

BAB IV IMPLEMENTASI DAN PEMBAHASAN

4.1. Implementasi

1. Pengujian Sistem

Pengetesan sistem berfungsi untuk memastikan bahwa elemen-elemen atau komponen-komponen dari sistem telah berfungsi dengan baik dan sesuai dengan apa yang diharapkan. Pengetesan perlu dilakukan untuk mencari kesalahan atau kelemahan yang mungkin masih terjadi di dalam sistem. Pengetesan sistem merupakan pengetesan program secara keseluruhan. Hasil yang diperoleh :

Sistem telah berjalan dengan baik sesuai yang diharapkan.

2. Pengujiab Program

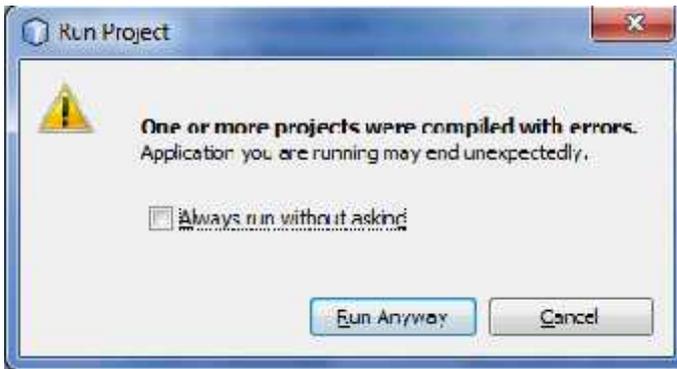
1. Kesalahan kode program (*syntax error*)

Kesalahan ini terjadi jika kode program yang ditulis tidak sesuai dengan prosedur penulisannya. Contoh sebagai berikut :

```
234  
235 private void btnUbahActionPerformed(java.awt.event.ActionEvent evt) {  
236     try(  
237         if (tidakBolehKosong()) {  
238             Admin a = new Admin();  
239             a.setIdAdmin(lblID.getText()  
240             a.setUsername(txtUser.getText());  
241             a.setPassword(txtPass.getText());  
242             daoAdmin.update(a);  
243             kondisiAwal();
```

Gambar 4.1 Kesalahan Kode Pemrograman

Kesalahan terjadi pada bagian yang terdapat tanda merah, yakni pada baris `a.setIdAdmin(lblID.getText())`, penulisan yang seharusnya dilakukan adalah `a.setIdAdmin(lblID.getText());`, karena penulisan kurang tanda ; sehingga program tidak bisa berjalan dan mendapatkan pesan kesalahan sebagai berikut :



Gambar 4.2 Pesan Kesalahan Kode Program

Setelah dilakukan pengetesan program berdasarkan kesalahan kode maka kesalahan seperti diatas sudah tidak terjadi.

2. Kesalahan proses (*run time error*)

Kesalahan ini menyebabkan program berhenti sebelum waktunya selesai karena kompiler menemukan kondisi yang belum terpenuhi dan tidak layak untuk dikerjakan. Kesalahan ini juga relatif mudah ditemukan, karena juga ditunjukkan letak serta sebab kesalahan. Contoh kesalahannya adalah sebagai berikut :

```

02 private void kondisiAwal() {
03
04     bersih();
05     lblID.setText(String.valueOf(ID));
06     lblID.setVisible(false);
07     btnSimpan.setEnabled(false);
08     btnHapus.setEnabled(false);
09     btnUbah.setEnabled(false);
10     muatSemuaAdmin();
11     isiTabel();
12     validasiInputan();
13
14
15
16 }

```

Gambar 4.3 Kesalahan Proses

Program selesai sebelum waktunya karena terjadinya penulisan kesalahan pada kondisi awal form admin kesalahannya adalah kurang menambahkan `daoAdmin=new AdminDAOImpl();` . Sehingga ketika program di running akan tetap berjalan tetapi ketika user memilih sub menu admin, maka form admin tidak akan tampil kemudian akan muncul seperti ini :

```

state:
kembali berhasilException in thread "AWT-EventQueue-0" java.lang.NullPointerException
at bright.fcom.FrmAdmin.muatSemuaAdmin(FrmAdmin.java:33)
at bright.fcom.FrmAdmin.kondisiKawal(FrmAdmin.java:31)
at bright.fcom.FrmAdmin.<init>(FrmAdmin.java:42)
at bright.fcom.MenuUtama.menjalankanActionPerformed(MenuUtama.java:58)
at bright.fcom.MenuUtama.access$800(MenuUtama.java:38)
at bright.fcom.MenuUtama$8.actionPerformed(MenuUtama.java:215)
at javax.swing.AbstractButton.fireActionPerformed(AbstractButton.java:2018)
at javax.swing.AbstractButton$Handler.actionPerformed(AbstractButton.java:2341)
at javax.swing.DefaultButtonModel.fireActionPerformed(DefaultButtonModel.java:402)
at javax.swing.DefaultButtonModel.setPressed(DefaultButtonModel.java:253)
at javax.swing.AbstractButton.doClick(AbstractButton.java:374)
at javax.swing.plaf.basic.BasicMenuItemUI.doClick(BasicMenuItemUI.java:337)

```

Gambar 4.4 Pesan Kesalahan Proses Running

Setelah dilakukan pengetesan program berdasarkan kesalahan proses maka kesalahan seperti diatas sudah tidak terjadi.

3. Kesalahan logika (*logical error*)

Kesalahan ini terjadi pada logika program yang dibuat. Kesalahan ini sulit ditemukan, karena tidak ada pemberitahuan mengenai kesalahan dan tetap akan didapatkan hasil proses program, tetapi hasilnya salah. Kesalahan ini merupakan kesalahan yang berbahaya, hasil yang salah dapat menyesatkan bagi yang menggunakannya.

```

732
733 private void btnProsesActionPerformed(java.awt.event.ActionEvent evt) {
734
735     if(!txtBayar.getText().isEmpty()){
736         int total, bayar, kembali;
737         total=Integer.parseInt(lblTotalSemua.getText());
738         bayar=Integer.parseInt(txtBayar.getText());
739         kembali=bayar+total;
740         txtKembali.setText(String.valueOf(kembali));
741     }
742 }

```

Gambar 4.5 Kesalahan Logika

Pada proses perhitungan transaksi, seharusnya untuk menghitung kembali adalah bayar dikurangi total, tetapi pada gambar diatas proses tersebut ditambahkan, saat proses tersebut dijalankan maka tetap akan didapatkan hasil tetapi hasilnya salah. Setelah dilakukan pengetesan program berdasarkan kesalahan logika maka kesalahan seperti diatas sudah tidak terjadi.

4.2. Rancangan Antarmuka

Rancangan antarmuka merupakan tampilan berbagai form yang terdapat pada sistem yang telah dikembangkan. Berikut ini merupakan uraian dari rancangan-rancangan antarmuka yang terdapat pada sistem.

1. Tampilan Form Login

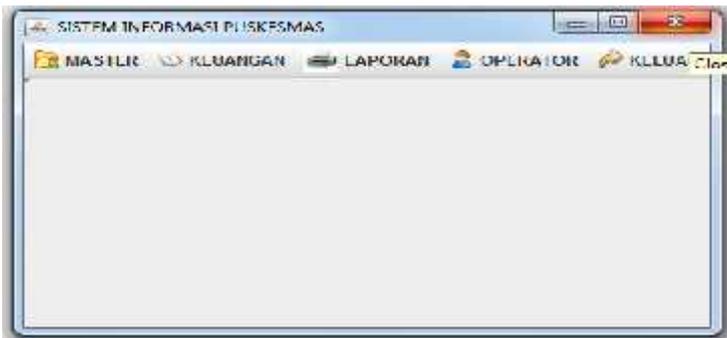
Untuk dapat mengakses sistem user harus mengisi username dan password. Apabila username dan password yang dimasukkan semua pengguna hak akses benar maka pengguna hak akses bisa menggunakan aplikasi tersebut. Jika username dan password salah maka pengguna harus mengisi kembali username dan password yang benar. Rancangan form login ini dapat dilihat pada gambar 4.6 dibawah ini.



Gambar 4.6 Form Login

2. Tampilan Form Utama

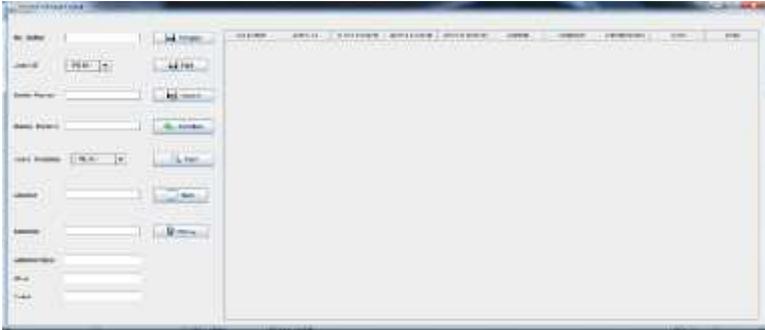
Form menu halaman utama ini merupakan menu aplikasi yang muncul setelah user berhasil melakukan proses login pada aplikasi ini, pengguna aplikasi dapat memilih berbagai macam proses pengolahan data yang dibutuhkan sesuai dengan kebutuhan. Rancangan form menu halaman utama ini dapat dilihat pada gambar 4.7 dibawah ini.



Gambar 4.7 Form Utama

3. Halaman Form Pendaftaran

Form Pendaftaran hanya dapat diakses oleh admin, form pendaftaran digunakan untuk menyimpan dan menghapus data pasien. Apabila akan menyimpan data pasien yang baru dan cukup menekan tombol simpan setelah data lengkap diisi. Apabila data ingin menghapus, cukup klik cari dengan memasukkan nama pasien maka data yang diinginkan akan ditampilkan, tekan tombol hapus jika data tersebut ingin dihapus. Rancangan form pendaftaran ini dapat dilihat pada gambar 4.8 dibawah ini.



Gambar 4.8 Form Pendaftaran

4. Halaman Form Pasien

Form Pasien digunakan oleh admin untuk mengelola data pasien. Terdapat beberapa perintah, simpan untuk menyimpan data pasien baru, cari untuk mencari data pasien yang sudah tersimpan sebelumnya, memperbaiki data pasien lama, hapus untuk menghapus data pasien lama, refresh data pasien untuk menampilkan data keseluruhan yang ada ditabel pasien, dan kembali untuk menutup form. Form pasien ini dapat dilihat pada gambar 14 dibawah ini.



Gambar 4.9 Form Pasien

5. Halaman Form Dokter

Form Dokter digunakan oleh admin untuk mengelola data dokter. Terdapat beberapa perintah, simpan untuk menyimpan data dokter baru, cari untuk mencari data dokter yang sudah tersimpan sebelumnya, memperbaiki data dokter lama, hapus untuk menghapus data dokter lama, refresh data dokter untuk menampilkan data keseluruhan yang ada ditabel dokter dan kembali untuk menutup form. Rancangan form dokter ini dapat dilihat pada gambar 14 dibawah ini.

The screenshot shows a web browser window with a form titled 'Form Dokter'. The form is organized into two columns. The left column contains several input fields: 'Nama Dokter', 'NPM', 'NIDN', 'NPI', 'Jenis Kelamin', 'Alamat', 'No. HP', and 'No. Medis'. Each field has a corresponding button to its right: 'Cari' (with a magnifying glass icon), 'Simpan' (with a floppy disk icon), 'Hapus' (with a trash can icon), 'Refresh' (with a circular arrow icon), and 'Kembali' (with a left-pointing arrow icon). The right column is currently empty, representing a table where the data would be displayed. The browser's address bar shows a URL starting with 'http://localhost:8080/...'. The window title is 'Form Dokter'.

Gambar 4.10 Form Dokter

6. Halaman Form Periksa

Form Pemeriksaan digunakan oleh admin untuk input data periksa pasien. Terdapat beberapa perintah, simpan untuk menyimpan, hapus, edit untuk edit data periksa, cari untuk mencari data eriksa pasien tertentu, refresh data tindakan medis, dan kembali untuk menutup form.. form periksa ini dapat dilihat pada gambar 4.11 dibawah ini.

The screenshot shows a web browser window with a form titled 'Form Periksa'. The form is organized into two columns. The left column contains several input fields: 'No. Pendaftaran', 'No. Rawat', 'Kategori Penyakit', 'Nama Pasien', 'Nama Dokter', 'No. Rawat', 'Etiologi/Gejala', 'Riwayat Penyakit', and 'Mata (Mata)'. Each field has a corresponding button to its right: 'Cari' (with a magnifying glass icon), 'Simpan' (with a floppy disk icon), 'Hapus' (with a trash can icon), 'Refresh' (with a circular arrow icon), and 'Kembali' (with a left-pointing arrow icon). The right column is currently empty, representing a table where the data would be displayed. The browser's address bar shows a URL starting with 'http://localhost:8080/...'. The window title is 'Form Periksa'.

Gambar 4.11 Form Periksa

7. Halaman Form Transaksi
- Form transaksi di gunakan oleh admin untuk melihat hasil transaksi/total transaksi. Terdapat beberapa tombol perintah di antaranya Refresh untuk melihat isi table ada data yang baru, total transaksi untuk melihat total semua hasil transaksi, kembali untuk kluar dari form transaksi. Form transaksi bisa dilihat digambar 4.11 dibawah ini.



Gambar 4.12 Form Transaksi

8. Halaman Form Operator
- Form Operator digunakan untuk menambahkan admin baru. Di form Operator terdapat beberapa tombol perintah. Simpan, update, delete, refresh, kembali. Form Operator bisa dilihat digambar 4.12 dibawah ini.



Gambar 4.13 Form Operator

Halaman Ini Sengaja Dikosongkan