

BAB IV PERANCANGAN SISTEM

4.1. Gambaran Umum Sistem yang Diusulkan

Sistem yang diusulkan adalah sistem pendaftaran pasien baru untuk rumah sakit umum yang akan ditempatkan pada meja/loket petugas administrasi bagian pendaftaran. Target pengguna untuk sistem ini adalah petugas administrasi rumah sakit yang akan mendapatkan hak akses untuk melakukan input data pasien, melihat data pasien, *setting* e-KTP Reader, *setting scanner*, dan membuat laporan data pendaftaran pasien.

Sistem pendaftaran pasien ini berbasis aplikasi *desktop* yang berjalan pada sistem operasi Windows karena dibangun dalam basis bahasa pemrograman VB.Net. Media penyimpanan data yang digunakan adalah MySQL dan tersimpan dalam komputer *server*, sehingga diperlukan komputer *client* dan komputer *server* yang saling terhubung dalam jaringan LAN.

Sistem pendaftaran pasien ini dapat terhubung dengan *hardware scanner* dan e-KTP Reader sebagai alat inputan pada aplikasi. *Scanner* dan e-KTP Reader tersebut menggantikan proses *input* yang dilakukan secara manual saat ini, sehingga diharapkan sistem ini akan lebih mampu membantu petugas administrasi bagian pendaftaran dalam melakukan pekerjaannya.

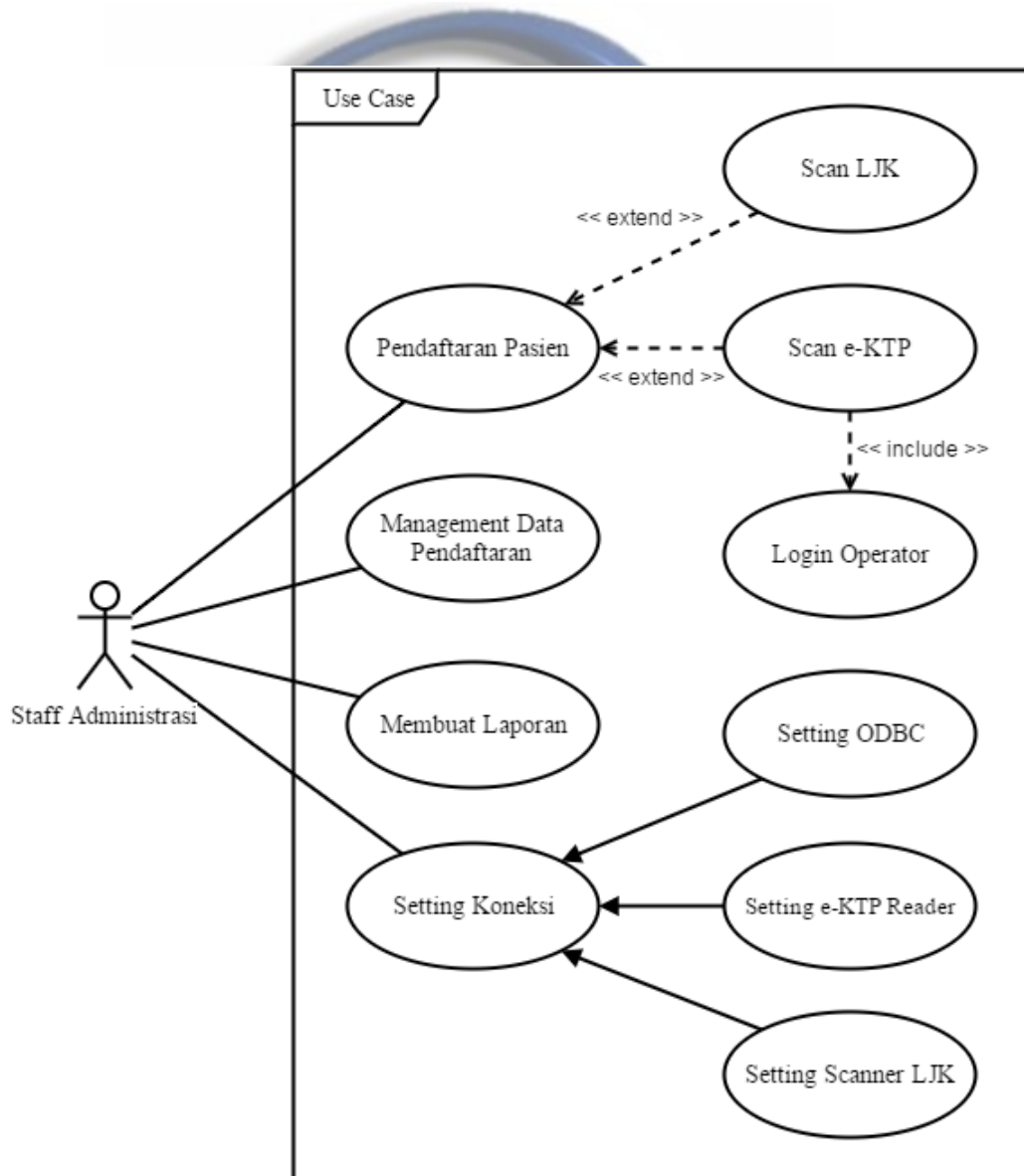
4.2. Perancangan Prosedur

Perancangan prosedur merupakan tahapan-tahapan instruksi yang harus dilakukan untuk mencapai tujuan. Perancangan prosedur merupakan hasil dari perubahan dan evaluasi sistem yang berjalan, dimana sistem yang diusulkan dapat memperbaiki kekurangan dari sistem yang sedang berjalan.

Perancangan prosedur yang diusulkan meliputi *use case diagram*, *skenario use case*, *activity diagram* dan *sequence diagram*.

4.2.1. Use Case Diagram

Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan aplikasi yang akan dibuat. *Use case* digunakan untuk mengetahui fungsi atau proses apa saja yang ada di dalam sebuah aplikasi dan siapa saja yang berhak menggunakan fungsi atau proses tersebut. Gambar 4.1 adalah *use case diagram* yang akan menggambarkan sistem pendaftaran pasien rumah sakit umum ini.



Gambar 4.1. Use Case Diagram

4.2.2. Skenario *Use Case*

Skenario *use case* akan mendeskripsikan urutan proses antara aktor dengan setiap fitur-fitur sistem sesuai dengan gambaran dari *use case diagram*. Berikut adalah skenario *use case* tersebut:

Tabel 4.1. Skenario *Use Case* Pendaftaran Pasien

Nama <i>Use-Case</i>	Pendaftaran Pasien
Aktor	Staff Administrasi Bag. Pendaftaran
Deskripsi	<i>Input data</i> pasien baru untuk disimpan ke <i>database</i> pada <i>server</i> .
<i>Normal Course</i>	<ol style="list-style-type: none"> 1. Aktor melakukan <i>scanning</i> e-KTP pasien pada e-KTP <i>Reader</i>. 2. Aktor melakukan <i>scanning</i> e-KTP penanggung jawab pada e-KTP <i>Reader</i>. 3. Aktor melakukan <i>scanning</i> form kuisisioner pendaftaran. 4. Sistem menyimpan data pasien dan penanggung jawabnya ke <i>database</i> pada <i>server</i>.
<i>Alternate Course</i>	<ol style="list-style-type: none"> 1.a. Pasien tidak memiliki e-KTP atau e-KTP tidak terbaca. Aktor melakukan input data secara manual. 2.a. Penganggung jawab tidak memiliki e-KTP atau e-KTP tidak terbaca. Aktor melakukan input data secara manual. 3.a. <i>Form</i> kuisisioner tidak terbaca. Aktor melakukan input data secara manual.
<i>Pre-Condition</i>	<ol style="list-style-type: none"> g. Aktor menerima <i>form</i> kuisisioner pendaftaran yang sudah terisi, e-KTP pasien, dan e-KTP penanggung jawab. h. Pasien belum terdaftar.
<i>Post-Condition</i>	Data pasien sudah terdaftar pada <i>server</i> .

Tabel 4.2. Skenario *Use Case* Setting e-KTP Reader

Nama <i>Use-Case</i>	<i>Setting e-KTP Reader</i>
Aktor	Staff Administrasi Bag. Pendaftaran
Deskripsi	Mengkonfigurasi sistem untuk dapat terhubung dengan e-KTP Reader.
<i>Normal Course</i>	<ol style="list-style-type: none"> 1. Aktor memilih port aktif yang dapat digunakan untuk membaca e-KTP Reader. 2. Aktor melakukan tes koneksi. 3. Sistem menghubungkan ke e-KTP Reader. 4. Sistem menyimpan data konfigurasi.
<i>Alternate Course</i>	-
<i>Pre-Condition</i>	<ol style="list-style-type: none"> a. E-KTP Reader sudah terhubung dengan port komputer. b. E-KTP Reader belum terhubung dengan sistem.
<i>Post-Condition</i>	E-KTP Reader terhubung dengan sistem, dan sistem dapat membaca data dari e-KTP.

Tabel 4.3. Skenario *Use Case* Setting Scanner

Nama <i>Use-Case</i>	<i>Setting Scanner</i>
Aktor	Staff Administrasi Bag. Pendaftaran
Deskripsi	Mengkonfigurasi sistem untuk dapat terhubung dengan scanner.
<i>Normal Course</i>	<ol style="list-style-type: none"> 1. Aktor mengisi nilai <i>delta intensity & threshold</i> untuk proses OMR. 2. Aktor melakukan tes scan. 3. Sistem mengunduh data dari Scanner. 4. Jika <i>form</i> belum bisa dibaca, kembali ke poin 1 5. Sitem menyimpan data konfigurasi.
<i>Alternate Course</i>	-

<i>Pre-Condition</i>	<ul style="list-style-type: none"> a. <i>Scanner</i> sudah terhubung dengan komputer. b. <i>Form</i> kuisisioner belum dapat dibaca oleh sistem.
<i>Post-Condition</i>	<i>Scanner</i> terhubung dengan sistem, dan sistem dapat membaca <i>form</i> kuisisioner pendaftaran melalui <i>scanner</i> .

Tabel 4.4. Skenario *Use Case* Setting Database

Nama <i>Use-Case</i>	<i>Setting Database</i>
Aktor	Staff Administrasi Bag. Pendaftaran
Deskripsi	Mengkonfigurasi sistem untuk dapat terhubung dengan <i>database</i> pada <i>server</i> .
<i>Normal Course</i>	<ol style="list-style-type: none"> 1. Aktor mengisi data konfigurasi <i>database</i>. 2. Aktor melakukan tes koneksi. 3. Sistem menghubungkan dengan OLEDB. 4. Sistem menyimpan data konfigurasi.
<i>Alternate Course</i>	-
<i>Pre-Condition</i>	<ul style="list-style-type: none"> a. Komputer sudah terhubung dengan <i>server</i>. b. <i>Database</i> pada <i>server</i> sudah dapat terhubung melalui OLEDB. c. Sistem belum terhubung dengan <i>database</i> pada <i>server</i>.
<i>Post-Condition</i>	Sistem dapat terhubung dengan <i>database</i> pada <i>server</i> melalui koneksi OLEDB.

Tabel 4.5. Skenario *Use Case* Management Data Pendaftaran

Nama <i>Use-Case</i>	<i>Management Data Pendaftaran</i>
Aktor	Staff Administrasi Bag. Pendaftaran
Deskripsi	Mengelola data pendaftaran pasien

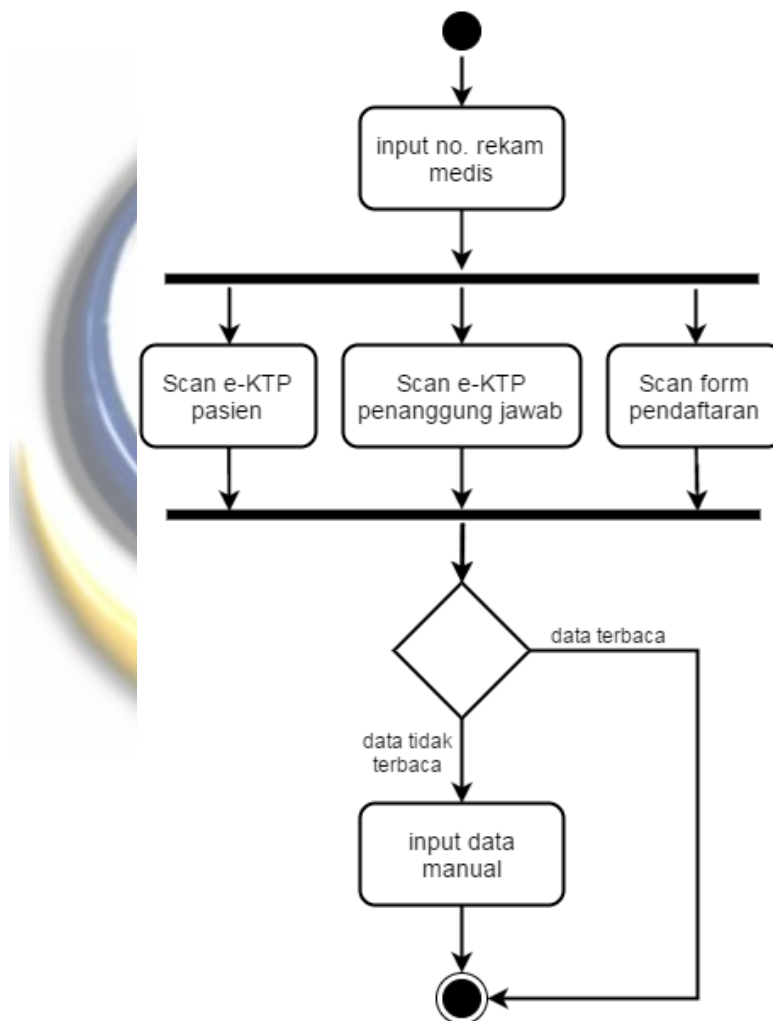
<i>Normal Course</i>	<ol style="list-style-type: none"> 1. Sistem mengambil data dari <i>database</i> pada server. 2. Sistem menampilkan data pendaftaran dalam bentuk tabel. 3. Aktor memilih salah satu baris data pendaftaran. 4. Aktor memilih <i>action</i> yang ingin dilakukan (<i>Delete/Update</i>). 5. Sistem menyimpan perubahan data pada <i>database</i>.
<i>Alternate Course</i>	-
<i>Pre-Condition</i>	Sistem telah terhubung dengan <i>server</i> .
<i>Post-Condition</i>	Perubahan-perubahan data telah tersimpan pada <i>database</i> .

Tabel 4.6. Skenario *Use Case* Membuat Laporan

Nama <i>Use-Case</i>	Membuat Laporan
Aktor	<i>Staff Administrasi Bag. Pendaftaran</i>
Deskripsi	Membuat laporan data pasien yang sudah terdaftar berdasarkan rentan waktu tertentu
<i>Normal Course</i>	<ol style="list-style-type: none"> 1. Aktor memilih tanggal awal data pendaftaran. 2. Aktor memilih tanggal akhir data pendaftaran. 3. Aktor memilih tombol buat laporan. 4. Sistem membuat laporan data pendaftaran berdasarkan rentan waktu yang sudah dipilih. 5. Aktor memilih tombol <i>print</i>. 6. Sistem mencetak laporan data pendaftaran.
<i>Alternate Course</i>	-
<i>Pre-Condition</i>	Laporan data pendaftaran pasien tidak ada
<i>Post-Condition</i>	Laporan data pendaftaran pasien telah dicetak

4.2.3. Activity Diagram

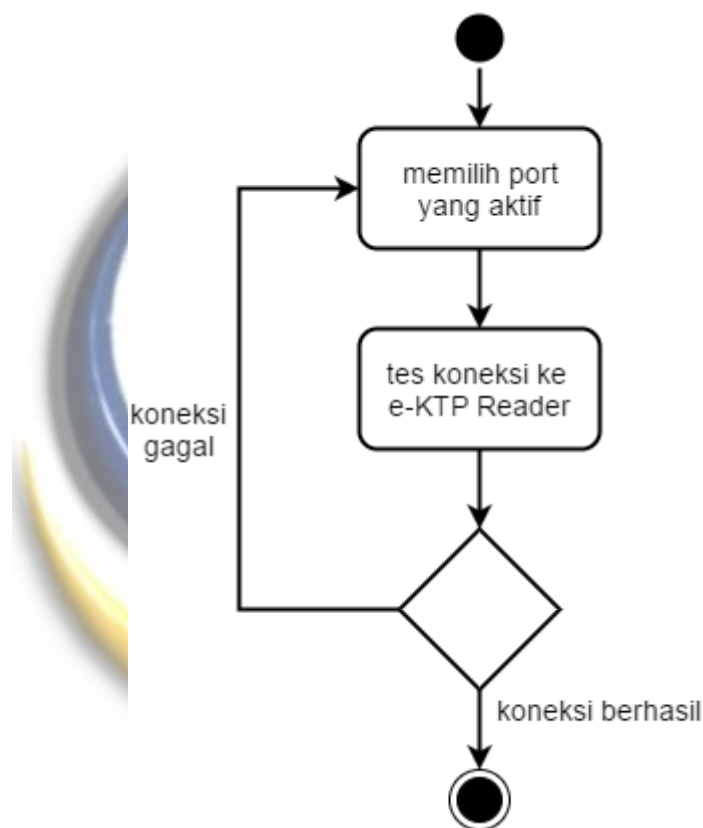
Activity diagram menggambarkan alir aktivitas dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, *decision* yang mungkin terjadi, dan bagaimana mereka berakhir. *Activity diagram* juga dapat menggambarkan proses paralel yang terjadi pada beberapa eksekusi. Berikut ini merupakan beberapa *activity diagram* yang diusulkan pada sistem pendaftaran pasien rumah sakit umum.



Gambar 4.2. Activity Diagram: Pendaftaran Pasien

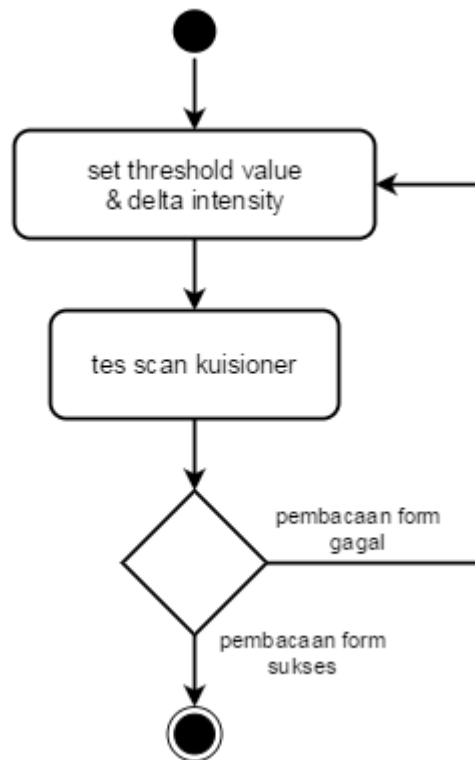
Gambar 4.2 menunjukkan alur proses pendaftaran pasien pada sistem yang usulkan. Sebelumnya, pendaftar diminta untuk mengisi lengkap *form*

kuisisioner pendaftaran, lalu menyerahkannya pada petugas beserta e-KTP pasien dan e-KTP penanggung jawab. Petugas mengisi nomor rekam medis sebagai kode unik untuk pasien tersebut. Kemudian petugas dapat melakukan *scanning* pada masing-masing e-KTP dan form pendaftaran. Jika ada salah satu atau beberapa proses *scanning* tidak dapat dilakukan/gagal, petugas dapat melakukan *input* data manual. Setelah data dianggap sudah cukup, kemudian data akan disimpan ke *database* pada *server*.



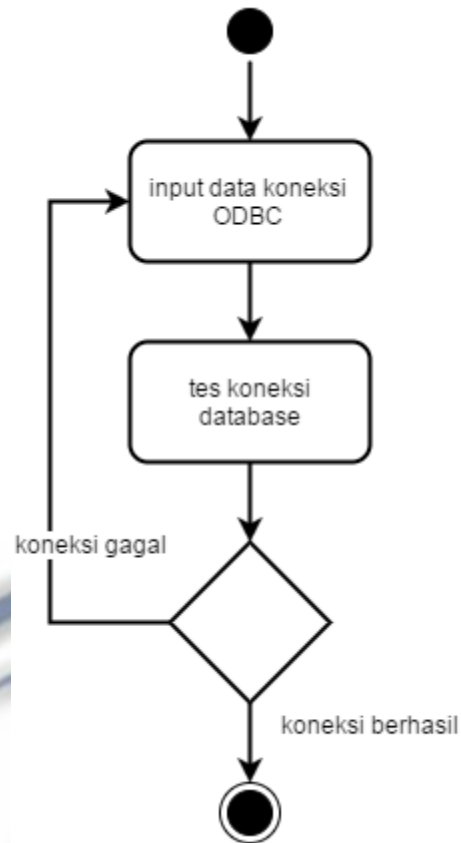
Gambar 4.3. Activity Diagram: Setting e-KTP Reader

Setting e-KTP Reader diperlukan untuk mengkonfigurasi sistem agar dapat terhubung dengan e-KTP Reader dan melakukan pengunduhan data e-KTP. Gambar 4.3 menunjukkan untuk menghubungkan sistem dengan e-KTP Reader, user harus memilih *port* yang digunakan terlebih dahulu secara manual. Jika *port* yang dipilih adalah benar *port* yang terhubung dengan e-KTP Reader, maka tes koneksi berhasil.



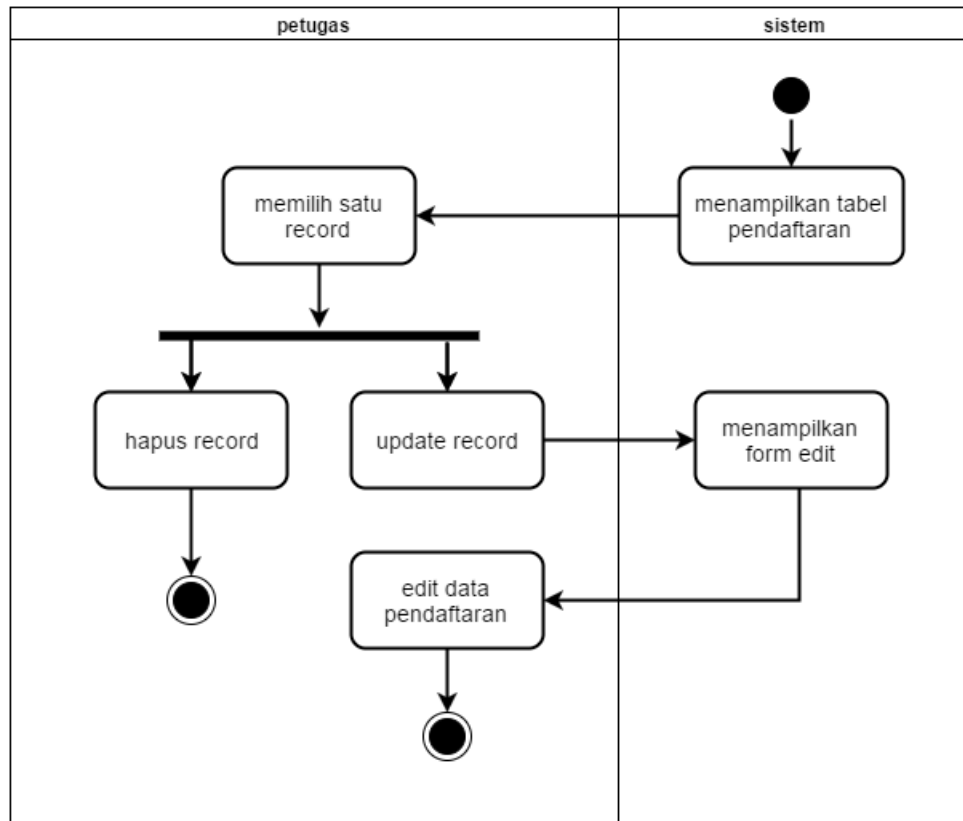
Gambar 4.4. Activity Diagram: Setting Scanner

Scanner akan digunakan untuk mengunduh data *form* pendaftaran. Karena setiap *scanner* memiliki kualitas yang beragam, maka hasil dari pengunduhan pun akan berbeda-beda, sehingga diperlukan penentuan nilai *threshold* dan *delta intensity* untuk menentukan titik balik nilai kontras dari proses OMR (Optical Mark Recognition) dengan cara *trial and error*. Gambar 4.4 menunjukkan alur proses menghubungkan *scanner* dengan sistem hingga bisa mendapatkan data yang dibutuhkan.



Gambar 4.5. Activity Diagram: Setting Database

Sebelum mengkonfigurasi *database* pada sistem, semestinya komputer sudah terhubung dengan *database* pada *server* melalui koneksi ODBC (*Open Database Connectivity*). Sehingga di dalam sistem, *user* hanya melakukan konfigurasi *database* untuk dapat terhubung dengan ODBC seperti ditunjukkan oleh gambar 4.5.



Gambar 4.6. *Activity Diagram*: Manajemen Data Pendaftaran

Manajemen data pendaftaran akan menyajikan data pasien per hari dalam bentuk tabel. *User* akan diberikan fungsi untuk menghapus atau mengubah data pasien yang sudah terdaftar bila terjadi kesalahan. Gambar 4.6 menunjukkan alur proses manajemen data yang dapat dilakukan oleh *user* pada sistem pendaftaran rumah sakit umum yang diusulkan.



Gambar 4.7. *Activity Diagram*: Membuat Laporan

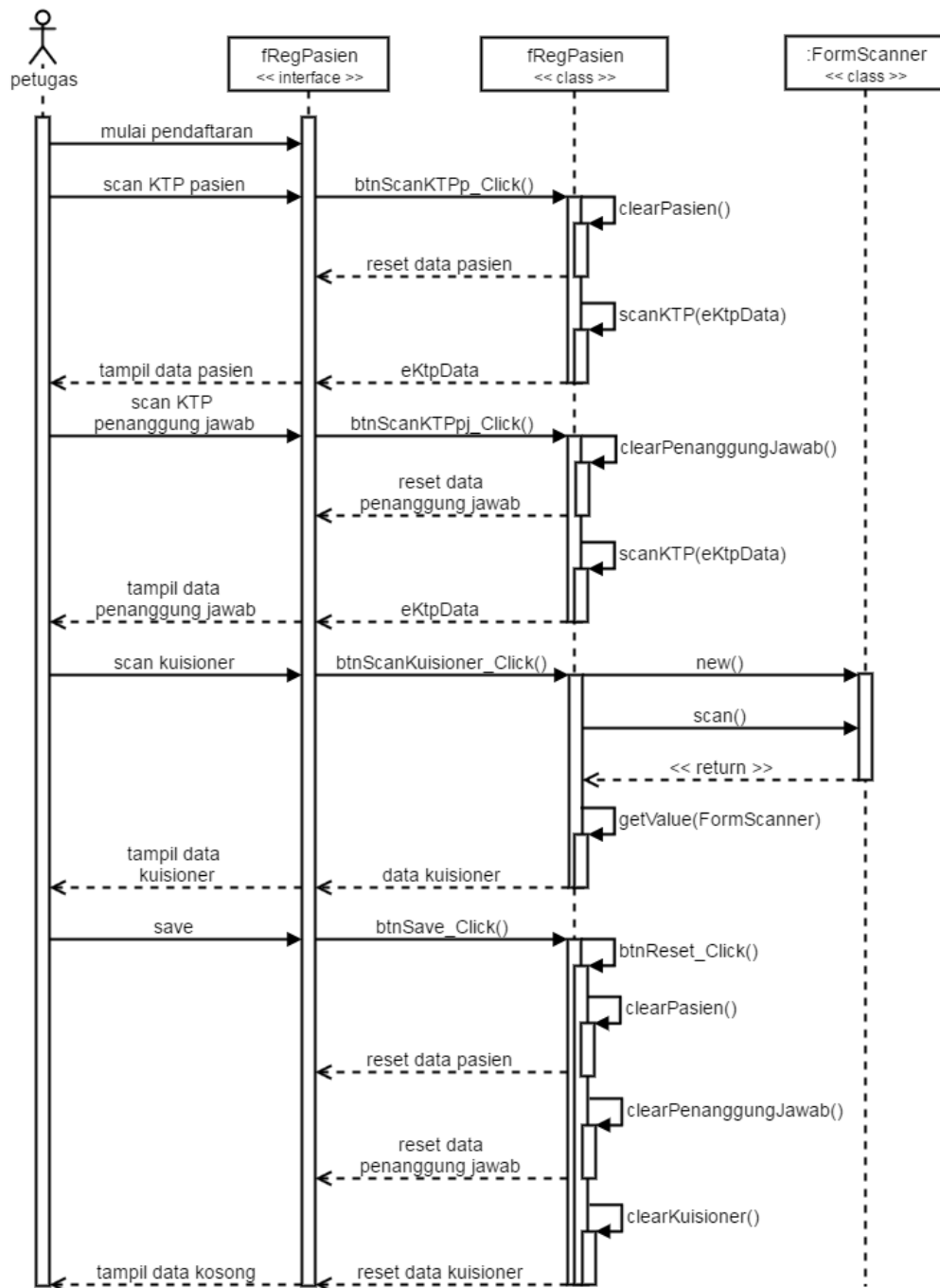
Sistem pendaftaran pasien ini dapat membuat laporan data pendaftaran pasien berdasarkan rentang tanggal yang ditentukan. Dan sebelum mencetak laporan, sistem akan menampilkan terlebih dahulu *review* laporannya. Jika sudah sesuai dengan yang diharapkan, *user* dapat memilih menu *print* untuk mencetak laporan tersebut. Gambar 4.7 menunjukkan alur proses pembuatan laporan pada sistem pendaftaran rumah sakit ini.

4.2.4. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan/perilaku objek pada *use case* dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram *sequence* maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang di instansiasi menjadi objek tersebut.

Berikut ini adalah *sequence diagram* yang diusulkan pada sistem pendaftaran pasien rumah sakit umum.

a. *Sequence Diagram Pendaftaran Pasien*



Gambar 4.8. *Sequence Diagram*: Pendaftaran Pasien

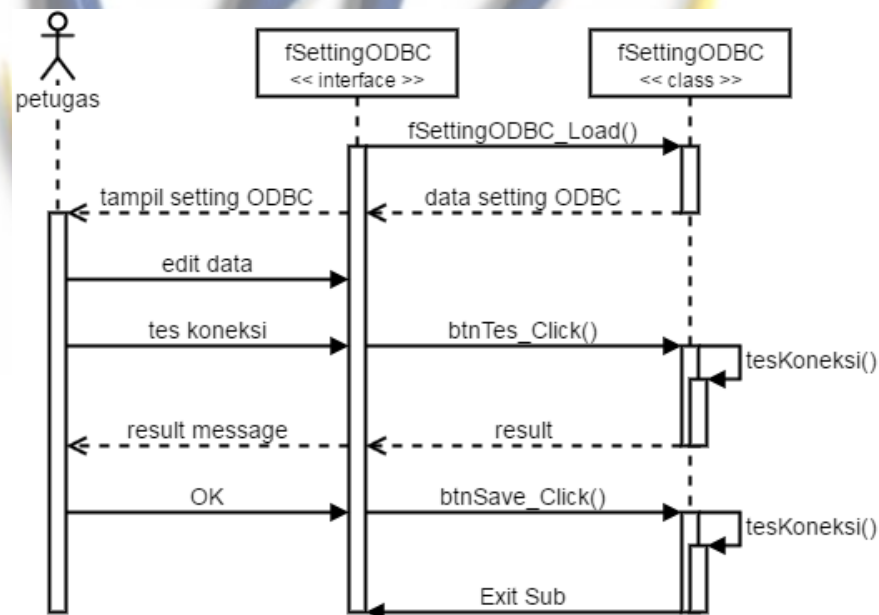
Keterangan:

Gambar 4.8 di atas adalah urutan interaksi aktor di dalam sistem yang terlibat pada *sequence diagram* pendaftaran pasien. Pada halaman awal

yaitu halaman pendaftaran pasien, aktor (petugas) akan disuguhkan fRegPasien dimana aktor harus mengisi data diri pasien dan penanggung jawabnya. Untuk mengisi data tersebut, aktor dapat melakukannya dengan cara melakukan *scanning* e-KTP milik pasien dan penanggung jawab, serta *scanning* form kuisisioner yang sudah diisi oleh pendaftar.

Untuk melakukan *scanning* e-KTP, dibutuhkan *library* libComm yang menghubungkan antara sistem dan e-KTP *Reader*. libComm membaca data di dalam e-KTP pasien dan penanggung jawab lalu mengunduh ke sistem. Sedangkan untuk mendapatkan data dari form kuisisioner dibutuhkan *library* WIA untuk menghubungkan antara sistem dengan *scanner*, dan *library* Emgu untuk mengolah data hasil *scan* yang berupa gambar menjadi data teks. Jika semua data yang dibutuhkan telah diisi, data akan disimpan melalui koneksi ODBC.

b. *Sequence Diagram Setting Database*



Gambar 4.9. *Sequence Diagram: Setting Database*

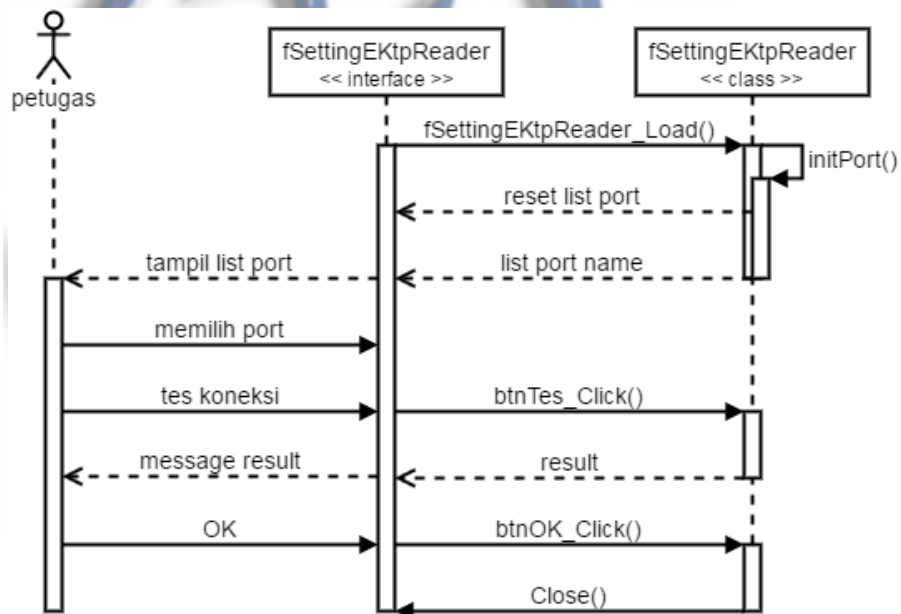
Keterangan:

Gambar 4.9 di atas adalah urutan interaksi aktor dengan sistem yang terlibat pada *sequence diagram setting database*. Interaksi terjadi pada form

Setting ODBC. *Database* dihubungkan dengan sistem melalui koneksi ODBC dan setiap konfigurasi disimpan pada *data store* *Settings* milik sistem.

Saat form *Setting* ODBC dibuka, sistem akan membaca data konfigurasi ODBC dari *Settings* dan menampilkan pada *form* tersebut. Jika aktor telah mengubah konfigurasi, aktor dapat melakukan tes koneksi lalu sistem akan mencoba menghubungkan sistem dengan *database* melalui ODBC dengan konfigurasi yang sudah diubah tersebut. Jika konfigurasi sudah lengkap dan berhasil, data akan disimpan kembali pada *Settings* saat aktor menekan tombol OK pada form.

c. *Sequence Diagram Setting e-KTP Reader*



Gambar 4.10. *Sequence Diagram: Setting e-KTP Reader*

Keterangan:

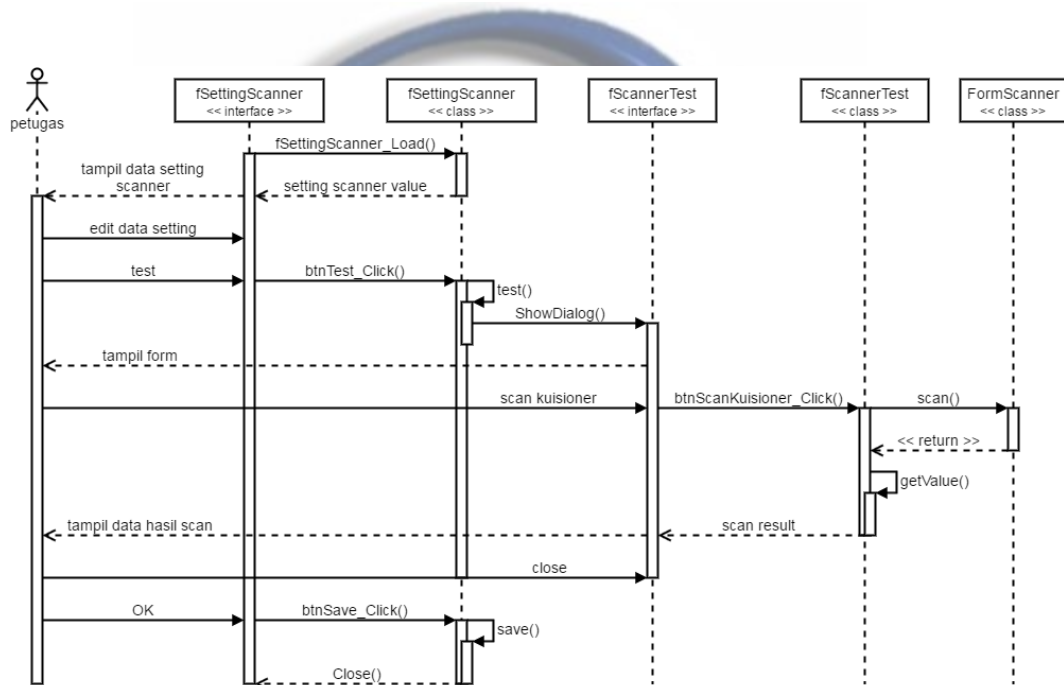
Gambar 4.10 di atas adalah urutan interaksi aktor dengan sistem yang terlibat pada *sequence diagram setting e-KTP Reader*. Interaksi terjadi pada form *Setting e-KTP Reader*.

Saat form *Setting e-KTP Reader* dibuka, sistem akan membaca nama *port* yang sudah tersimpan sebelumnya pada *Settings* dan menampilkannya

sebagai *default value*. Kemudian sistem membaca *list port* yang sedang terhubung dengan komputer dan menampilkannya pada *form* sebagai *option*.

Aktor dapat melakukan tes koneksi setelah memilih *port* yang digunakan untuk e-KTP *Reader*. Sistem lalu memberikan pesan setelah Sistem mencoba menghubungkan dengan e-KTP *Reader* dan membaca e-KTP. Jika dianggap sudah berhasil, sistem akan menyimpan perubahan data pada *Settings* saat aktor memilih OK.

d. Sequence Diagram Setting Scanner



Gambar 4.11. Sequence Diagram: Setting Scanner

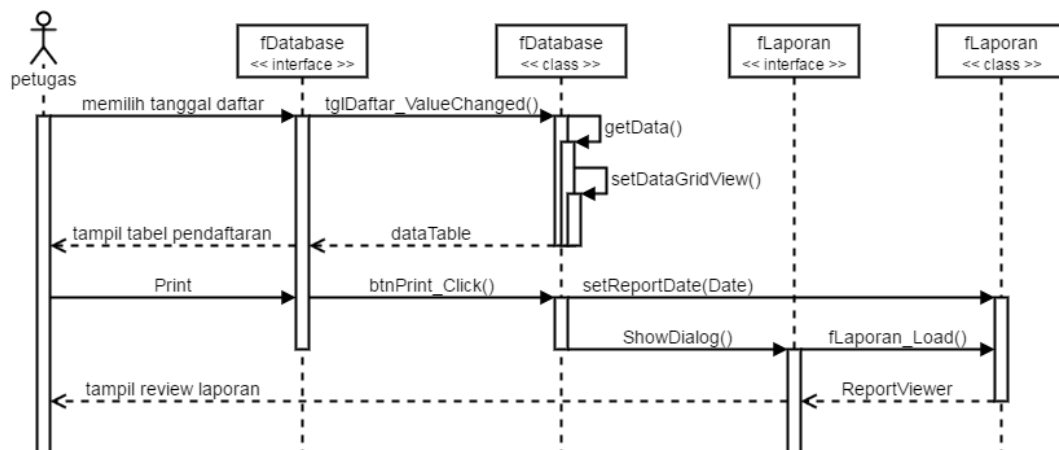
Keterangan:

Gambar 4.11 di atas adalah urutan interaksi aktor dengan sistem yang terlibat pada *sequence diagram setting scanner*. Interaksi terjadi pada form *Setting Scanner* dan form *Scanner Test*.

Saat pertama kali form *Setting Scanner* dijalankan, sistem akan mengambil data *threshold* dan *delta intensity* dari *Settings* sistem, dan menjadikannya sebagai *default value* pada form tersebut.

Aktor dapat mengganti nilai *threshold* dan *intensity* pada form untuk menyesuaikan dengan *scanner* yang dipakai. Setelah mengganti nilai tersebut, lalu aktor dapat melakukan *scanning test* untuk memastikan nilai yang dimasukkan sudah tepat, atau dapat langsung memilih tombol OK untuk menyimpan perubahan tersebut pada data *Settings* sistem.

e. *Sequence Diagram* Membuat Laporan



Gambar 4.12. *Sequence Diagram*: Membuat Laporan

Keterangan :

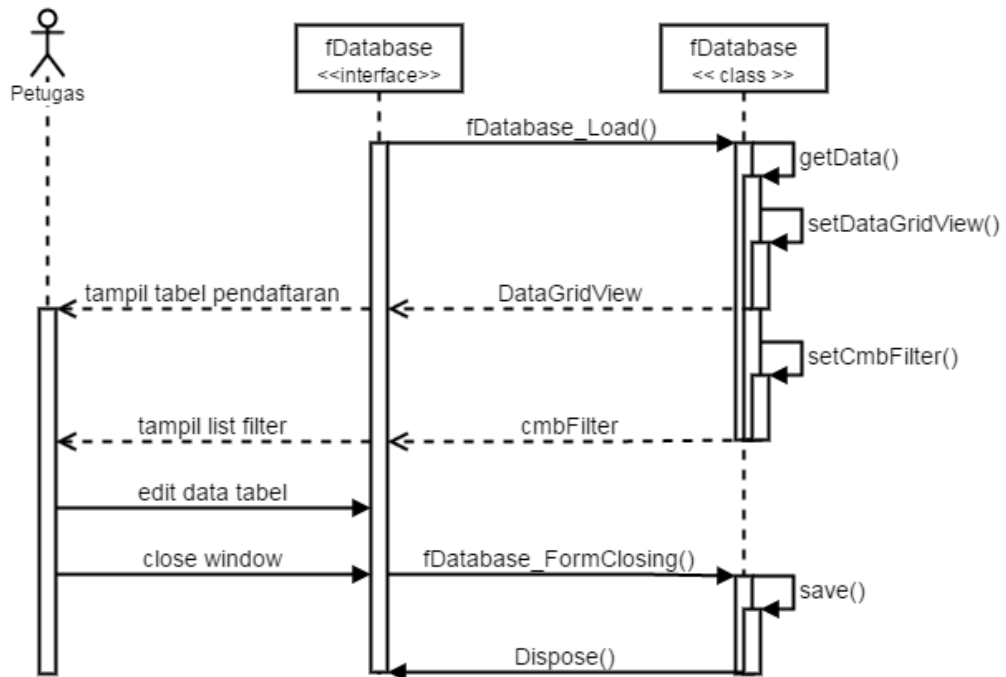
Gambar 4.12 menggambarkan urutan interaksi antara aktor dengan sistem yang terlibat pada *Sequence Diagram* Membuat Laporan. Interaksi terjadi pada *form* Laporan Pendaftaran Pasien setelah mendapatkan data dan tugas dari *form* Database Pendaftaran Pasien.

Laporan yang dibuat adalah laporan harian pendaftaran pasien sehingga sistem memerlukan tanggal daftar sebagai parameter utama untuk mencetak laporan. Tanggal daftar tersebut merujuk pada tanggal yang dipilih pada *form* Database Pendaftaran Pasien.

Setelah diketahui tanggal laporan, sistem akan mengambil data dari *database server* melalui koneksi ODBC untuk dapat diolah oleh DataSet Laporan (dsLaporan.xsd). Hasil pengolahan data dimasukkan pada variabel rDsLaporan dan dikembalikan pada *Form* Laporan sebagai Report Data

Source (rDataSource) kemudian disuguhkan untuk aktor melalui *Report Viewer*.

f. *Sequence Diagram Management Data Pendaftaran*



Gambar 4.13. *Sequence Diagram: Management Data Pendaftaran*

Keterangan:

Gambar 4.13 merupakan urutan interaksi antara aktor dengan sistem yang terlibat pada *Sequence Diagram Management Data Pendaftaran*. Interaksi terjadi pada pada *form Database Pendaftaran Pasien*.

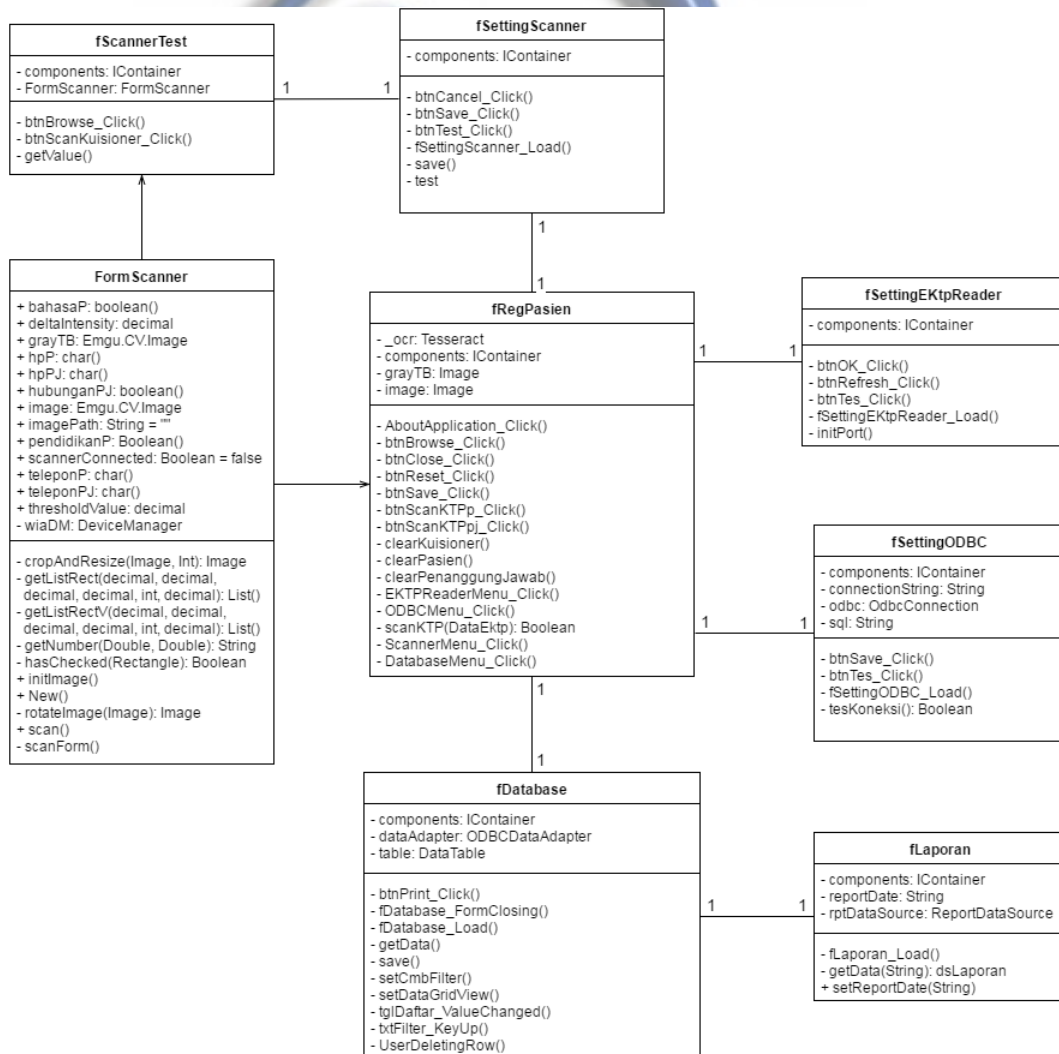
Secara umum pengolahan data pendaftaran pasien dilakukan secara langsung pada tabel pasien baru. Sedangkan tabel pasien baru menampilkan data pendaftaran pasien sesuai dengan tanggal daftar yang dipilih oleh aktor. Data diambil dari *server* melalui koneksi ODBC dan dimasukkan pada *DataTable*, lalu ditampilkan pada *form* oleh *DataGridView*. Pada tabel *DataGridView* aktor dapat mengolah data (*edit* dan *delete*) namun data perubahan akan disimpan ketika *form Database Pendaftaran Pasien* hendak ditutup.

4.3. Perancangan Data

Perancangan data mendeskripsikan struktur data dan aliran data pada setiap bagian sistem yang akan dibangun. Dalam dokumentasi ini, perancangan data yang diusulkan akan dibagi menjadi perancangan *class diagram*, *deployment diagram*, dan *entity relationship diagram*.

4.3.1. Class Diagram

Class diagram bertujuan untuk menggambarkan struktur statis *class* di dalam sistem. *Class* berfungsi untuk merepresentasikan sesuatu yang ditangani oleh sistem. Berikut *class diagram* Sistem Pendaftaran Pasien Rumah Sakit Umum yang diusulkan.

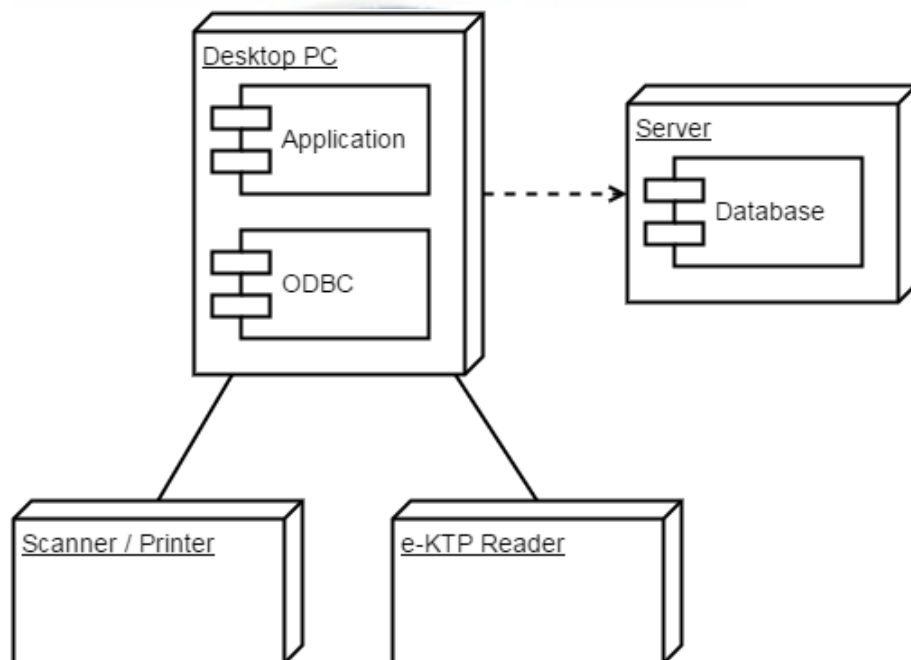


Gambar 4.14. Class Diagram

4.3.2. *Deployment Diagram*

Deployment diagram menampilkan *processors*, *devices*, dan *connections*. Setiap model berisi *deployment diagram* tunggal yang menunjukkan hubungan antara *processor* dan *device*. Diagram ini menggambarkan detail bagaimana komponen disebar ke dalam infrastruktur sistem, dimana komponen akan diletakkan, bagaimana kemampuan jaringan pada lokasi tersebut, spesifikasi *server*, dan hal-hal lain yang bersifat fisikal.

Berikut adalah gambaran *Deployment Diagram* yang diusulkan untuk Sistem Pendaftaran Pasien di Rumah Sakit Umum.



Gambar 4.15. *Deployment Diagram*

Sistem Pendaftaran Pasien di Rumah Sakit Umum ini berjalan dalam unit desktop PC yang bergantung (*dependency*) pada *database* dalam *server*. Sistem dihubungkan pada *server* melalui koneksi ODBC (*Open Database Connection*), dan untuk membaca e-KTP serta lebar LJK *desktop PC* harus dihubungkan pada e-KTP Reader serta *Document Scanner* melalui port USB. Sedangkan *printer* berfungsi untuk mencetak *output* berupa laporan pendaftaran pasien harian.

4.3.3. Entity Relationship Diagram

Entity Relationship Diagram menggambarkan detail perancangan basis data untuk aplikasi atau sistem yang akan dibuat, terdiri dari nama tabel, nama *field*, serta hubungan/relasi antar tabel.

Sistem Pendaftaran Pasien di Rumah Sakit Umum yang diusulkan hanya memiliki satu tabel sehingga tidak memiliki relasi. Hal tersebut dikarenakan untuk mempermudah aplikasi lain dalam mengolah atau mengambil informasi hasil dari sistem ini.

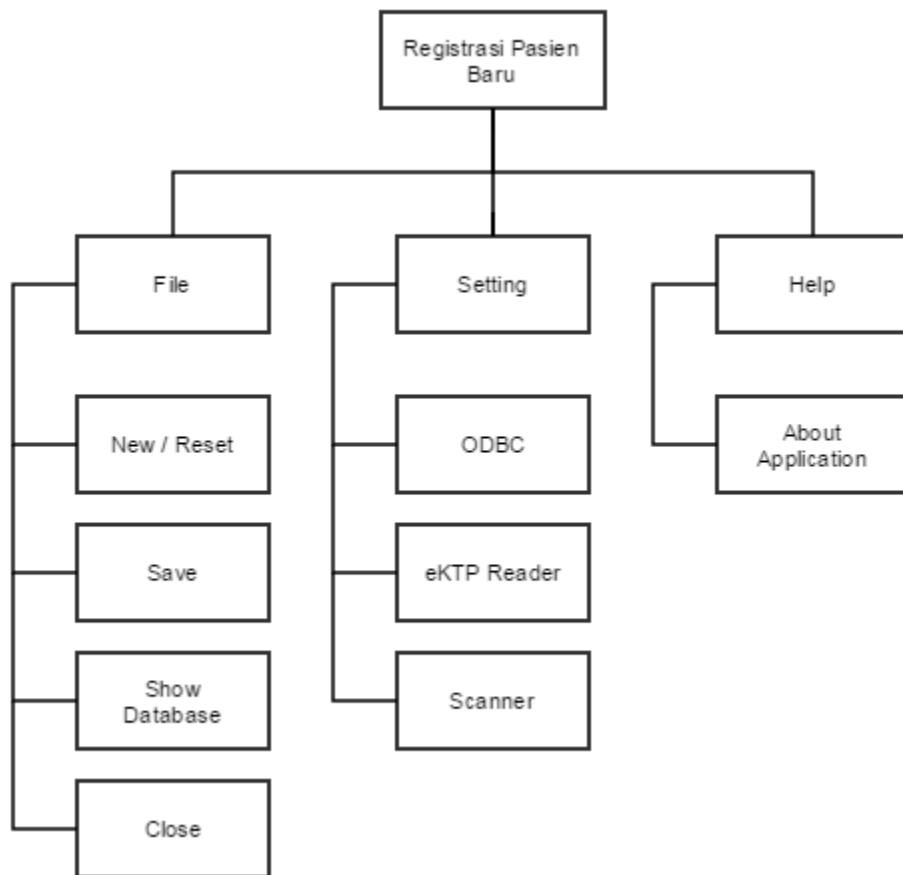
4.4. Perancangan Antarmuka

Perancangan antarmuka merupakan rancangan/desain untuk tampilan depan pada komputer, peralatan, mesin, perangkat komunikasi mobile, aplikasi perangkat lunak, atau situs web, yang berfokus pada pengalaman pengguna serta interaksi.

Tujuan dari perancangan antarmuka adalah untuk membuat interaksi pengguna sesederhana dan seefisien mungkin, dalam hal mencapai tujuan pengguna atau apa yang sering disebut dengan *user-centered design*. Desain antarmuka pengguna yang baik dapat memberikan penyelesaian pekerjaan dengan menggunakan tangan tanpa menarik perhatian yang tidak perlu terhadap dirinya sendiri.

Perancangan antarmuka kali ini meliputi perancangan struktur menu, perancangan *input*, dan perancangan *output*.

4.4.1. Struktur Menu



Gambar 4.16. Struktur Menu

4.4.2. Perancangan *Input*

Perancangan *input* adalah perancangan tampilan pada saat melakukan proses *input* data yang berkaitan dengan Sistem Pendaftaran Pasien di Rumah Sakit Umum. Rancangan *input* ini berfungsi untuk merencanakan tataletak setiap komponen-komponen yang akan digunakan ke dalam sistem, dibuat dengan sesederhana mungkin agar sistem dapat dengan mudah dipahami oleh pengguna. Adapun perancangan *input* yang diusulkan adalah sebagai berikut.

The image displays three screenshots of a web application form titled "Registrasi Pasien Baru".

- Top Left Screenshot:** Shows the "Pasien" tab. It includes a "No. Rekam Medis" field, a "Scan KTP" button, and various input fields for patient information: "No. KTP / NIK", "Nama", "Tempat / Tgl. Lahir" (with a date selector set to "09 September 2015"), "Jenis Kelamin", "Gol. Darah", "Alamat" (with sub-fields for "Greeting", "RT", "Kel. / Desa", "RW", "Kecamatan"), "Agama", "Status Perkawinan", and "Pekerjaan". Buttons for "Reset", "Save", and "Close" are at the bottom.
- Top Right Screenshot:** Shows the "Penanggung Jawab" tab. It has the same layout as the "Pasien" tab, with a "Scan KTP" button.
- Bottom Screenshot:** Shows the "Kuisisioner" tab. It features a "Scan" button and a "Browse" button. The "Pasien" sub-section includes "Pendidikan" (dropdown), "Bahasa" (radio buttons for Indonesian, Chinese, Japanese, English), "Telepon", and "Hp". The "Penanggung Jawab" sub-section includes "Hubungan" (dropdown), "Telepon", and "Hp". A large empty box with an 'X' is on the right. Buttons for "Reset", "Save", and "Close" are at the bottom.

Gambar 4.17. Perancangan *Input*: Registrasi Pasien Baru

Secara garis besar, proses *input* pada *form* Registrasi Pasien Baru terdiri dari tiga langkah utama yaitu *scan* e-KTP pasien, *scan* e-KTP penanggung jawab, dan *scan* kuisisioner, sehingga langkah-langkah tersebut disuguhkan ke dalam tiga tab. Satu-satunya input manual pada *form* ini adalah nomor rekam medis, sehingga diletakan sebuah *textbox* rekam medis pada di bagian atas *form*. Pada *tab* Pasien akan langsung diarahkan pada tombol Scan KTP, begitu pula pada *tab* Penanggung Jawab. Pada *tab* Kuisisioner *user* langsung diarahkan pada tombol Scan Kuisisioner. Tombol Reset diletakan pada pojok kiri bawah menjauh dari tombol Save dan Close di pojok kanan bawah agar terhindar dari kesalahan salah klik.

Gambar 4.18. Perancangan *Input: Setting Database*

Pada *form Setting Database* (Gambar 4.18) disediakan empat *textbox* untuk konfigurasi ODBC. Tombol *Tes Koneksi* pada pojok kiri bawah sedangkan tombol *OK* dan tombol *Cancel* pada pojok kanan bawah.

Gambar 4.19. Perancangan *Input: Setting e-KTP Reader*

Form Setting e-KTP Reader digunakan untuk memilih *port* yang tersedia, sehingga ditempatkan *combobox* untuk menampilkan *list port* serta tombol *Refresh* disebelahnya. Tombol *Tes Koneksi* tetap di pojok kiri bawah sedangkan tombol *OK* dan tombol *Cancel* di pojok kanan bawah.

Gambar 4.20. Perancangan *Input: Setting Scanner*

Form Setting Scanner mengatur nilai *threshold* dan *delta intensity* bertipe *numeric* sehingga disediakan dua *numeric-up-down*. Tombol *Test*, *OK*, dan *Cancel* ditempatkan sejajar dibawah *form*.

Gambar 4.21. Perancangan *Input: Database* Pendaftaran Pasien

Form Database Pendaftaran Pasien menampilkan tabel data pendaftaran pasien per hari sehingga disediakan satu *data-grid-view* dan satu *date-time-picker*. Selain itu disediakan *text-box* filter untuk melakukan pencarian data menurut *key-word* tertentu serta tombol *Print* untuk mencetak data yang sedang ditampilkan pada tabel.

4.4.3. Perancangan *Output*

Perancangan *output* merupakan perancangan yang berkaitan dengan hasil dari data yang sudah diolah. Berikut ini adalah perancangan *output* berupa laporan *hardcopy* yang dapat dihasilkan oleh Sistem Pendaftaran Pasien di Rumah Sakit Umum.

LAPORAN PENDAFTARAN PASIEN
9/13/2016 12:49:07 PM

No.	No. Rekam Medis	Nama Pasien	JK	Alamat	Penanggung Jawab	Tip/HP
-----	-----------------	-------------	----	--------	------------------	--------

Halaman 1 dari 1

Gambar 4.22. Perancangan *Output*: Laporan Pendaftaran Pasien.