT NULL,
  `tgl_lahir` date NOT NULL,
  `pekerjaan` text NOT NULL,
  `agama` text NOT NULL,
  `gol_darah` varchar(2) NOT NULL,
  `alamat` varchar(300) NOT NULL,
```

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

-- --------------------------------------------------------

--
-- Table structure for table `rekam_medis`
--

CREATE TABLE IF NOT EXISTS `rekam_medis` (
  `id_pasien` varchar(15) NOT NULL,
  `no_medis` varchar(15) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `resep`
--

CREATE TABLE IF NOT EXISTS `resep` (
  `no_resep` varchar(15) NOT NULL,
  `tanggal` date NOT NULL,
```

```
  `no_medis` varchar(15) NOT NULL,
  `id_pasien` varchar(15) NOT NULL,
  `status` int(1) NOT NULL,
  PRIMARY KEY (`no_resep`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------

--
-- Table structure for table `supplier`
--

CREATE TABLE IF NOT EXISTS `supplier` (
  `KodeSupplier` varchar(8) NOT NULL,
  `NamaSupplier` varchar(250) NOT NULL,
  `NoTelp` varchar(15) NOT NULL,
  `Alamat` varchar(300) NOT NULL,
  PRIMARY KEY (`KodeSupplier`),
  UNIQUE KEY `NoTelp` (`Alamat`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- --------------------------------------------------------
```

```
--
-- Table structure for table `user`
--

CREATE TABLE IF NOT EXISTS `user` (
  `kd_user` varchar(15) NOT NULL,
  `nama` varchar(100) NOT NULL,
  `username` varchar(50) NOT NULL,
  `password` varchar(15) NOT NULL,
  `alamat` varchar(200) NOT NULL,
  `telpon` varchar(15) NOT NULL,
  `Posisi` varchar(50) NOT NULL,
  PRIMARY KEY (`kd_user`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
```

4.4.5. Implementasi Antar Muka

Tahap implemantasi antar muka yang dilakukan adalah dengan menggunakan sebuah *file project* yang berekstensi .java yang ada dalam program Netbeans, yang di dalamnya terdapat *form-form* untuk pembuatan aplikasi.

1. Implementasi Login

Login merupakan hal penting agar dapat masuk ke dalam menu utama aplikasi pengelolaan data pasien dan data transaksi. Fasilitas *login* ini diperuntukan bagi petugas medis dan admin.

Dalam melakukan *login*, *input*-an *password* dan *username* harus benar, apabila salah, maka admin tidak bisa mengakses aplikasi ini.

Tabel 4.6 Implementasi Login

Sub Menu	Deskripsi	Nama File
Menu Login	File program yang form login	FLogin.java
Login	Untuk login admin (petugas)	
Batal	Untuk batal melakukan login	
Keluar	Untuk keluar dari form login	

2. Imlpementasi Halaman Utama

Implementasi halaman utama dari aplikasi ini adalah sebagai berikut :

Tabel 4.7 Halaman Utama

Sub Menu	Deskripsi	Nama File
Menu Utamam	File program yang form login	FHome.java
Akses	Untuk menampilkan sub menu hak akses.	
Master	Untuk menampilkan sub menu data master.	
Proses	Untuk menampilkan sub menu proses.	
Laporan	Menampilkan form laporan	FLaporan.java

3. Implementasi Halaman Berdasarkan Sub Menu Akses

Implementasi sub-menu akses dari aplikasi ini adalah sebagai berikut :

Tabel 4.8 Implementasi Sub Menu Akses

Sub Menu	Deskripsi	Nama File
Logout	Untuk logout *user*	
Keluar	Untuk keluar dari aplikasi	

4. Implementasi Halaman Berdasarkan Sub Menu Master

Implementasi sub-menu master dari aplikasi ini adalah sebagai berikut :

Tabel 4.9 Implementasi Sub Menu Master

Sub Menu	Deskripsi	Nama File
Obat	Untuk mengolah data obat	FObat.java
Pasien	Untuk mengolah data pasien	FPasien.java

5. Implementasi Halaman Berdasarkan Sub Menu Proses

Implementasi sub-menu proses dari aplikasi ini adalah sebagai berikut :

Tabel 4.10 Implementasi Sub Menu Proses

Sub Menu	Deskripsi	Nama File
Pendaftaran	Untuk input data pendaftaran	FDaftar.java
Diagnosa	Untuk input data diagnosa	FMedis.java

6. Implementasi Halaman Berdasarkan Sub Menu Laporan

Implementasi sub-menu laporan dari aplikasi ini adalah sebagai berikut :

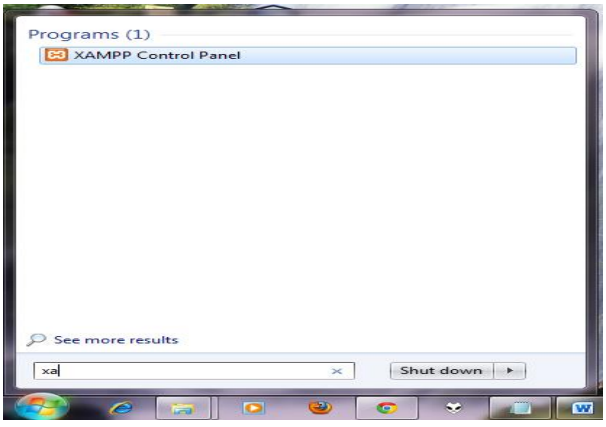
Tabel 4.11 Implementasi Sub Menu Laporan

Sub Menu	Deskripsi	Nama File
Menu Rekam Medis	Untuk mencetak laporan rekam medis	FLaporan.java
Menu Kunjungan	Untuk mencetak laporan kunjungan	FLaporan.java

4.4.6. Implementasi Instalasi Program

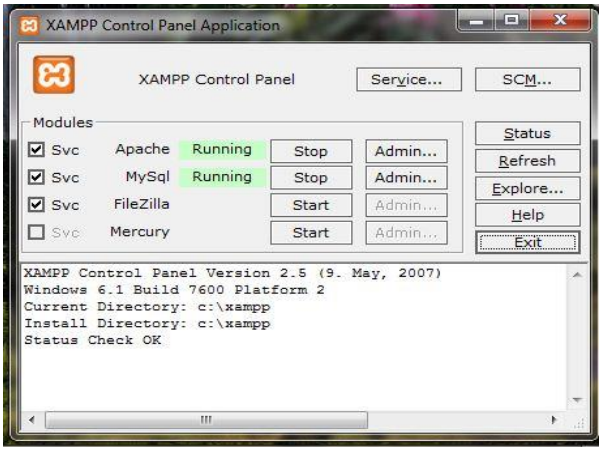
Pengimplementasian sistem ini bertujuan untuk menerangkan secara singkat penggunaan program aplikasi sistem informasi Puskesmas Karangsembung Cirebon. Berikut adalah tahapan singkat tahap instalasi program yaitu:

1. Klik menu start kemudian cari xampp – XAMPP Control Panel



Gambar 4.32 Menjalankan XAMPP Control Panel

2. Setelah di klik, XAMPP akan otomatis aktif, Apache dan MySQL



Gambar 4.33 XAMPP Control Panel Aktif

4.4.7. Penggunaan Program

Kegiatan ini bertujuan untuk menerangkan secara singkat mengenai penggunaan program sistem informasi ini. Adapun tampilan dari program tersebut adalah sebagai berikut:

1. Tampilan Form Login

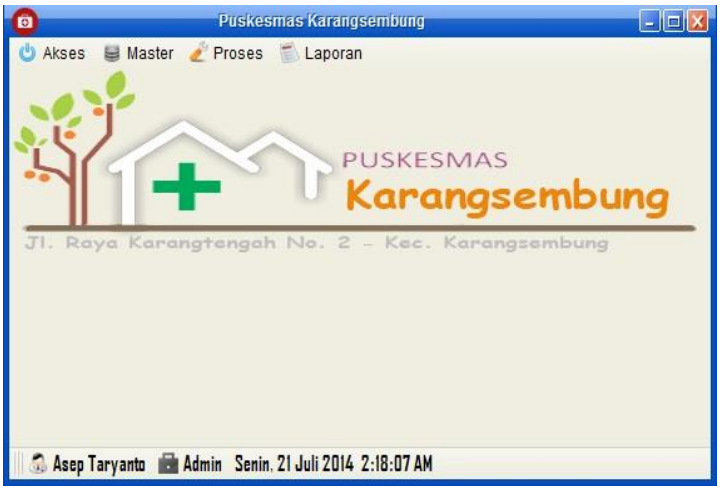
Form login ini berfungsi sebagai keamanan sistem. Untuk melakukan login, isi password dan username dan klik tombol login. Jika username dan login benar, maka form utama akan bisa diakses.



Gambar 4.34. Form Login.

2. Tampilan Menu Utama

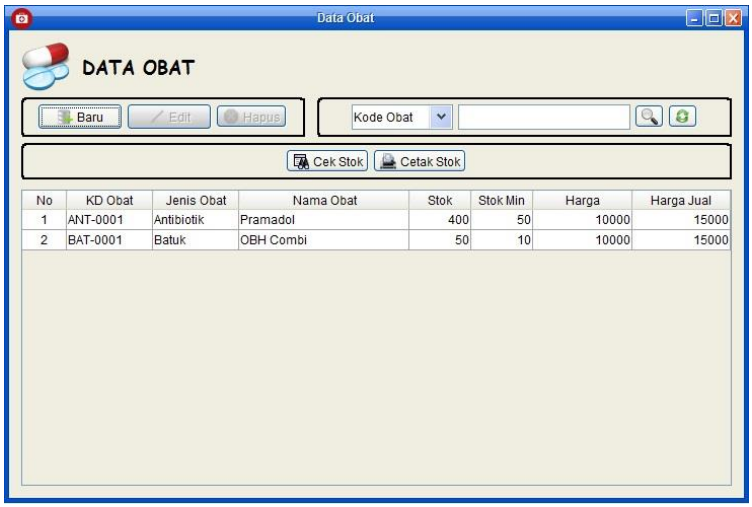
Form menu utama ini berfungsi untuk meng-akses sub-sub menu yang ada pada aplikasi ini. Untuk memilih sub-sub menu yang ada pada aplikasi ini, klik pada nama menu yang diinginkan.



Gambar 4.35. Tampilan Menu Utama.

3. Tampilan Data Obat

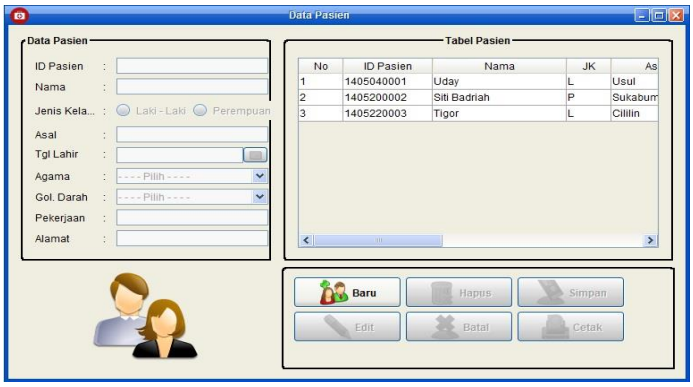
Form ini berfungsi untuk melihat, meng-input, dan mencari data obat.



Gambar 4.36. Tampilan Data Obat.

4. Tampilan Data Pasien

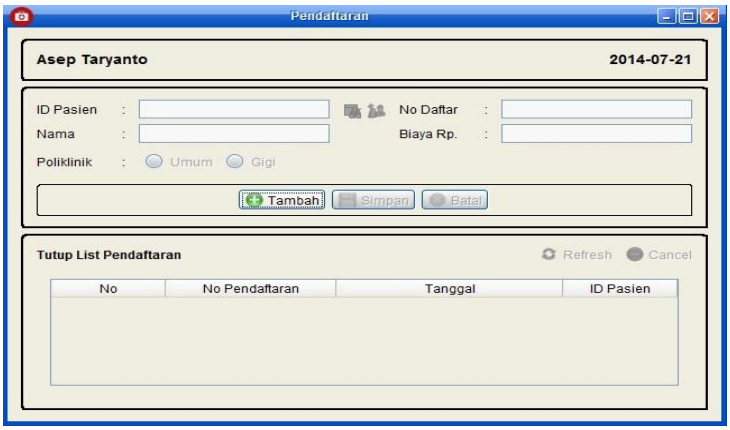
Form ini berfungsi untuk melihat, meng-input, dan merubah data pasien.



Gambar 4.37. Tampilan Data Pasien.

5. Tampilan Form Pendaftaran

Form ini berfungsi untuk memasukan data pendaftaran kunjungan pasien.



Gambar 4.38. Tampilan Form Pendaftaran.

6. Tampilan Form Surat Sakit

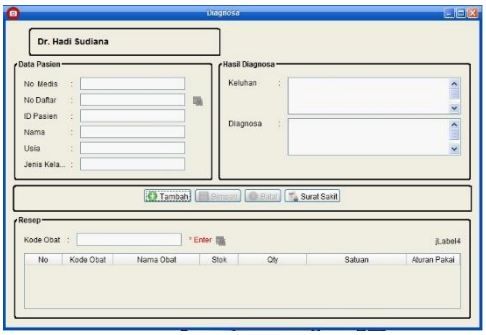
Form ini berfungsi untuk membuat surat sakit.



Gambar 4.39. Tampilan Form Surat Sakit.

7. Tampilan Form Diagnosa

Form ini berfungsi untuk input data diagnosa.



Gambar 4.40. Tampilan Form Diagnosa.

8. Tampilan Rekam Medis.



Gambar 4.41. Tampilan Rekam Medis.

9. Tampilan Surat Sakit.



Gambar 4.42. Tampilan Surat Sakit.

4.5. Pengujian

Adapun teknik pengujian perangkat lunak yang digunakan penulis dalam membangun sistem informasi di pangkas rambut mahkota ini, adalah teknik pengujian *blackbox testing.* Menurut Al-Bahra bin Ladjamudin (2006:379), metode *blackbox testing* ini berfokus pada keperluan fungsional dari *software*, yang memungkinkan untuk membuat himpunan kondisi *input* yang akan melatih seluruh syarat-syarat fungsional suatu program. Uji coba *blackbox* merupakan pendekatan yang melengkapi untuk menemukan kesalahan yang terjadi.

4.5.1. Rencana Pengujian

Berikut ini adalah rencana pengujian sistem informasi ini:

Tabel 4.12. Rencana Pengujian SI Puskesmas Karangsembung

Kelas Uji	**Butir Uji**	**Tingkat Pengujian**	**Jenis Pengujian**
Login	Pengecekan Login	Sistem	Blackbox
Pengisian Data	Pengisian Data Pendaftaran	Modul	Blackbox
	Pengisian Data Diagnosa	Modul	Blackbox
	Pencarian Jumlah Kunjungan	Modul	Blackbox
	Pencarian Data Rekam Medis	Modul	Blackbox

4.5.2. Kasus dan Hasil Pengujian

Setelah melakukan rencana pengujian, maka hal selanjutnya yang perlu dilakukan adalah melakukan studi kasus dan bagaimana hasil dari pengujian

studi kasus tersebut. Adapun, kasus dan hasil pengujian yang dilakukan adalah sebagai berikut :

1. Pengecekan Login

Berikut ini adalah kasus dan hasil pengujian login:

Tabel 4.13. Kasus dan Hasil Pengujian Pengecekan Login Admin

Kasus dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Username : admin Password : admin	Tercantum pada textbox username dan textbox password.	Mengisi login sesuai dengan yang diharapkan.	[X] Diterima [ ] Ditolak
Klik tombol login	Menampilkan pesan “login berhasil” dan dapat mengakses sub menu yang ada di menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Username :admon Password : admon	Tercantum pada textbox username dan textbox password.	Kasir gagal melakukan login	[X] Diterima [ ] Ditolak

Klik Tombol Login	Menampilkan pesan “ada kesalahan pada username dan password” dan user tidak dapat masuk dan mengakses ke dalam menu utama	Tombol dapat berfungsi sesuai dengan yang diharapkan	[X] Diterima [ ] Ditolak

2. Pengisian Data pendaftaran

Berikut ini adalah kasus dan hasil pengujian pengisian data pendaftaran:

Tabel 4.14. Kasus dan Hasil Pengujian Pengisian Data Pendaftaran.

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Klik tombol “Tambah”	No pendaftaran akan otomatis muncul	Dapat memasukan data pasien	[X] Diterima [ ] Ditolak
Data-data pengambilan diisi secara lengkap	Penyimpanan data ke dalam file data pemesanan	Dapat melakukan penyimpanan data dengan benar	[X] Diterima [ ] Ditolak
Klik tombol simpan	Data tersimpan ke dalam database data pemesanan	Muncul pesan “Input berhasil”, dan data dapat	[X] Diterima [ ] Ditolak

		disimpan.	
Kasus dan Hasil Uji (Data Salah)			
Data-data pasien tidak diisi secara lengkap	Tidak dapat melakukan penyimpanan	Tidak dapat melakukan penyimpan data	[X] Diterima [ ] Ditolak
Klik tombol Simpan	Tidak dapat melakukan penyimpanan	Muncul pesan "Silahkan Lengkapi Data"	[X] Diterima [ ] Ditolak

3. Pengisian Data Diagnosa

Berikut ini adalah kasus dan hasil pengujian pengisian data diagnosa:

Tabel 4.15. Kasus dan Hasil Pengujian Pengisian Data Diagnosa.

Kasus dan Hasil Uji (Data Benar)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Pilih *No Pendaftaran* pada *combobox*	Tombol simpan dan batal aktif, serta data pasien akan otomatis muncul	Dapat melakukan pengisian data sesuai dengan yang diharapkan	[X] Diterima [ ] Ditolak
Data-data diagnosa diisi secara lengkap	Penyimpanan data ke dalam file data pemesanan	Dapat melakukan penyimpanan data dengan benar	[X] Diterima [ ] Ditolak

Klik tombol simpan	Data tersimpan ke dalam database data pemesanan	Muncul pesan “Input berhasil”, dan data dapat disimpan.	[X] Diterima [ ] Ditolak
Kasus dan Hasil Uji (Data Salah)			
Data-data diagnosa tidak diisi secara lengkap dan ada field yang masih kosong	Tidak dapat melakukan penyimpanan	Tidak dapat melakukan penyimpan data	[X] Diterima [ ] Ditolak
Klik tombol Simpan	Tidak dapat melakukan penyimpanan	Muncul pesan “Silahkan Lengkapi Data”	[X] Diterima [ ] Ditolak

4. Pencarian Jumlah Kunjungan

Berikut ini adalah kasus dan hasil pengujian pencarian jumlah kunjungan:

Tabel 4.16. Kasus dan Hasil Pengujian Pencarian Jumlah Kunjungan

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan periode yang	Pencarian data ke dalam file	Dapat melakukan pencarian data	[X] Diterima [ ] Ditolak

akan dicari	kunjungan	dan hasilnya ditampilkan pada tabel	
Klik tombol cari	Data bisa ditemukan	Muncul pesan “Data ditemukan”, dan muncul data yang dicari.	[X] Diterima [ ] Ditolak
Kasus Dan Hasil Uji (Data Salah)			
periode tidak diisi / salah	Tidak dapat melakukan pencarian	Pencarian data gagal dilakukan	Tidak dapat melakukan pencarian
Klik tombol cari	Data tidak ditemukan	Muncul pesan “Data tidak ditemukan	Tidak dapat melakukan pencarian

5. Pencarian Data Rekam Medis

Berikut ini adalah kasus dan hasil pengujian pencarian data rekam medis

Tabel 4.17. Kasus dan Hasil Pengujian Pencarian Data Rekam Medis

Kasus Dan Hasil Uji (Data Normal)			
Data Masukan	**Yang Diharapkan**	**Pengamatan**	**Kesimpulan**
Masukkan periode yang	Pencarian data ke dalam file	Dapat melakukan pencarian data	[X] Diterima [ ] Ditolak

akan dicari	rekam medis	dan hasilnya ditampilkan pada tabel	
Klik tombol cari	Data bisa ditemukan	Muncul pesan “Data ditemukan”, dan muncul data yang dicari.	[X] Diterima [ ] Ditolak
Kasus Dan Hasil Uji (Data Salah)			
Periode tidak diisi / salah	Tidak dapat melakukan pencarian	Pencarian data gagal dilakukan	Tidak dapat melakukan pencarian
Klik tombol cari	Data tidak ditemukan	Muncul pesan “Data tidak ditemukan	Tidak dapat melakukan pencarian

4.5.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian yang sudah dilakukan dengan beberapa kondisi yang memungkinkan pada saat program dijalankan, maka dapat ditarik kesimpulan bahwa perangkat lunak yang dibuat sudah berjalan dengan cukup baik sebagaimana yang diharapkan oleh perusahaan.