# BAB V

# IMPLEMENTASI DAN PENGUJIAN SISTEM

## 5.1. Implementasi

Tahap selanjutnya dalam membangun sebuah perangkat lunak adalah implementasi. Implementasi merupakan merupakan kumpulan dari elemen-elemen yang telah didesain kedalam bentuk pemograman untuk menghasilkan suatu tujuan yang dibuat berdasarkan kebutuhan yang telah dianalisis sebelumnya. Implementasi *waterfall* Sistem Informasi Rekam Medis pada Klinik Utama Cahaya Qalbu dibentuk menggunakan bahasa dan lingkungan pemrograman *Java*, dengan basis data yang digunakan adalah MySQL. Bahasa pemograman *Java* dapat dijalankan di berbagai Sistem Operasi tetapi penulis disini menggunakan Windows XP Professional dikarenakan cukup banyak dan sering digunakan.

### 5.1.1. Batasan Implementasi

Dalam proses pengimplementasian perangkat lunak ini, batasan digunakan sebagai pengendalian aplikasi, ada beberapa hal yang perlu dibatasi dalam implementasi ini, yaitu :

1. Perangkat lunak yang digunakan dalam pengimplementasian basis data ini adalah MySQL yang telah terkompilasi kedalam sebuah perangkat lunak bebas yaitu xampp.
2. Hanya bagian terkait yang dapat menggunakan sistem informasi rekam medis pasien ini.

3. Tidak semua kebutuhan Sistem Informasi Rekam medis dapat terpenuhi, tetapi hanya pada subsistem yang meliputi pengelolaan data pasien, pendaftaran pasien, resep, data obat, data penanganan, dan cetak bukti bayar.
4. Sistem yang diterapkan masih mengacu kepada sistem terakhir yang di observasi.

### 5.1.2. Implementasi Perangkat Lunak

Perangkat lunak yang diperlukan dan dipakai dalam pembuatan aplikasi sistem informasi ini adalah :

1. Windows XP Professional SP3 sebagai sistem operasi yang penulis pakai
2. Netbeans IDE 7.1.1 sebagai aplikasi perangkat lunak dalam membangun aplikasi sistem informasi penjualan
3. iReport sebagai perancang report yang merupakan '*Visual Report Designer*' untuk *JasperReports* yang merupakan salah satu aplikasi yang sangat berguna untuk pembuatan laporan, kartu pasien dan juga bukti pembayaran
4. Sementara itu penulis menggunakan MySQL sebagai pengembang dalam pembuatan basis data/*database* dan perangkat lunaknya menggunakan xampp WIN32 1.7.7.

### 5.1.3. Implementasi Perangkat Keras

**Kebutuhan Minimal**

1. Proccesor : Intel Pentium III 800 MHz
2. Memory : 512 Mb RAM
3. VGA : 64Mb
4. Printer : Standar
5. Hard Disk : 20 Gb
6. Mouse dan Keyboard

**Kebutuhan yang dianjurkan**

1. Proccesor : Pentium IV 2.8 GHz (atau lebih tinggi)
2. Memory : 2 Gb RAM (atau lebih besar)
3. VGA : 128 Mb
4. Printer : DeskJet
5. Harddisk : 80 Gb
6. Mouse dan Keyboard

### 5.1.4. Implementasi Basis Data (Sintaks SQL)

Implementasi basis data dilakukan dengan menggunakan bahasa SQL, dimana aplikasi pemrograman yang digunakan adalah MySQL, implementasi basis datanya dalam bahas SQL adalah sebagai berikut :

```
--CREATE DATABASE dbklinikcq;
--use CQ
--
-- Table structure for table `tb_petugas`
--
CREATE TABLE `tb_petugas` (
  `id_petugas` varchar(10) collate latin1_swedish_ci NOT NULL,
```

```
   `nama_petugas` varchar(30) collate latin1_swedish_ci NOT NULL,
  `username` varchar(20) collate latin1_swedish_ci NOT NULL,
  `password` varchar(20) collate latin1_swedish_ci NOT NULL,
  `type` varchar(20) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_petugas`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;
--
-- Table structure for table `pasien`
--
CREATE TABLE `pasien` (
  `tgl_daftar` varchar(10) collate latin1_swedish_ci NOT NULL,
  `id_pasien` varchar(10) collate latin1_swedish_ci NOT NULL,
  `nama_pasien` varchar(30) collate latin1_swedish_ci NOT NULL,
  `jenkel` varchar(12) collate latin1_swedish_ci NOT NULL,
    `usia` varchar(10) collate latin1_swedish_ci NOT NULL,
    `alamat` varchar(50) collate latin1_swedish_ci NOT NULL,
       `no_hp` varchar(20) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_pasien`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;
--
-- Table structure for table `tb_antri_umum`
--
CREATE TABLE `tb_antri_umum` (
  `id_antri_umum` varchar(10) collate latin1_swedish_ci NOT NULL,
  `id_pasien` varchar(10) collate latin1_general_ci NOT NULL,
    `nama_um` varchar(30) collate latin1_swedish_ci NOT NULL,
  `alamat_um` varchar(50) collate latin1_swedish_ci NOT NULL,
    `tgl_bayar_umum` varchar(10) collate latin1_swedish_ci NOT NULL,
    `keterangan_u` varchar(20) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_antri_umum`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;

--
-- Table structure for table `tb_antri_gigi`
--
CREATE TABLE `tb_antri_gigi` (
  `id_antri_gigi` varchar(10) collate latin1_swedish_ci NOT NULL,
```

```
  `id_pasien` varchar(10) collate latin1_general_ci NOT NULL,
   `nama_gg` varchar(30) collate latin1_swedish_ci NOT NULL,
  `alamat_gg` varchar(50) collate latin1_swedish_ci NOT NULL,
   `tgl_bayar_gigi` varchar(10) collate latin1_swedish_ci NOT NULL,
   `keterangan ` varchar(20) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_antri_gigi`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;


--
-- Table structure for table `tb_medrec_gigi`
--
CREATE TABLE `tb_medrec_gigi` (
  `id_medrec_gigi` varchar(15) collate latin1_swedish_ci NOT NULL,
  `id_pasien` varchar(10) collate latin1_general_ci NOT NULL,
   `nama_pgg` varchar(30) collate latin1_swedish_ci NOT NULL,
  `tgl_pem_gigi` varchar(10) collate latin1_swedish_ci NOT NULL,
   `diagnosa_gg` varchar(300) collate latin1_swedish_ci NOT NULL,
   `penanganan_gg` varchar(300) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_medrec_gigi`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;

--
-- Table structure for table `tb_medrec_umum`
--
CREATE TABLE `tb_medrec_umum` (
  `id_medrec_umum` varchar(15) collate latin1_swedish_ci NOT NULL,
  `id_pasien` varchar(10) collate latin1_general_ci NOT NULL,
   `nama_pum` varchar(30) collate latin1_swedish_ci NOT NULL,
  `tgl_pem_umum` varchar(10) collate latin1_swedish_ci NOT NULL,
   `diagnosa` varchar(300) collate latin1_swedish_ci NOT NULL,
   `penanganan` varchar(300) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_medrec_gigi`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;

--
-- Table structure for table `tb_bayar_obat`
```

```
--
CREATE TABLE `tb_bayar_obat` (
  `id_bayar` varchar(10) collate latin1_swedish_ci NOT NULL,
  `id_pasien` varchar(10) collate latin1_general_ci NOT NULL,
  `id_medrec` varchar(10) collate latin1_general_ci NOT NULL,
   `nama_byr` varchar(30) collate latin1_swedish_ci NOT NULL,
  `poliklinik` varchar(30) collate latin1_swedish_ci NOT NULL,
   `penanganan` varchar(300) collate latin1_swedish_ci NOT NULL,
   `tgl_bayar` varchar(10) collate latin1_swedish_ci NOT NULL,
   `bayar_resep` int(20) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_bayar`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;

--
-- Table structure for table `tb_biaya`
--
CREATE TABLE `tb_biaya` (
  `id_biaya` varchar(10) collate latin1_swedish_ci NOT NULL,
  `nama_biaya` varchar(30) collate latin1_general_ci NOT NULL,
   `biaya` int(10) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_biaya`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;

--
-- Table structure for table `tb_obat`
--
CREATE TABLE `tb_obat` (
  `id_obat` varchar(10) collate latin1_swedish_ci NOT NULL,
  `nama_obat` varchar(30) collate latin1_general_ci NOT NULL,
   `harga` int(10) collate latin1_swedish_ci NOT NULL,
   `satuan` varchar(10) collate latin1_swedish_ci NOT NULL,
  PRIMARY KEY  (`id_obat`)
) ENGINE=MyISAM DEFAULT CHARSET=latin1
COLLATE=latin1_swedish_ci;
```

### 5.1.5. Implementasi Antar Muka

Implementasi dilakukan dengan sebuah *file project* yang berektensi *.java* yang ada dalam program *netbeans* yang didalamnya terdapat form-form untuk pembuatan suatu aplikasi.

#### 5.1.5.1 Implementasi *Login*

Merupakan kunci utama untuk masuk ke dalam program aplikasi, inputan yang dimasukan harus benar, apabila salah maka *user* akan gagal menggunakan aplikasi ini.

**Tabel 5.1** Implementasi *Login*

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Menu Utama | *File* program yang menampilkan menu utama | FUtama.java |
| *Login* | Untuk *Login* pengguna | - |
| *Exit* | Untuk keluar dari program | - |

#### 5.1.5.2 Implementasi Halaman Utama

Implementasi halaman utama perangkat lunak ini adalah sebagai berikut:

**Tabel 5.2** Implementasi halaman utama

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Menu Utama | File program yang menampilkan menu utama | FUtama.java |

| | | |
|---|---|---|
| *File* | Untuk menampilkan sub-menu *File* | - |
| Managemen Data | Untuk menampilkan sub-menu Managemen Data | - |
| Registrasi | Untuk menampilkan sub-menu Registrasi | - |
| Pembayaran | Untuk menampilkan sub-menu Pembayaran | - |
| Pemeriksaan | Untuk menampilkan sub-menu Pemeriksaan | - |
| Laporan | Untuk menampilkan sub-menu Laporan | - |

#### 5.2.5.2 Implementasi Halaman Berdasarkan sub *Menu File*

Implementasi halaman berdasarkan sub *menu file* yang dibuat dalam bentuk *file* program dapat dilihat pada tabel berikut :

**Tabel 5.3** Implementasi halaman berdasarkan sub menu *file*

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| *Login* | Menangani *login* pengguna | - |
| *Logout* | Menangani *logout* pengguna | - |
| *Exit* | Perintah keluar dari program | - |

#### 5.1.5.3 Implementasi Halaman Berdasarkan Sub Menu Managemen Data

Implementasi halaman berdasarkan sub menu managemen data yang dibuat dalam bentuk file program dapat dilihat pada tabel berikut:

**Tabel 5.4** Implementasi halaman berdasarkan sub menu managemen data

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Data *User* | *File* program untuk menambah, mengedit dan menghapus data *User* | FUser.java |
| Data Biaya | *File* program untuk menambah, mengedit dan menghapus data Biaya | FBiaya |
| Data Obat | *File* program untuk menambah, mengedit dan menghapus data Obat | FObat |

#### 5.1.5.4 Implementasi Halaman Berdasarkan Sub Menu Registrasi

Implementasi halaman berdasarkan sub menu registrasi yang dibuat dalam bentuk *file* program dapat dilihat pada tabel berikut :

**Tabel 5.5** Implementasi halaman berdasarkan sub menu registrasi

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Pendaftaran | *File* program untuk menginput , merubah , mencari atau | FPendaftaran.java |

| | | |
|---|---|---|
| | menghapus data pasien | |

### 5.1.5.5 Implementasi Halaman Berdasarkan Sub Menu Pembayaran

Implementasi halaman berdasarkan sub menu pembayaran yang dibuat dalam bentuk *file* program dapat dilihat pada tabel berikut :

**Tabel 5.5** Implementasi halaman berdasarakan sub menu pembayaran

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Poli Umum | *File* program untuk menginput data antri pasien setelah melakukan pembayaran pendaftaran. | FBayarUmum.java |
| Poli Gigi | *File* program untuk menginput data antri pasien setelah melakukan pembayaran pendaftaran juga untuk melakukan pembayaran penanganan medis tambahan pada poli gigi. | FBayarGigi.java |

### 5.1.5.6 Implementasi Halaman Berdasarkan Sub Menu Pemeriksaan

Implementasi halaman berdasarkan sub menu pemeriksaan yang dibuat dalam bentuk *file* program dapat dilihat pada tabel berikut :

**Tabel 5.6** Implementasi halaman berdasarakan sub menu pemeriksaan

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Poli Umum | *File* program untuk menginput data rekam medis pada poli umum dan melihat riwayat rekam medis | FPemUmum.java |
| Poli Gigi | *File* program untuk menginput data rekam medis pada poli gigi dan melihat riwayat rekam medis | FPemGigi.java |
| *View* Antri Umum | Untuk melihat daftar antri pasien poliklinik umum per hari pemeriksaan | FDataUmum.java |
| *View* Antri Gigi | Untuk melihat daftar antri pasien poliklinik gigi per hari pemeriksaan | FDataGigi.java |

### 5.1.5.7 Implementasi Halaman Berdasarkan Sub Menu Laporan

Implementasi halaman berdasarkan sub menu laporan yang dibuat dalam bentuk *file* program dapat dilihat pada tabel berikut :

**Tabel 5.7** Implementasi halaman berdasarkan sub menu laporan

| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| Laporan | *File* Program untuk mencetak atau *view* data laporan keuangan pada Klinik Utama Cahaya Qalbu pada berbagai bagian | FLaporan.java |

#### 5.1.5.8 Implementasi Halaman Berdasarkan Sub Menu *About*

Implementasi halaman berdasarkan sub menu *About* yang dibuat dalam bentuk *file* program dapat dilihat pada tabel berikut :

**Tabel 5.8** Implementasi halaman berdasarkan sub menu about

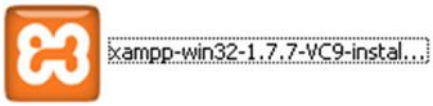
| Sub Menu | Deskripsi | Nama File |
|---|---|---|
| About | *File* program untuk melihat hak penciptaan/spesifikasi *software* dan data programer | FAbout.java |

### 5.1.6. Implementasi Instalasi Program

Dalam proses Implementasi Instalasi Program ini, akan dijelaskan bagaimana cara menginstal XAMPP sebagai pengembang dalam pengelolaan basis data dan juga cara menginstal program yang di usulkan oleh penulis kepada perusahaan tempat penulis melakukan penelitian.

#### 5.1.6.1 Instalasi XAMPP

Pertama yang harus dilakukan adalah membuka file instalasi xampp win32-1 1.7.7 yang berekstensi.exe lalu double click untuk menginstalnya.

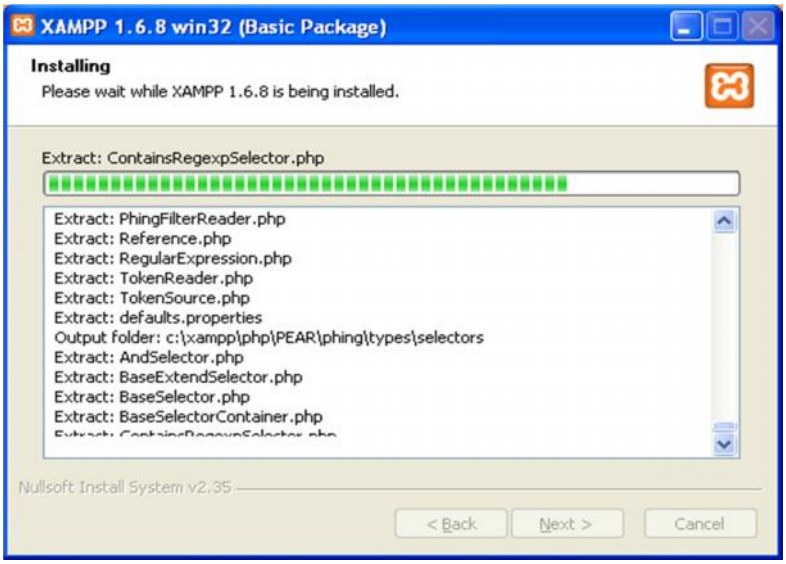



**Gambar 5.1** Icon Instalasi xampp-win32-1.7.7

Setelah mendouble click, selanjutnya akan tampil seperti pada gambar 5.2 lalu pilih next dan ikuti langkah selanjutnya untuk menginstal.




**Gambar 5.2** Tampilan Awal Instalasi xampp



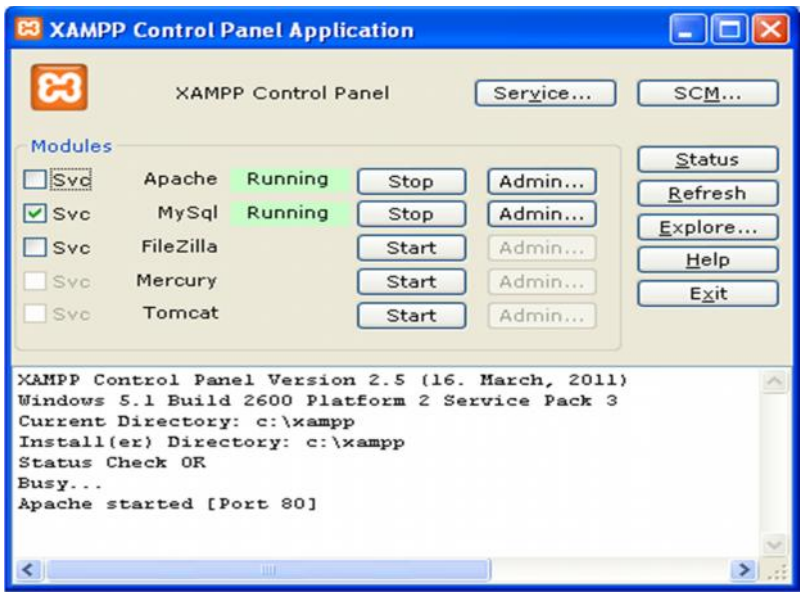

**Gambar 5.3** Tampilan Proses Instalasi xampp

Tunggu sampai proses instalasi selesai, kemudian klik Finish. Maka dalam jendela windows akan otomatis muncul icon xampp-control seperti pada gambar 5.4.



**Gambar 5.4** Icon xampp-control

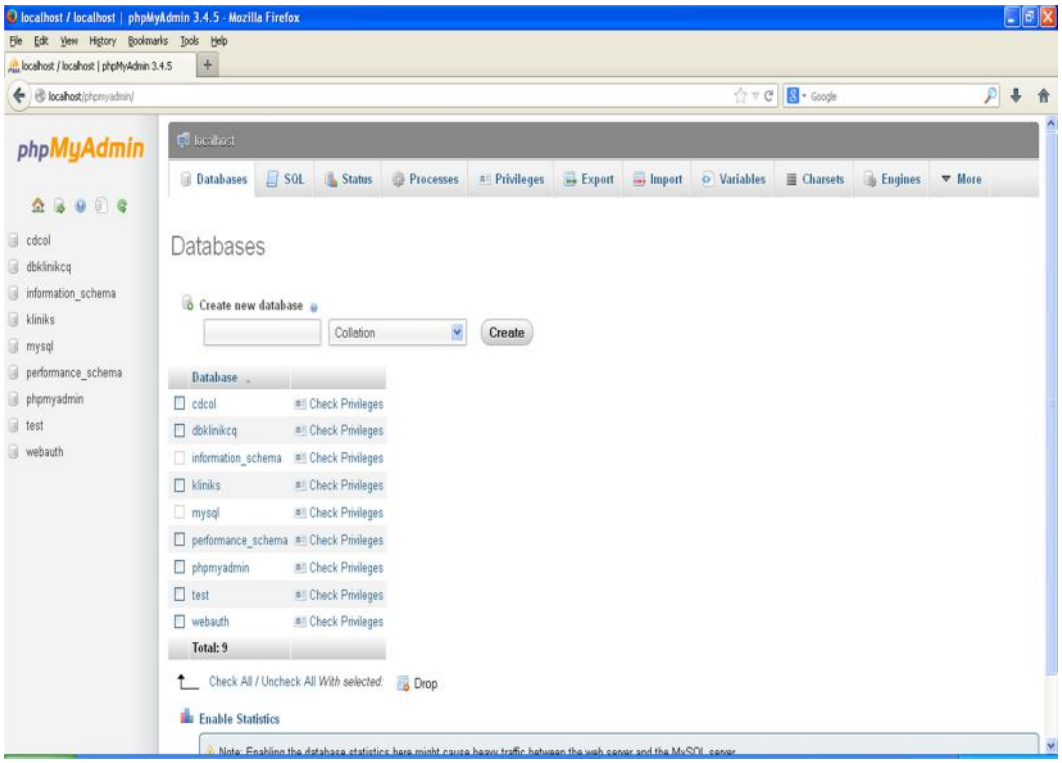
Setelah muncul di desktop, double click xampp-control untuk menjalankannya. Maka akan muncul tampilan seperti pada gambar 5.5




**Gambar 5.5** Tampilan xampp control panel

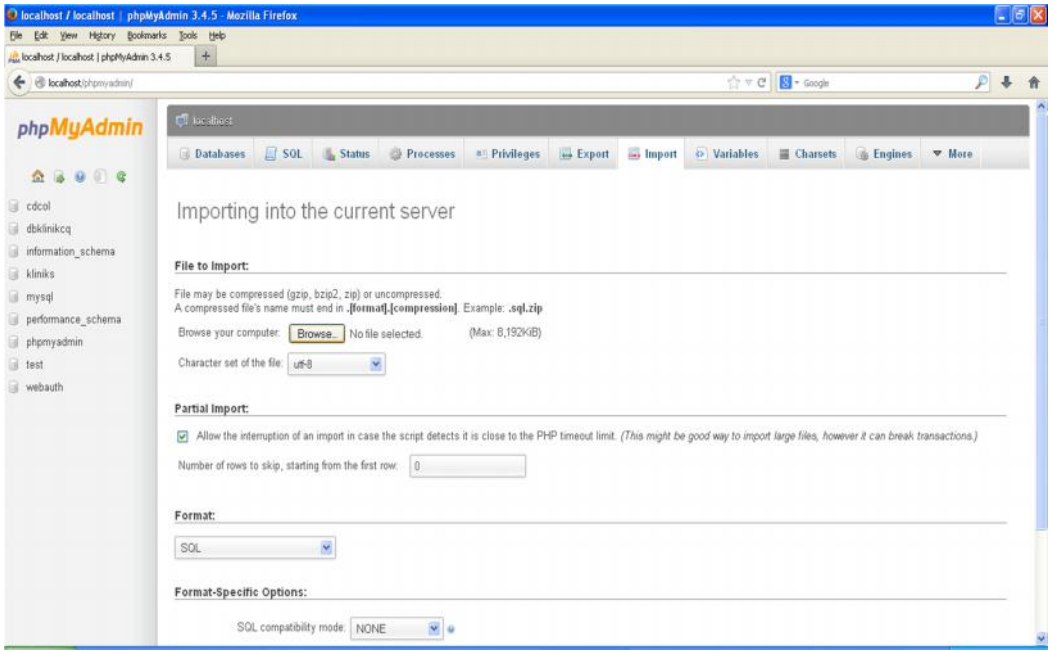
Jalankan Apache dan MySql nya dengan menekan tombol start di sisi kanannya sehingga pada Apache dan MySql muncul progressbar "Running".

Untuk setting database Klik "admin.." dari xampp control panel atau ketikkan "localhost/phpmyadmin" pada browser anda sehingga akan tampil seperti berikut.




**Gambar 5.6** Tampilan PhpMyAdmin

Kemudian klik tab import lalu klik Browse, setelah itu pilih database yang berformat .sql yaitu "dbklinikcq.sql" kemudian klik Go dan selesai.




**Gambar 5.7** Tampilan Import Database

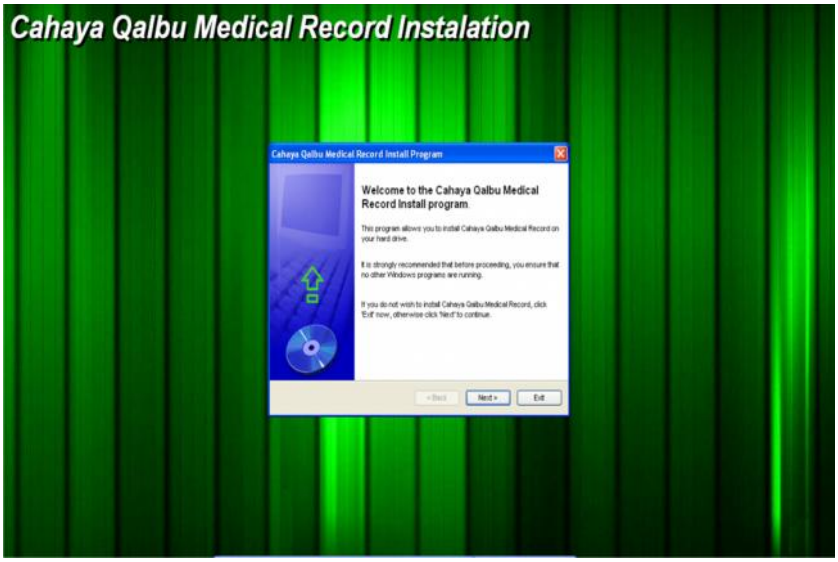
### 5.1.6.2 Instalasi Program Klinik Cahaya Qalbu

Langkah pertama yang dilakukan adalah dengan membuka file instalasi program Cahaya Qalbu dengan melakukan double klik pada program tersebut.



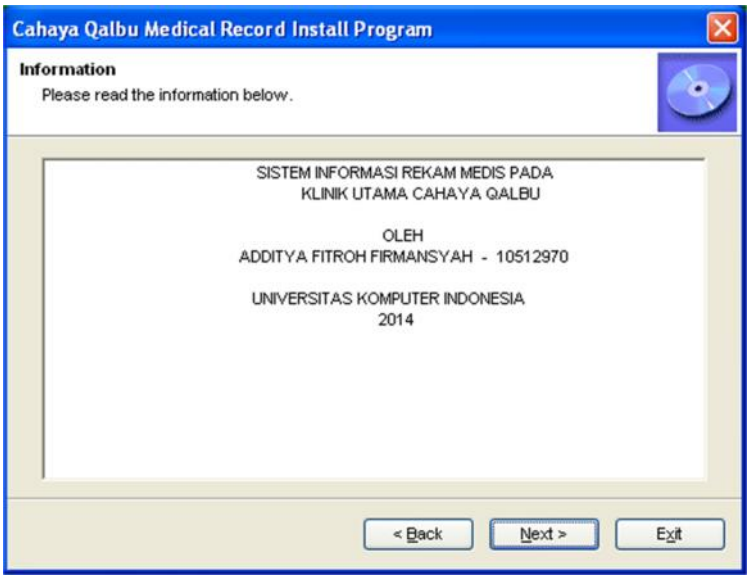

**Gambar 5.8** Icon Instalasi Program Cahaya Qalbu

Setelah itu kemudian akan tampil tampilan awal instalasi program Cahaya Qalbu ini,klik *next.*



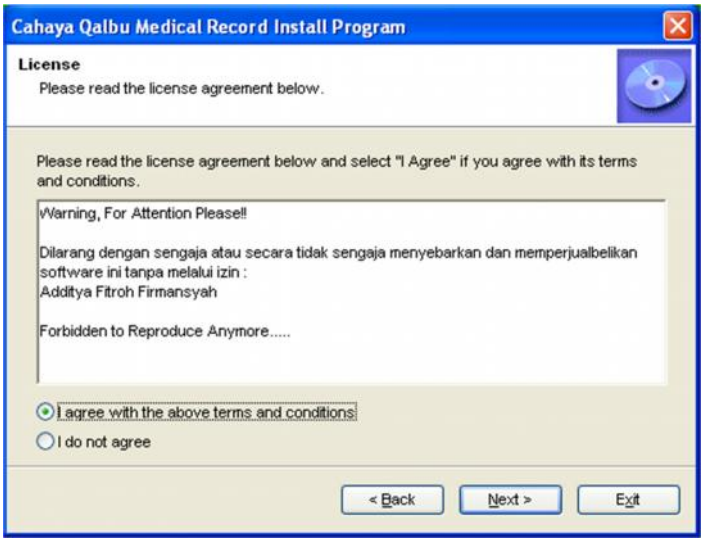

**Gambar 5.9** Instalasi Program CQ

Setelah itu akan tampil keterangan pembuat program dan keterangan nama program yang akan di install.Klik *next.*



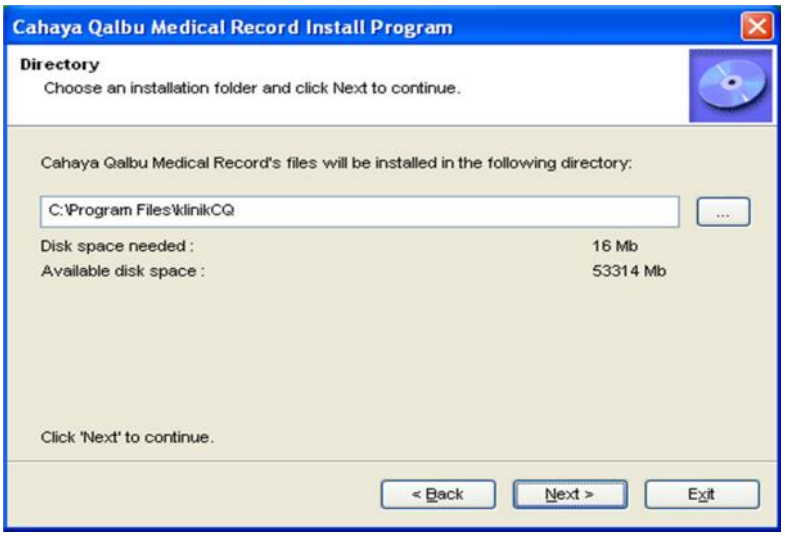

**Gambar 5.10** Informasi Program CQ

Kemudian akan ada persetujuan yang dibuat untuk program yang akan digunakan juga berisi peringatan.Klik *next.*




**Gambar 5.11** License Instalasi Program CQ

Lalu akan ada konfirmasi direktori kemana program itu akan di install,pilih direktori sesuai keinginan anda dengan mengklik "…" atau jika sudah sesuai klik *next.*



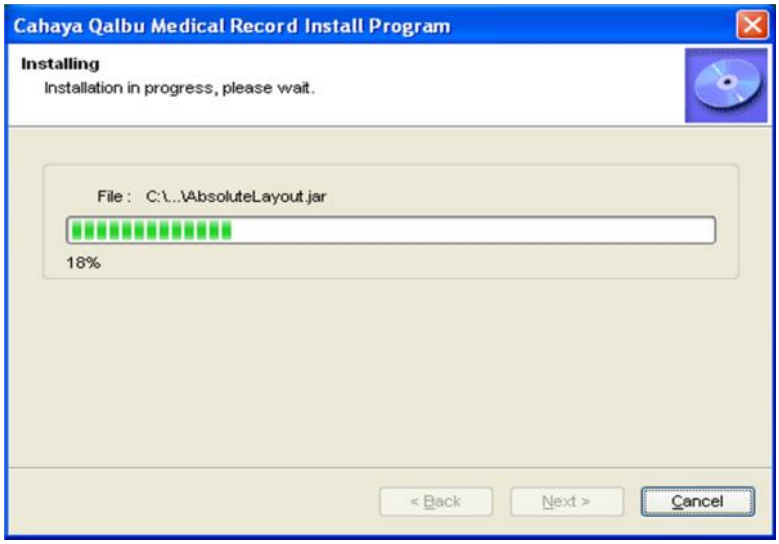

**Gambar 5.12** Directory Instalasi Program CQ

Kemudian ditampilkan pesan persetujuan untuk membuat *folder* destinasi program klik "*yes*" jika setuju dan "*no*" jika ingin meninjau kembali.



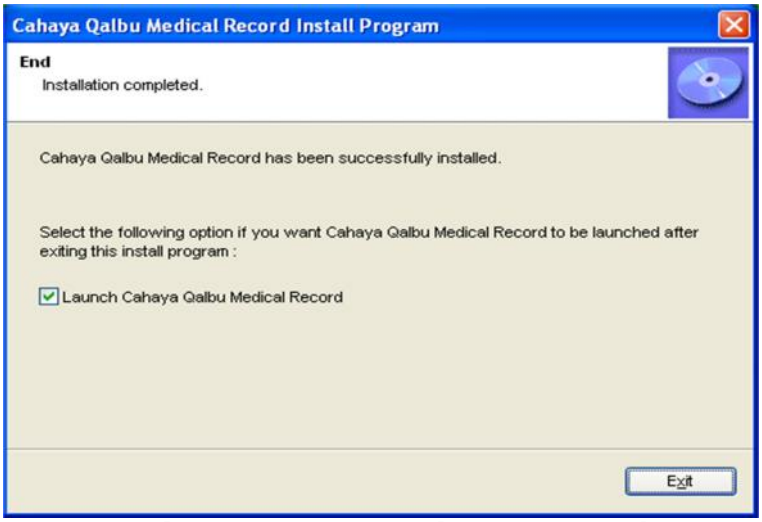

**Gambar 5.13** Directory Folder Instalasi Program CQ

Tunggu beberapa saat untuk proses instalasi program Cahaya Qalbu.




**Gambar 5.14** Proses Instalasi Program CQ

Setelah beberapa saat proses instalasi selesai,centang kotak jika langsung menggunakan program ini kemudian klik *next.*



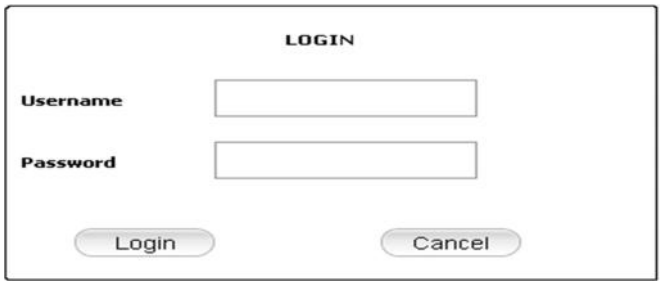

**Gambar 5.15** Proses Instalasi Program Selesai

### 5.1.7 Penggunaan Program

Kegiatan ini bertujuan untuk menerangkan secara singkat penggunaan program sistem informasi penginputan data pasien. Adapun tampilannya adalah sebagai berikut :

1. Tampilan Form *Login*

   Form *login* ini berfungsi sebagai keamanan sistem. Untuk melakukan login, isi data *login* dengan *Username* "superadmin" dan *password* "superadmin" dan klik tombol *Login*, jika berhasil maka form utama akan memunculkan *menu bar* sesuai hak akses yang telah ditentukan. Tampilan form ini adalah sebagai berikut :




**Gambar 5.16** Tampilan Form *Login*

Apabila ada kesalahan dalam pengisian *username* atau *password* maka akan muncul tampilan pesan berikut ini :

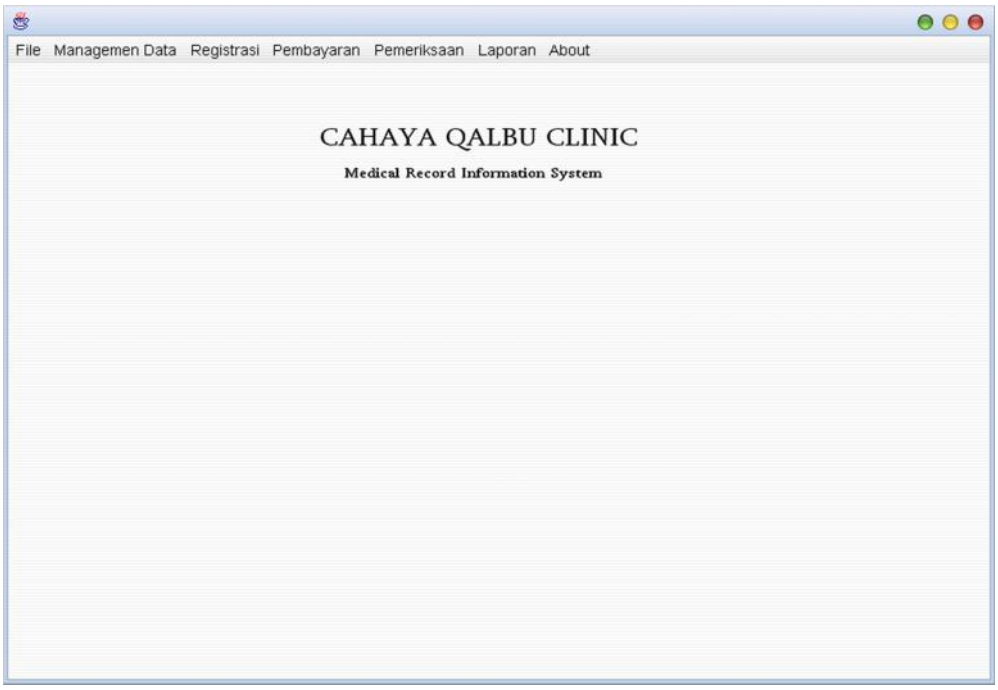



**Gambar 5.17** Tampilan Pesan Kesalahan *Input*

2. Tampilan Form *Home*

Tampilan menu utama akan memunculkan menu-menu yang bisa digunakan oleh *user* setelah validasi *username* dan *password* pada form *login* dan menu yang bisa diakses adalah :

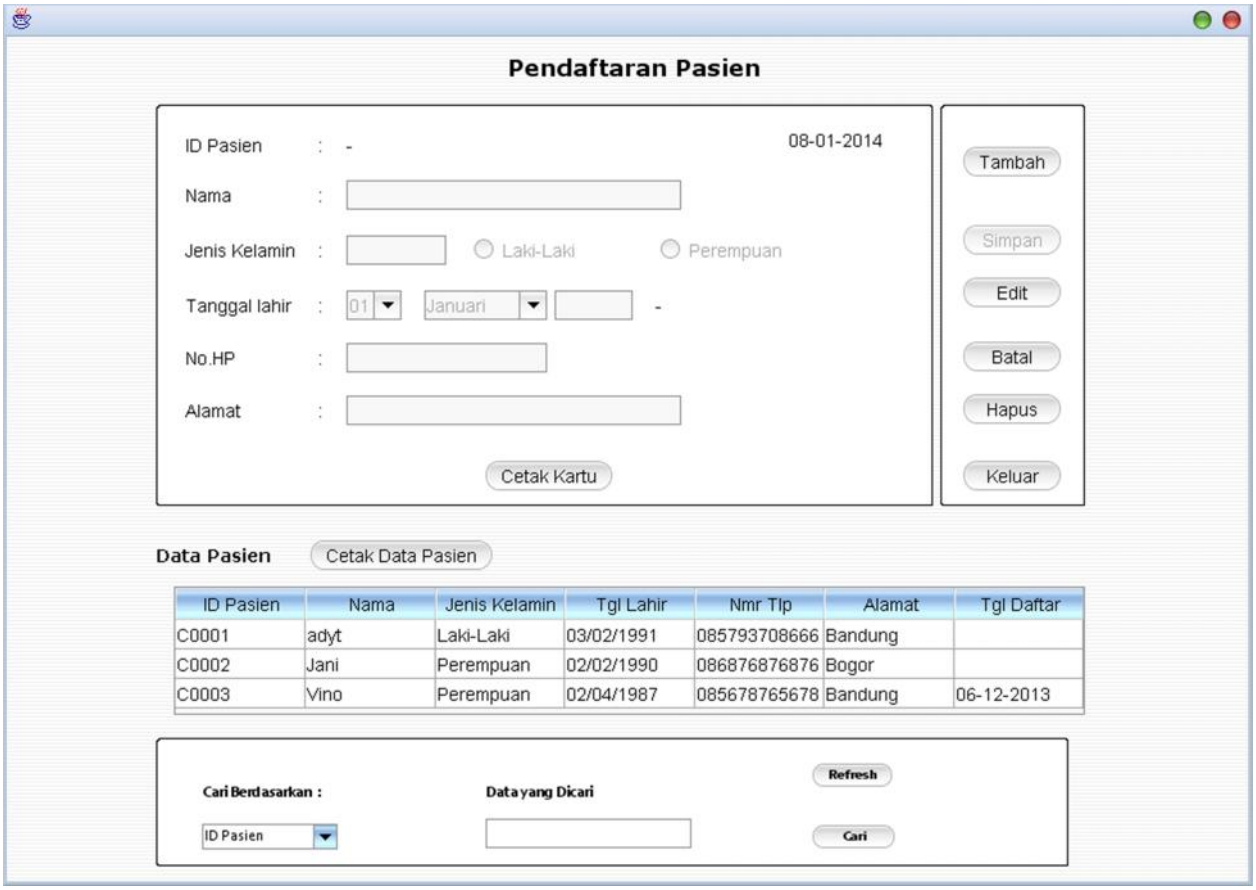
1. *File* : Submenu *Logout* dan *Exit*
2. Managemen Data : Submenu Data User, Biaya dan Obat
3. Registrasi : Submenu Pendaftaran
4. Pembayaran : Submenu Poli Umum, Gigi dan Pembayaran
5. Pemeriksaan : Submenu Poli Umum, Gigi, View Poli Umum dan View Poli Gigi
6. Laporan : Submenu Data Laporan
7. *About* : Submenu *About*




**Gambar 5.18** Tampilan Menu Utama

3. Tampilan Form Pendaftaran

Digunakan untuk menginput data pasien sekaligus melakukan pencarian data pasien terdiri dari *field* ID Pasien, Nama, Tgl, Jenis Kelamin, TglLahir, Alamat, dan Nmr.Hp. Pertama klik *button* Tambah sehingga *field* menjadi aktif dan *field* ID Pasien akan otomatis Menampilkan ID Pasien yang akan diinputkan selanjutnya isi semua *field* kemudian klik Simpan dan Cetak Kartu Pasien untuk mencetak kartu pasien. Klik Keluar Untuk Keluar Dari Form Ini.

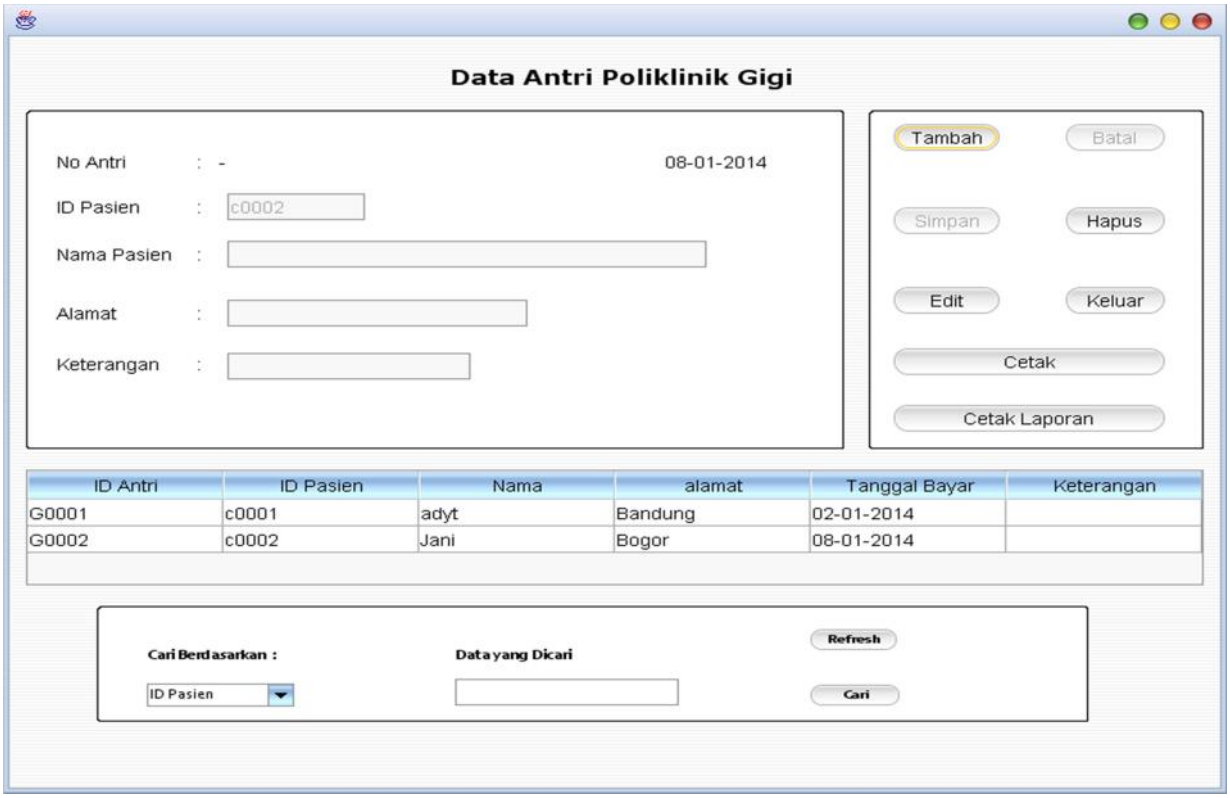



**Gambar 5.19** Tampilan Form Pendaftaran

4. Tampilan Form Data Antri Gigi

Digunakan untuk menentukan nomor antri dan memasukan data pasien Poli Gigi pada status pasien tunggu untuk melakukan pemeriksaan, terdiri dari *field* No Antri, Tgl, ID Pasien, Nama, Alamat dan Keterangan. Pertama Klik *button* Tambah untuk mengaktifkan *field* dan *field* nomor antri terisi secara otomatis berdasarkan urutan pasien yang sebelumnya mendaftar, kemudian isi *field* ID Pasien berdasarkan yang tertera dalam kartu pasien sehingga ketika pada *field* ID Pasien ditekan *Enter* maka *field*

Nama dan Alamat Pasien akan terisi karena mengambil data pasien dari *database* lalu klik Simpan dan klik Cetak untuk mencetak bukti antri bila diperlukan. Klik keluar untuk keluar dari Form.

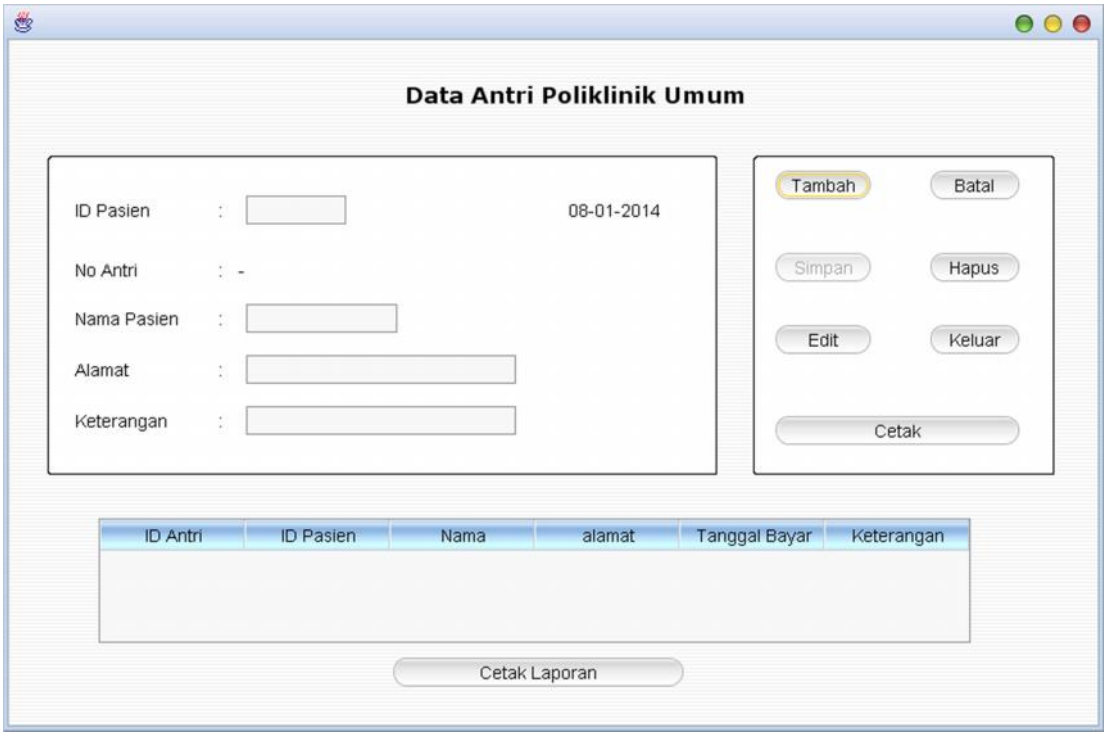



**Gambar 5.20** Tampilan Form Data Antri Gigi

5. Tampilan Form Data Antri Umum

Digunakan untuk menentukan nomor antri dan memasukan data pasien Poli Umum pada status pasien tunggu untuk melakukan pemeriksaan, terdiri dari *field* No Antri, Tgl, ID Pasien, Nama, Alamat dan Keterangan. Pertama Klik *button* Tambah untuk mengaktifkan *field* dan *field* nomor antri terisi secara otomatis berdasarkan urutan pasien yang sebelumnya mendaftar, kemudian isi *field* ID Pasien berdasarkan yang tertera dalam kartu pasien sehingga ketika pada *field* ID Pasien ditekan *Enter* maka *field*

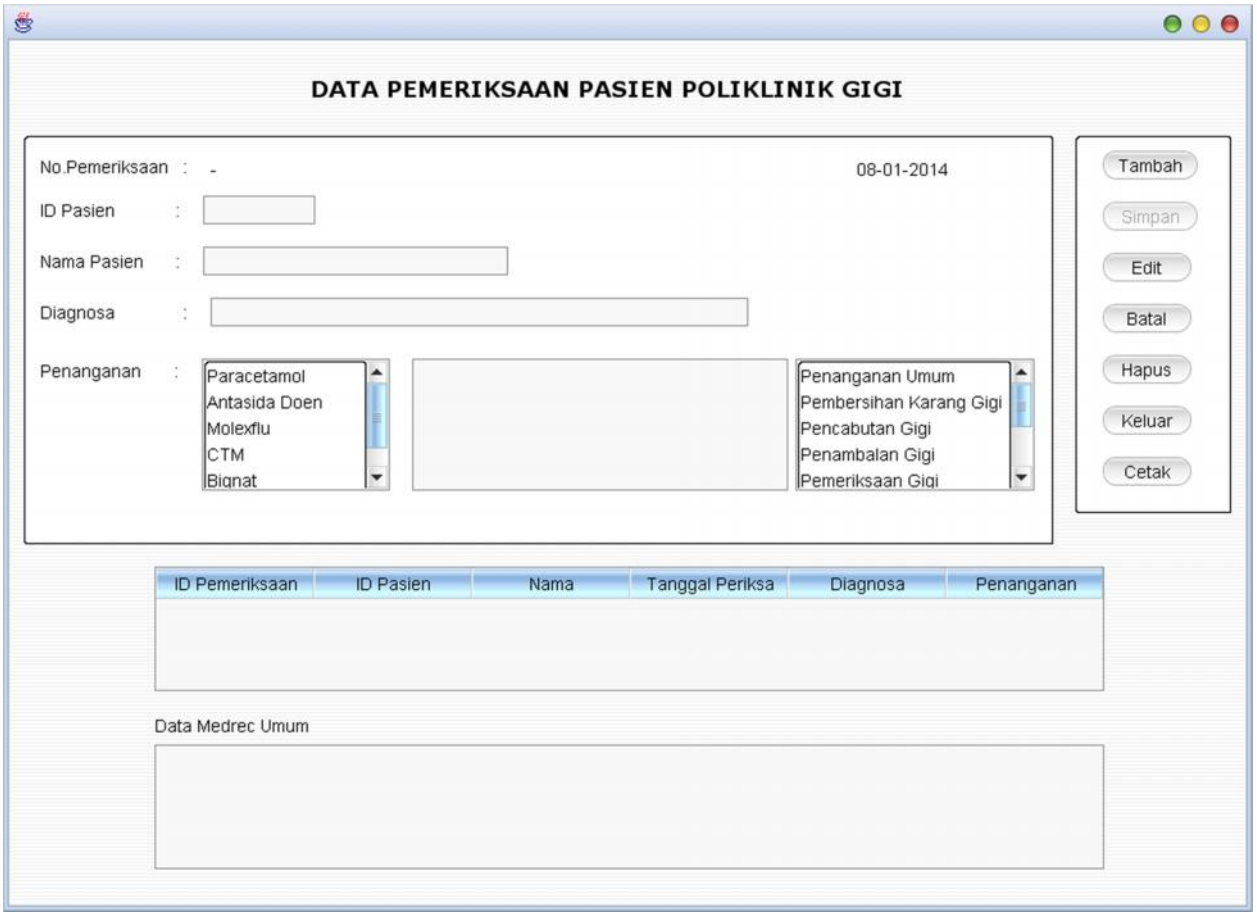
Nama dan Alamat Pasien akan terisi karena mengambil data pasien dari *database* lalu klik Simpan dan klik Cetak untuk mencetak bukti antri bila diperlukan. Klik keluar untuk keluar dari Form.




**Gambar 5.21** Tampilan Form Data Antri Umum

6. Tampilan Form Data Rekam Medis Poli Gigi

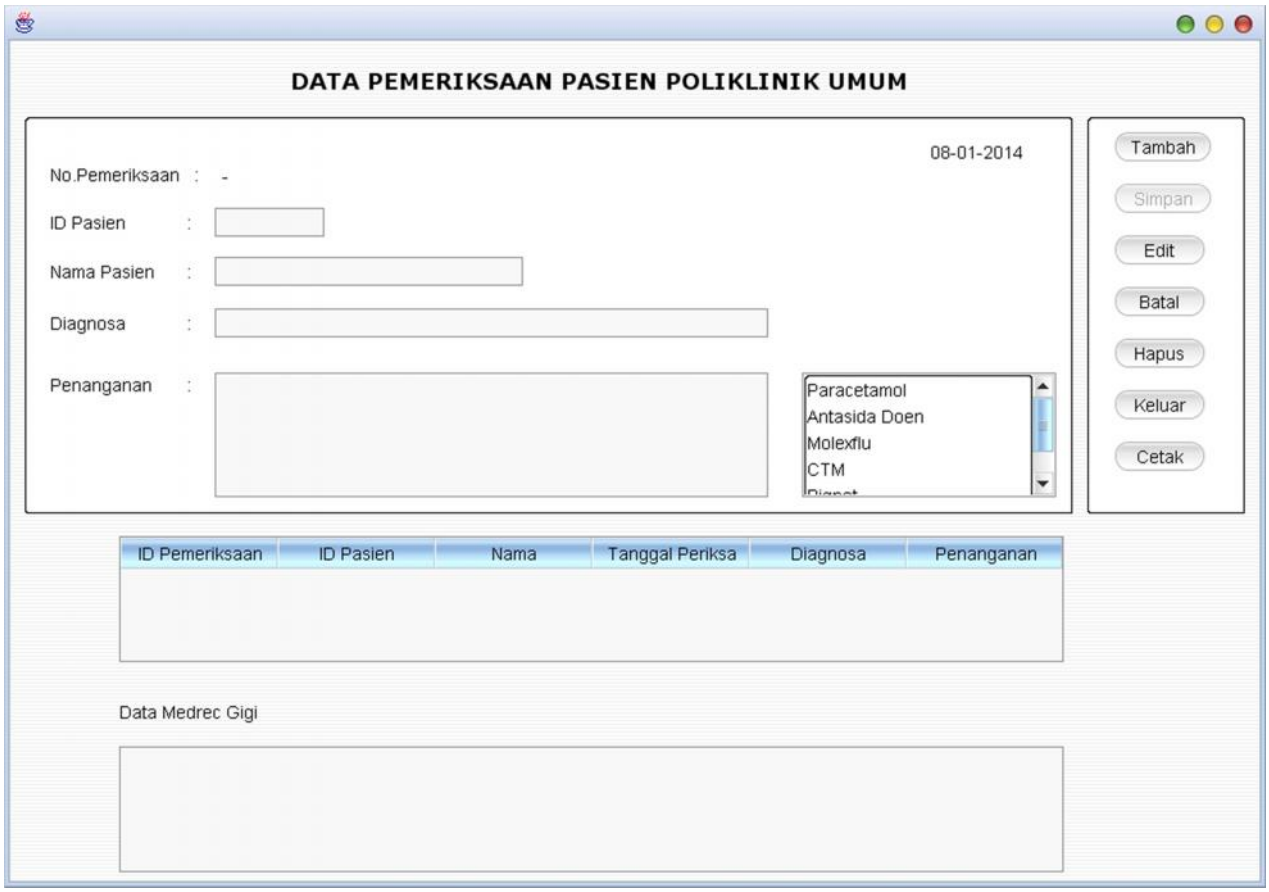
Digunakan untuk mencatat hasil pemeriksaan pasien pada Poli Gigi. Terdiri No.Pemeriksaan, ID Pasien, Nama, Diagnosa dan Penanganan. Pertama Klik Tambah kemudian masukan ID Pasien yang diperiksa, lalu masukan data pemeriksaan, untuk daftar obat atau penanganan cukup dengan sorot dan klik dari daftarnya maka secara otomatis akan tampil pada *field* Penanganan. Kemudian selanjutnya setelah terisi semua *field* Klik tombol cetak lalu klik tombol Simpan. Klik keluar untuk keluar dari Form ini.




**Gambar 5.22** Tampilan Form Data Rekam Medis Poli Gigi

7. Tampilan Form Data Rekam Medis Poli Umum

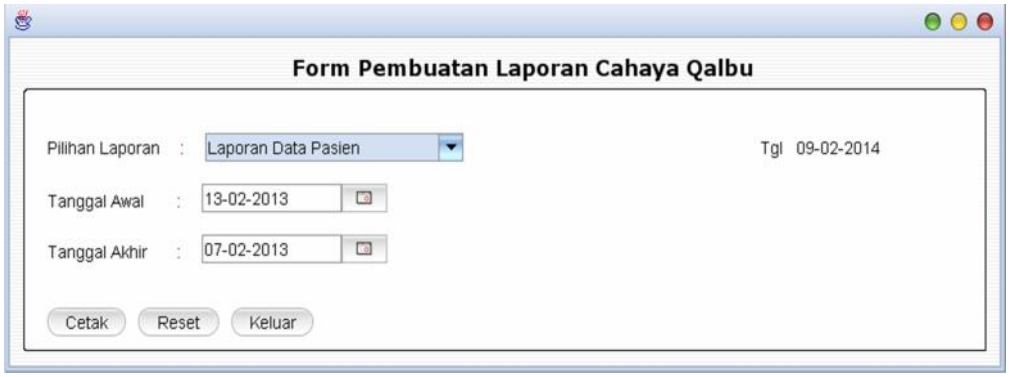
Digunakan untuk mencatat hasil pemeriksaan pasien pada Poli Umum. Terdiri No.Pemeriksaan, ID Pasien, Nama, Diagnosa dan Penanganan. Pertama Klik Tambah kemudian masukan ID Pasien yang diperiksa, lalu masukan data pemeriksaan, untuk daftar obat atau penanganan cukup dengan sorot dan klik dari daftarnya maka secara otomatis akan tampil pada *field* Penanganan. Kemudian selanjutnya setelah terisi semua *field* Klik tombol cetak lalu klik tombol Simpan. Klik keluar untuk keluar dari Form ini.




**Gambar 5.23** Tampilan Form Data Rekam Medis Poli Umum

8. Tampilan Form Laporan

Digunakan jika ingin melihat laporan yang telah terjadi sesuai dengan periode yang diinginkan seperti Laporan Keuangan Poliklinik Umum dan Poliklinik Gigi, pertama isi tanggal awal dan tanggal akhir sesuai dengan laporan yang ingin dibuat kemudian pilih tipe laporan, klik Tombol Cetak Laporan sehingga nanti akan ditampilkan Data Laporan.




**Gambar 5.24** Tampilan Form Laporan

9. Tampilan About

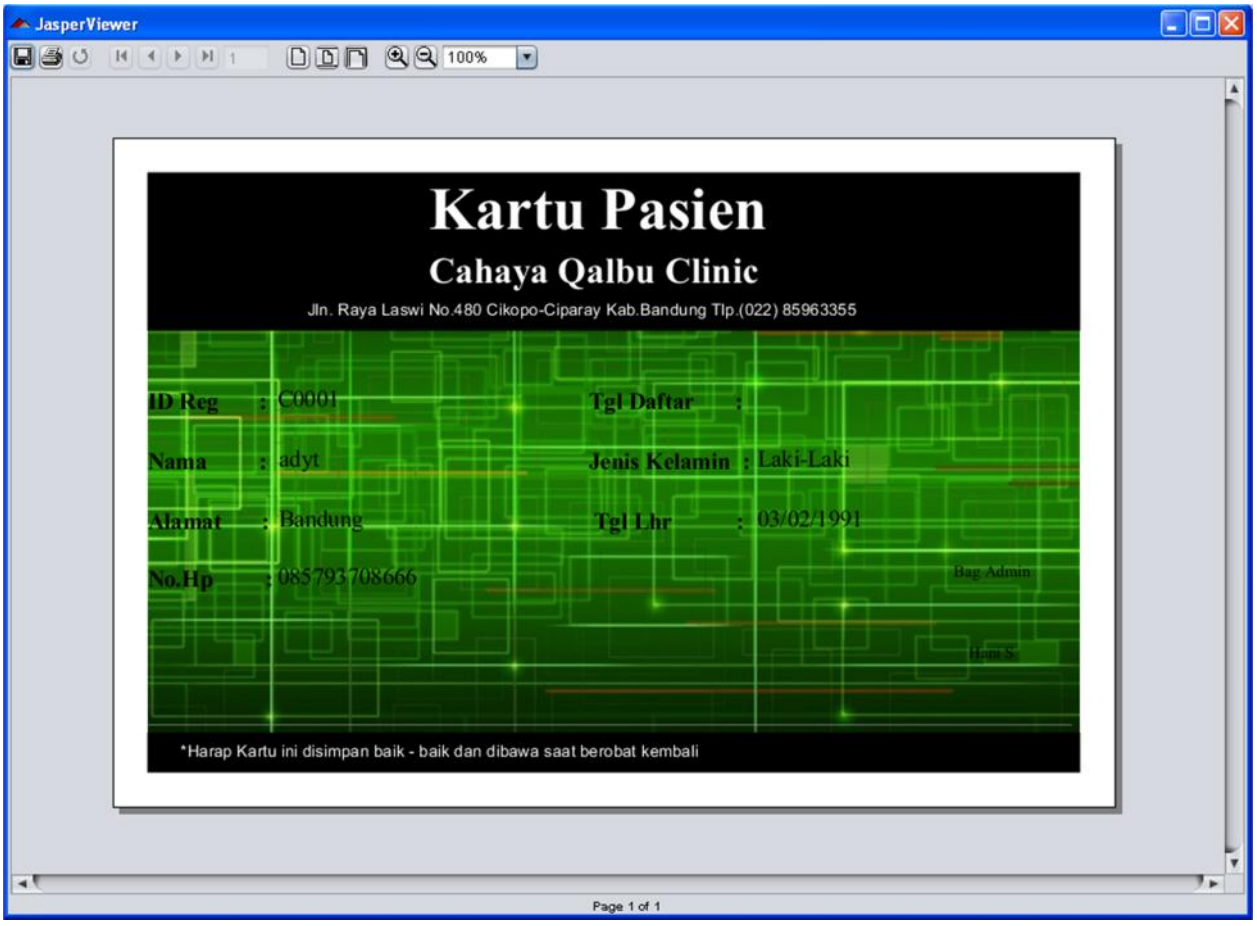
Berisi tentang data programmer yang dibuat juga tujuan mula *software* ini dirancang Bisa diakses dengan mengklik About.




**Gambar 5.25** Tampilan About *Software*

10. Tampilan Cetak Kartu Pasien

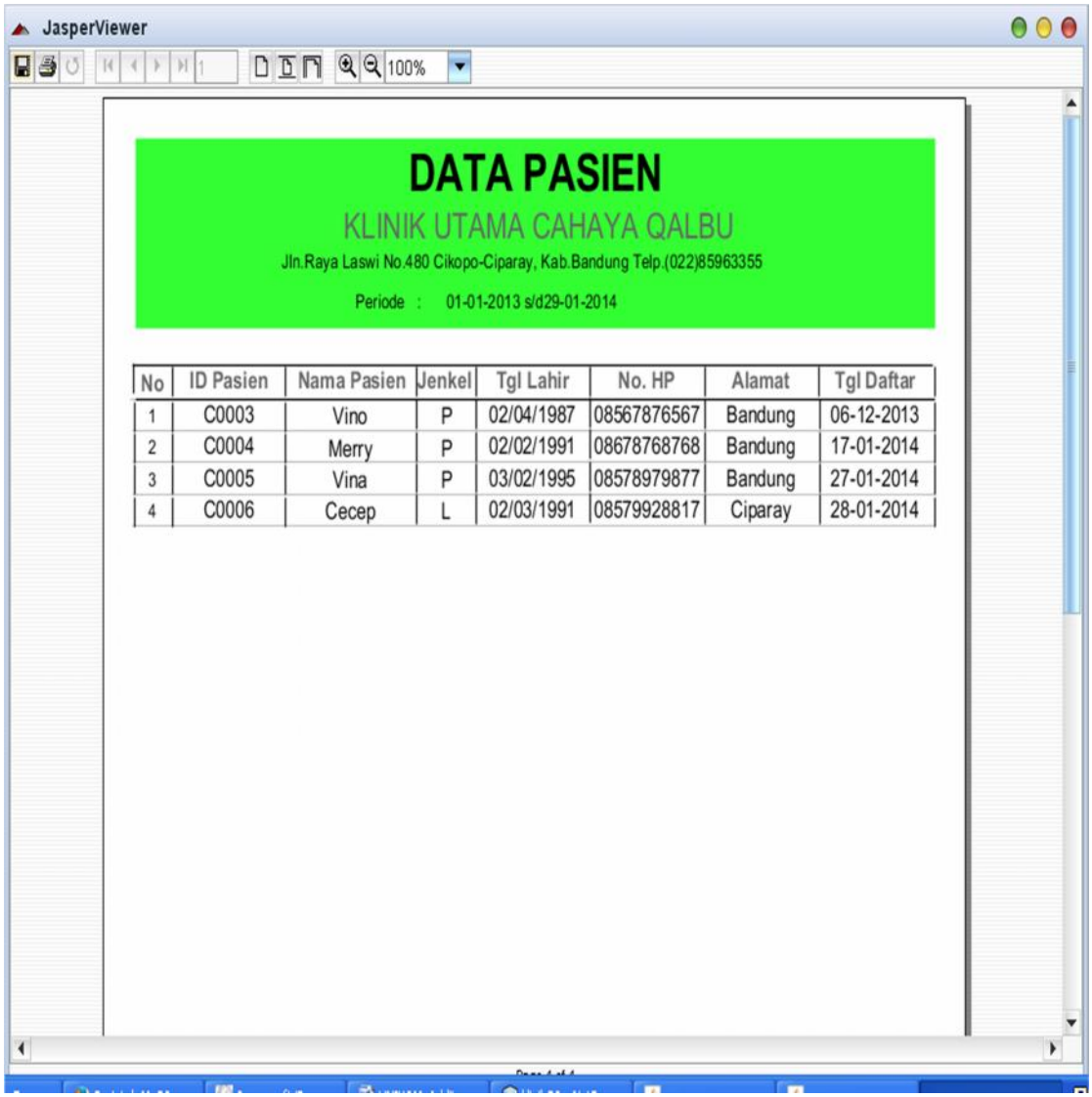
Tampilan Karu pasien ini akan muncul ketika pada Form Pendaftaran tadi mengklik *button* Cetak Kartu, Kartu ini digunakan untuk melanjutkan ke proses data antrian dan pemberian nomor antri, terdiri dari Tanggal Daftar, Nama Pasien, Jenis Kelamin, Tgl Lahir, Alamat, Nmr.HP, dan ID Pasien Klik Tombol Print di kiri atas layar untuk mencetak kartu pasien.




**Gambar 5.26** Cetak Kartu Pasien

11. Tampilan Cetak Data Pasien

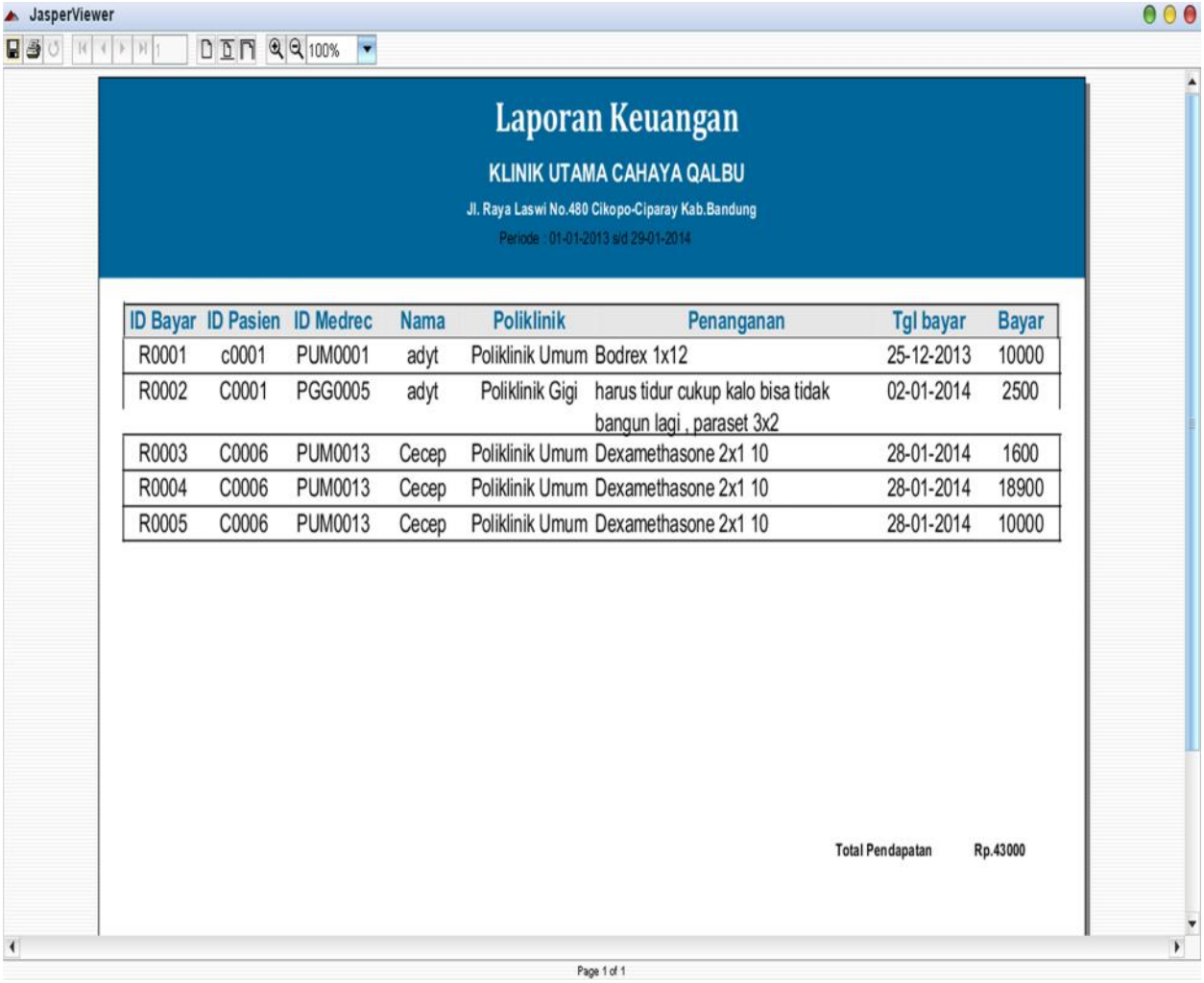
Laporan data pasien ini akan muncul dengan mengklik menu Laporan Pada Halaman Utama kemudian Klik Submenu Data Laporan, kemudian masukan periode yang diinginkan terdiri dari kolom No, ID Pasien, Nama, Jenis Kelamin, Umur/Tgl Lahir, Alamat, dan Nmr Hp Klik *Save* untuk menyimpan dan Klik Print untuk mencetak Laporan data pasien ini.




**Gambar 5.27** Cetak Daftar Pasien

12. Tampilan Cetak Laporan Keuangan

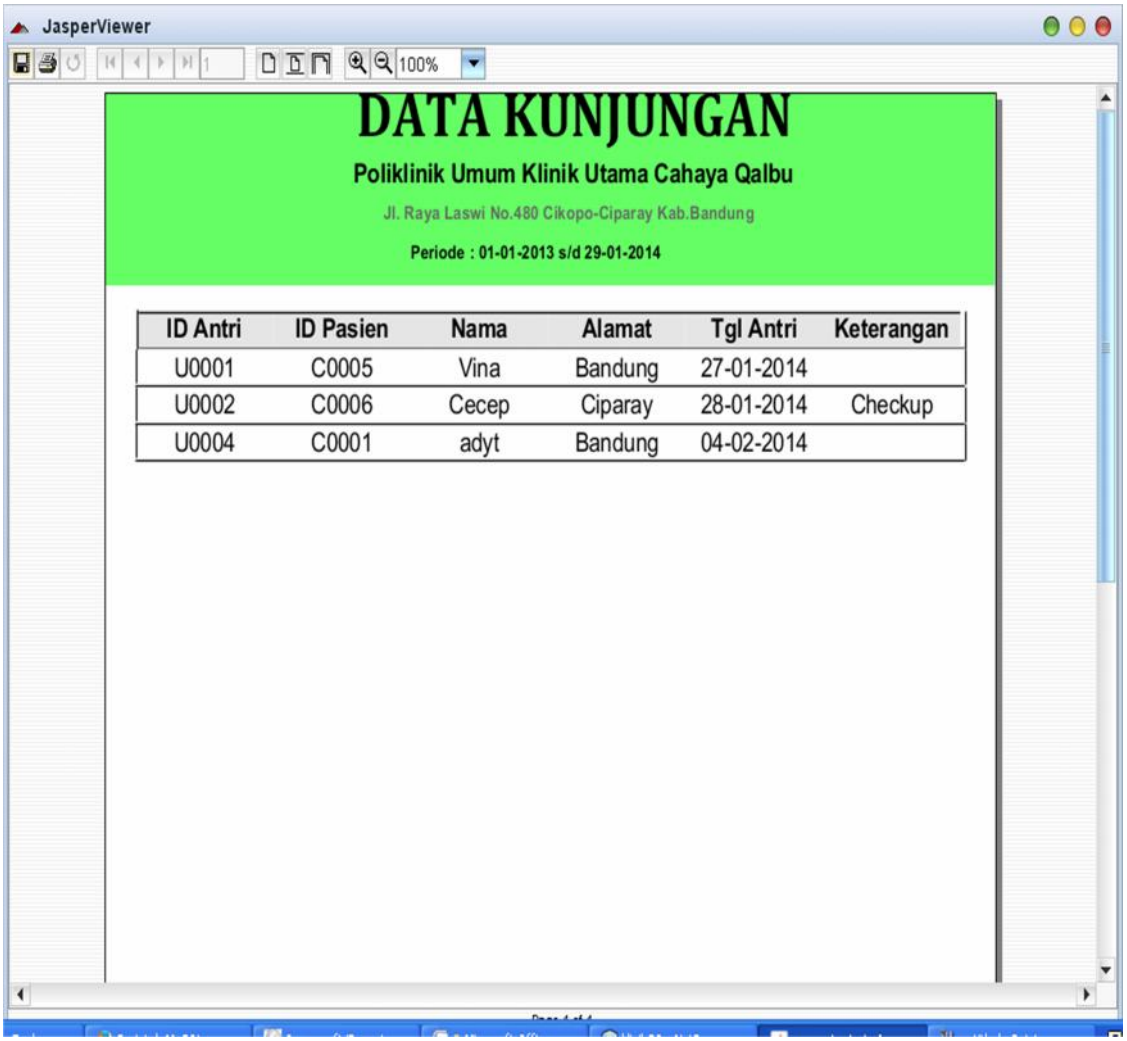
Laporan Data Lab ini akan bisa dimunculkan ketika mengakses Form Laporan setelah menginput tanggal awal dan akhir periode yang diinginkan, Laporan ini terdiri dari Kolom ID Bayar, ID Pasien, ID Medrec, Nama, Poliklinik, Penanganan, Tgl Bayar dan Bayar Klik *save* untuk menyimpan dan Print untuk mencetak Laporan Keuangan ini.




**Gambar 5.28** Cetak Laporan Keuangan

13. Tampilan Cetak Laporan Data Kunjungan

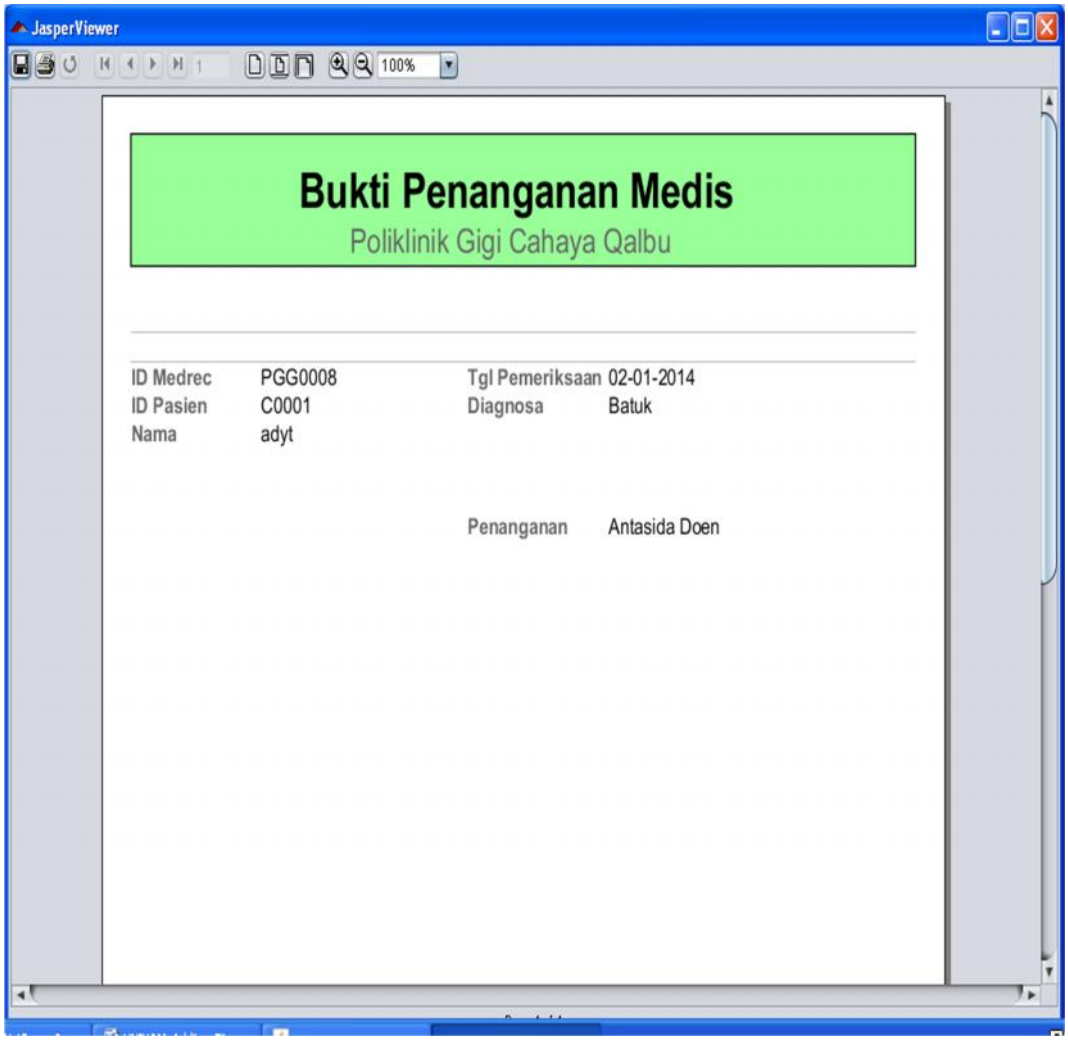
Laporan Data Kunjungan ini akan bisa dimunculkan ketika mengakses Form Laporan setelah menginput tanggal awal dan akhir periode yang diinginkan, Laporan ini terdiri dari Kolom ID Antri, ID Pasien, Nama, Alamat, Tgl Antri dan Keterangan Klik *save* untuk menyimpan dan Print untuk mencetak Laporan Data Kunjungan ini.




**Gambar 5.29** Cetak Laporan Data Kunjungan

14. Tampilan Cetak Data Pemeriksaan

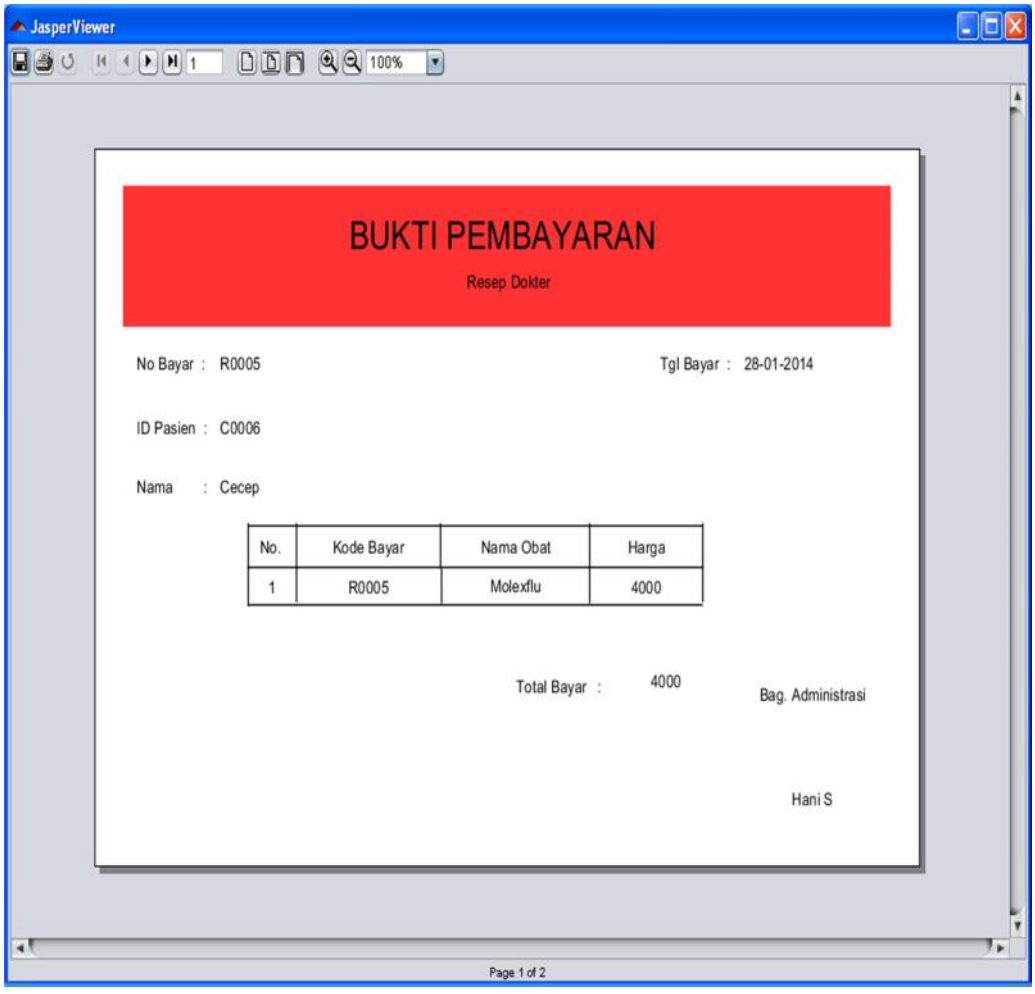
Laporan Data Pemeriksaan ini akan bisa dimunculkan ketika mengakses Form Laporan Pememeriksaan baik Poli Gigi maupun Umum. Setelah menginput data pemeriksaan kemudian melakukan pencetakan dengan menekan tombol cetak hasilnya terdiri dari ID Medrec, ID Pasien, Nama, Tgl pemeriksaan, Diagnosa dan Penanganan Klik *save* untuk menyimpan dan Print untuk mencetak Data Pemeriksaan ini.




**Gambar 5.30** Cetak Form Pemeriksaan

15. Tampilan Cetak Bukti Bayar

Bukti Bayar ini akan ditampilkan ketika pada Form Pembayaran baik Pembayaran Untuk Poli Gigi Maupun Poli Umum dengan mengklik Cetak, terdiri dari ID Bayar, ID Pasien,ID Medrec, Nama, Poliklinik, Tgl Bayar, Penanganan dan Total Bayar, Klik Print pada kiri atas layar untuk mencetak bukti bayar.




**Gambar 5.31** Cetak Bukti Bayar

## 5.2. Pengujian

Pengujian perangkat lunak adalah elemen kritis dari jaminan kualitas perangkat lunak dan merepresentasikan spesifikasi, desain dan pengkodean. Meningkatnya visibilitas (kemampuan) perangkat lunak sebagai suatu elemen sistem dan “biaya” yang muncul akibat kegagalan perangkat lunak, memotivasi dilakukannya perencanaan yang baik melalui pengujian yang teliti. Pada dasarnya, pengujian merupakan satu langkah dalam proses rekayasa perangkat lunak yang dapat dianggap sebagai hal yang merusak daripada membangun.

Dalam 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 karena jika kita salah menerapkannya pengujian tidak akan efektif.
2. Tahapan pengujian yang merupakan langkah-langkah dalam melakukan pengujian.

### 5.2.1. Rencana Pengujian

Berikut ini adalah rencana pengujian sistem informasi pengolahan data pasien :

**Tabel 5.10** Rencana Pengujian SI Pengolahan data Pasien

| Kelas Uji | Butir Uji | Tingkat Pengujian | Jenis Pengujian |
|---|---|---|---|
| Login | Pengecekan Login | Sistem | *Black Box* |
| Pengisian Data | Pengisian data pasien | Modul | *Black Box* |
| | Pencarian data pasien | 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 :

1. Pengecekan login admin

Berikut ini kasus dan hasil pengujian login user:

**Tabel 5.11** Pengecekan Login

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Username : admin<br>Bagian : admin<br>Password : admin | Data Input valid pada textbox username, label bagian, dan textbox pasword | mengisi login sesuai yang diharapkan | [X] Diterima<br>[ ] Ditolak |
| Klik tombol Login | Login berhasil dengan hak | Tombol dapat berfungsi | [X] Diterima |

| | | | |
|---|---|---|---|
| | aksesnya dan menampilkan pesan “Login Berhasil ” | sesuai yang diharapkan | [ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Username : admundo<br>Bagian :<br>-<br>Password : admindi | Tercantum pada textbox username, label bagian, dan textbox password | User tidak berhasil login | [X] Diterima<br><br>[ ] Ditolak |
| Klik tombol Login | Tidak dapat masuk ke menu utama dan menampilkan pesan “Ada kesalahan pada username dan password !” | Tombol dapat berfungsi sesuai yang diharapkan | [X] Diterima<br><br>[ ] Ditolak |

2. Pengisian data pendaftaran pasien

Berikut ini kasus dan hasil uji pengujian *input* data pasien :

**Tabel 5.12** Pengisian Data Pasien

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Klik tombol Tambah | Tombol Simpan, Edit aktif dan ID Pasien otomatis muncul sesuai urutan yang diharapkan | Dapat melakukan pengisian data sesuai yang diharapkan | [X] Diterima<br><br>[ ] Ditolak |
| Data-data pasien diisi secara lengkap | Penyimpanan data kedalam file pasien | Dapat melakukan penyimpanan | [X] Diterima<br><br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| | | data dengan benar | |
| Klik tombol Simpan | *Field – field input* kembali kosong dan data tersimpan di file data pasien | Dapat melakukan penyimpanan data dengan benar, muncul pesan "*Input* Berhasil" | [X] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Data-data barang diisi secara tidak lengkap dan ada yang kosong | Penyimpanan tidak dapat dilakukan | Tidak dapat melakukan penyimpanan data | [X] Diterima<br>[ ] Ditolak |
| Klik tombol Simpan | Penyimpanan tidak dapat dilakukan | Penyimpanan gagal dan tampil pesan "Silahkan ulangi *input* data, masih terdapat data yang kosong" | |

3. Pencarian data pasien

Berikut ini kasus dan hasil uji pencarian data pasien :

**Tabel 5.13** Pencarian Data Pasien

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Pilih pencarian berdasarkan yang diinginkan,isi sesuai data yang dicari | Pencarian data kedalam file data pasien | Dapat melakukan pencarian data dan hasilnya ditampilkan pada tabel | [X] Diterima<br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| Klik Tombol Cari | Data pasien bisa dicari sesuai tanggal masuk,id pasien, nama pasien,jenis kelamin,umur, alamat,nmr tlp, maupun keterangan | Dapat mencari data sesuai yang diharapkan, muncul pesan"Data ditemukan" | [X] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Data tidak diisi | Pencarian data kedalam file Data pasien | melakukan pencarian | [X] Diterima<br>[ ] Ditolak |
| Klik tombol Cari | Data pasien tidak dapat dicari | Tidak dapat mencari data, Muncul pesan "Data tidak ditemukan" | [X] Diterima<br>[ ] Ditolak |

4. Pengisian data antri pasien

Berikut ini kasus dan hasil uji pengujian data antri pasien :

**Tabel 5.14** Pengisian Data Antri Pasien

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Klik Tombol Tambah dan isi id pasien,tekan *enter* pada *field* id pasien | No antri keluar otomatis dan juga data pasien ketika menekan *enter* di *field* id pasien | Otomatisasi berjalan dengan baik | [X] Diterima<br>[ ] Ditolak |
| Klik tombol Simpan | Data tersimpan di file data antri | Dapat melakukan penyimpanan data dengan | [X] Diterima<br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| | | benar | |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Id pasien tidak diisi dan melakukan *input* manual | Penyimpanan tidak valid | Melakukan penyimpanan yang dapat mengacaukan data pasien | [X] Diterima<br>[ ] Ditolak |
| Klik tombol Simpan | Penyimpanan dapat dilakukan tetapi data tidak valid | Penyimpanan Berhasil tetapi data tidak valid | [X] Diterima<br>[ ] Ditolak |

5. Pengisian data rekam medis pasien

Berikut ini kasus dan hasil uji pengujian data rekam medis pasien :

**Tabel 5.15** Pengisian Data Rekam Medis Pasien

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Klik Tombol Tambah dan isi id pasien,tekan *enter* pada *field* id pasien | Data Riwayat medis pasien keluar otomatis dan juga data pasien ketika menekan *enter* di *field* id pasien | Otomatisasi berjalan dengan baik | [X] Diterima<br>[ ] Ditolak |
| Klik tombol Simpan | Data tersimpan di *database* rekam medis pasien | Dapat melakukan penyimpanan data dengan benar | [X] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Id pasien tidak diisi atau mencoba melakukan | Penyimpanan tidak valid , *field* masih disable | Tidak dapat Melakukan penyimpanan yang dapat | [X] Diterima<br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| *input* manual | | mengacaukan data pasien | |
| Klik tombol Simpan | Penyimpanan dapat dilakukan tetapi data tidak valid | Penyimpanan Berhasil tetapi data tidak valid | [X] Diterima<br>[ ] Ditolak |

6. Pengisian data bayar obat

Berikut ini kasus dan hasil uji pengujian data bayar obat :

**Tabel 5.16** Pengisian Data Bayar Obat

| **Kasus dan Hasil Uji (Data Normal)** | | | |
|---|---|---|---|
| **Data Masukan** | **Yang diharapkan** | **Pengamatan** | **Kesimpulan** |
| Klik Tombol Tambah dan isi id medrec,tekan *enter* pada *field* id medrec | Data Medrec keluar otomatis dan juga data pasien ketika menekan *enter* di *field* id medrec sehingga bag admin dapat melihat obat yang dibutuhkan dan melakukan kalkulasi | Otomatisasi berjalan dengan baik | [X] Diterima<br>[ ] Ditolak |
| Klik tombol Simpan | Data tersimpan di *database* pembayaran obat | Dapat melakukan penyimpanan data dengan benar | [X] Diterima<br>[ ] Ditolak |
| **Kasus dan Hasil Uji (Data Salah)** | | | |
| Id medrec tidak diisi dan melakukan *input* manual | Penyimpanan tidak valid | Melakukan penyimpanan yang dapat mengacaukan data medrec | [ ] Diterima<br>[X] Ditolak |
| Klik tombol Simpan | Penyimpanan dapat dilakukan tetapi data tidak | Penyimpanan Berhasil tetapi data tidak valid | [X] Diterima<br>[ ] Ditolak |

| | | | |
|---|---|---|---|
| | valid | | |

### 5.2.3. Kesimpulan Hasil Pengujian

Berdasarkan hasil pengujian dengan beberapa kondisi yang memungkinkan pada saat program digunakan maka dapat disimpulkan perangkat lunak yang dibuat sudah berjalan cukup baik berdasarkan fungsi yang diharapkan.