

BAB III

ANALISA DAN PERANCANGAN DESAIN SISTEM

Analisis terhadap suatu sistem yang sedang berjalan merupakan suatu langkah penting dalam pemahaman permasalahan yang ada sebelum dilakukannya pengambilan keputusan atau tindakan dalam menyelesaikan permasalahan tersebut. Setelah dilakukan analisis terhadap sistem yang berjalan, langkah berikutnya adalah melakukan perancangan sistem baru. Dimana dalam perancangan sistem ini dapat memberikan gambaran tentang sistem yang akan dibuat.

Dengan kemajuan teknologi seperti saat ini, perusahaan ataupun organisasi skala menengah ke atas masih menggunakan cara-cara sederhana dan manual dalam menerapkan proses bisnisnya. Tidak hanya perusahaan, rumah sakit dan penyedia layanan kesehatan juga masih menggunakan sistem manual dalam melakukan kegiatan-kegiatan operasionalnya dimana salah satunya adalah proses rekam kesehatan. Masih adanya kelemahan dalam penggunaan sistem manual dalam proses rekam kesehatan di rumah sakit diantaranya : penggunaan media kertas masih rentan digunakan, terjadinya diagnosa yang sama berulang kali, proses penanganan pasien masih lambat, masih sulitnya mengetahui data pasien masing-masing rumah sakit, untuk mengetahui banyaknya kunjungan pasien ke rumah sakit masih sulit dan masih sulitnya untuk mengetahui riwayat kesehatan seorang pasien.

Metode yang digunakan dalam penelitian tugas akhir ini adalah :

3.1 Studi Literatur

Studi literatur merupakan suatu kegiatan untuk mengumpulkan dan mempelajari bahan-bahan literatur yang berhubungan dengan permasalahan yang dibahas dalam tugas akhir ini. Untuk keperluan tugas akhir ini, informasi didapatkan dari buku, jurnal, artikel di internet. Pada studi literatur ini ada beberapa tahapan yang dilakukan untuk memperoleh informasi-informasi yang berkaitan dengan penelitian tugas akhir ini, yaitu :

1. Mempelajari desain sistem berorientasi objek dan UML secara umum yang bertujuan untuk mendapatkan dasar teori yang akan digunakan sebagai landasan berpikir pada penelitian tugas akhir ini secara keseluruhan.
2. Mempelajari sistem rekam medis yang bertujuan untuk mendapatkan konsep proses rekam medis dan proses yang berhubungan dengan rekam medis secara tepat. Sehingga menghasilkan desain sistem informasi yang dapat diimplementasikan.
3. Mempelajari desain jaringan terpusat yang bertujuan untuk mendapatkan konsep proses penyimpanan data secara terpusat dan data *center*. Sehingga dapat dihasilkan desain jaringan yang tepat dan nantinya dapat diimplementasikan.

3.2 Analisa Kebutuhan

Merupakan suatu kegiatan untuk mengetahui, merancang dan menetukan kebutuhan-kebutuhan yang harus ada dalam sistem tersebut termasuk didalamnya rekomendasi yang harus dilakukan untuk memperbaiki sistem yang ada. Analisa kebutuhan sistem pada tugas akhir ini merupakan tahapan yang sangat vital

mengingat cakupan yang harus ada dalam sistem ini kompleks serta ada banyak hal yang harus diperhatikan dalam menentukan kebutuhan sistem itu sendiri. Data dan informasi yang diperlukan tersebut diperoleh dari berbagai sumber terkait untuk memberikan masukan yang lengkap bagi pengembangan sistem informasi ini. Data dan informasi tersebut antara lain :

1. Observasi dan Survei Study Kasus

Pada tahap ini dilakukan survey di 3 Rumah Sakit di Regional Kodya Denpasar yaitu RSUP. Sanglah, Rs. Wangaya dan RS. Bhakti Rahayu serta Dinas Kesehatan untuk mengetahui kondisi yang sebenarnya di dalam maupun luar organisasi.

2. Mengumpulkan Data Mengenai Sistem Rekam Medis dan Tipe Penyedia Layanan Kesehatan

Pada tahap ini dilakukan pengumpulan data-data tertulis yang berkaitan dengan proses yang ada dan dokumen-dokumen yang berkaitan dengan proses rekam kesehatan dan tipe rumah sakit. Selanjutnya data-data yang diperoleh akan digunakan ke dalam rancangan sistem untuk diolah.

3. Study literatur, Penelitian, dan Teori yang Berkaitan dengan Obyek Penelitian

Tahap pengumpulan pustaka dipergunakan untuk memperkuat dasar teori guna mendukung penggerjaan dan penelitian tugas akhir, untuk itu di dalam pencarian pustaka dipergunakan :

a. Jurnal

Paper-paper yang diterbitkan oleh suatu organisasi nasional maupun internasional dalam bentuk majalah fisik maupun elektronik. Paper-paper yang diambil adalah yang berhubungan dengan bidang ilmu kesehatan khususnya

rekam medis, rekam medis baik manual maupun elektronik, perancangan sistem informasi dan penggunaan basis data terpusat.

b. Prosiding

Paper-paper yang dibutuhkan adalah paper-paper yang diterbitkan di suatu seminar dalam bentuk buku yang berhubungan dengan bidang ilmu kesehatan khususnya rekam medis, rekam medis baik manual maupun elektronik, perancangan sistem informasi dan penggunaan basis data terpusat.

c. Text Book

Text book yang dibutuhkan adalah materi-materi yang berhubungan dengan bidang sistem rekam medis manual maupun elektronik dan perancangan sistem informasi.

4. Menentukan Kebutuhan dari Sistem Informasi Rekam kesehatan Elektronik Terpusat

Tahap ini dilakukan sebelum melakukan pemodelan yaitu menentukan kebutuhan rancangan sistem sesuai dengan masing-masing tipe rumah sakit agar semua rumah sakit yang dapat bergabung dala sistem ini sesuai dengan tipe rumah sakit tersebut

Data-data yang dibutuhkan oleh sistem antara lain sebagai berikut :

- a) Data tipe dan fasilitas yng disediakan masing-masing rumah sakit
- b) Data syarat-syarat pendaftaran
- c) Tahapan untuk mendapatkan data rekam kesehatan di Rumah Sakit
- d) Alur pasien

Setelah itu berdasarkan data yang ada akan diproses sehingga menghasilkan output yang berguna untuk bagian rekam medis, dokter, pasien dan

rumah sakit dalam menyimpan dan mengetahui riwayat kesehatan pasien yang ditangani pada rumah sakit.

Tabel 3.1 berikut menunjukkan kebutuhan-kebutuhan terhadap sistem didapat dari hasil analisa kebutuhan.

Tabel 3.1 Tabel Kebutuhan Sistem

Proses yang Sedang Berjalan	Sistem yang Dibutuhkan
Mencatat pendaftaran pasien yang masih menggunakan lembaran kertas secara manual sehingga sering terjadi kesalahan.	Dibutuhkan suatu sistem basis data yang dapat menyimpan dan memproses data-data kebutuhan pendaftaran pasien.
Pencarian data pasien masih menggunakan sistem manual karena data pasien disimpan dalam satu berkas di sebuah rak, sehingga membutuhkan waktu yang lama untuk memperoleh informasi tentang pasien.	Dibutuhkan sistem basis data untuk menyimpan dan memproses data pasien sehingga membantu dalam proses pencarian data pasien secara cepat melalui menu pencarian pasien.
Proses pencatatan hasil rekam medis yang dilakukan oleh dokter masih menggunakan lembaran kertas rekam medis secara manual sehingga sering terjadi kesalahan.	Dibutuhkan suatu sistem basis data untuk menyimpan data rekam kesehatan secara elektronik.
Proses rujukan pasien masih dilakukan dengan membawa hasil rekam medis dari rumah sakit asal untuk dilanjutkan ke rumah sakit tujuan rujukan	Dibutuhkan suatu sistem basis data yang dapat menyimpan data rekam medis secara terpusat (sentralisasi) agar dapat diakses dari rumah sakit yang terhubung dengan sistem

5. Analisa dan Memodelkan Proses Bisnis dengan Unified Modeling Language

Langkah ini dilakukan untuk memodelkan proses bisnis dari sistem informasi rekam kesehatan elektronik terpusat yang akan dirancang. Rancangan

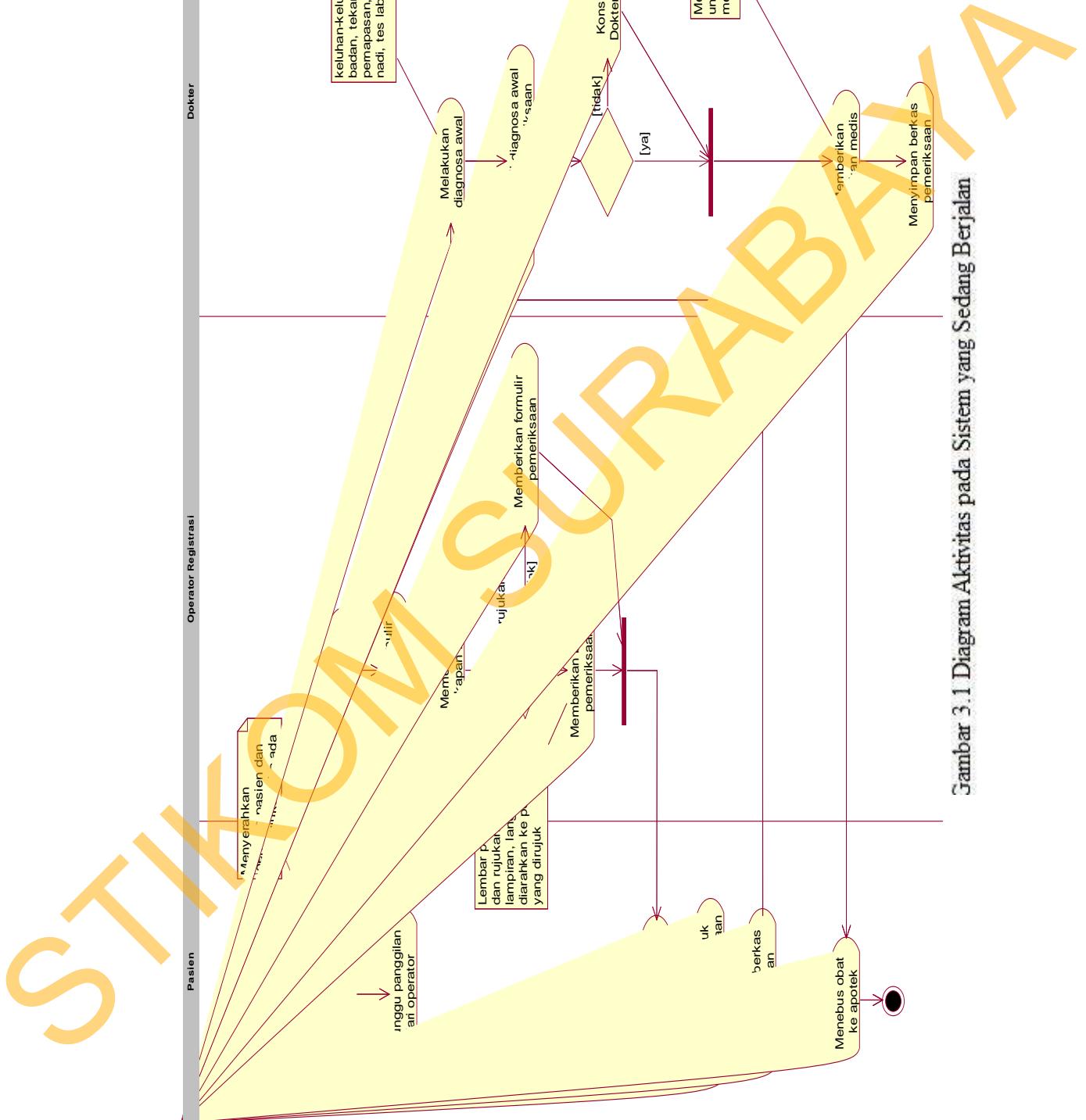
ini dibuat dalam bentuk diagram aliran data, yang digambarkan dengan menggunakan Rational Rose.

Pada tahap ini mulai dibuat rancangan sistem berdasarkan data-data yang sudah didapatkan. Urutan langkah-langkah perancangan sistem informasi berorientasi objek dengan UML adalah sebagai berikut :

- a. Pemodelan bisnis berupa *business use case diagram*
- b. Desain diagram aktivitas
- c. Pemodelan use case sistem berupa *use case diagram*
- d. Analisis aliran kejadian (*flow of event*), *sequential diagram* dan *collaboration diagram* level analisis
- e. Desain *sequential diagram*, *collaboration diagram*, *class diagram*, *statechart diagram* dan *deployment diagram*.

3.3 Analisa Sistem

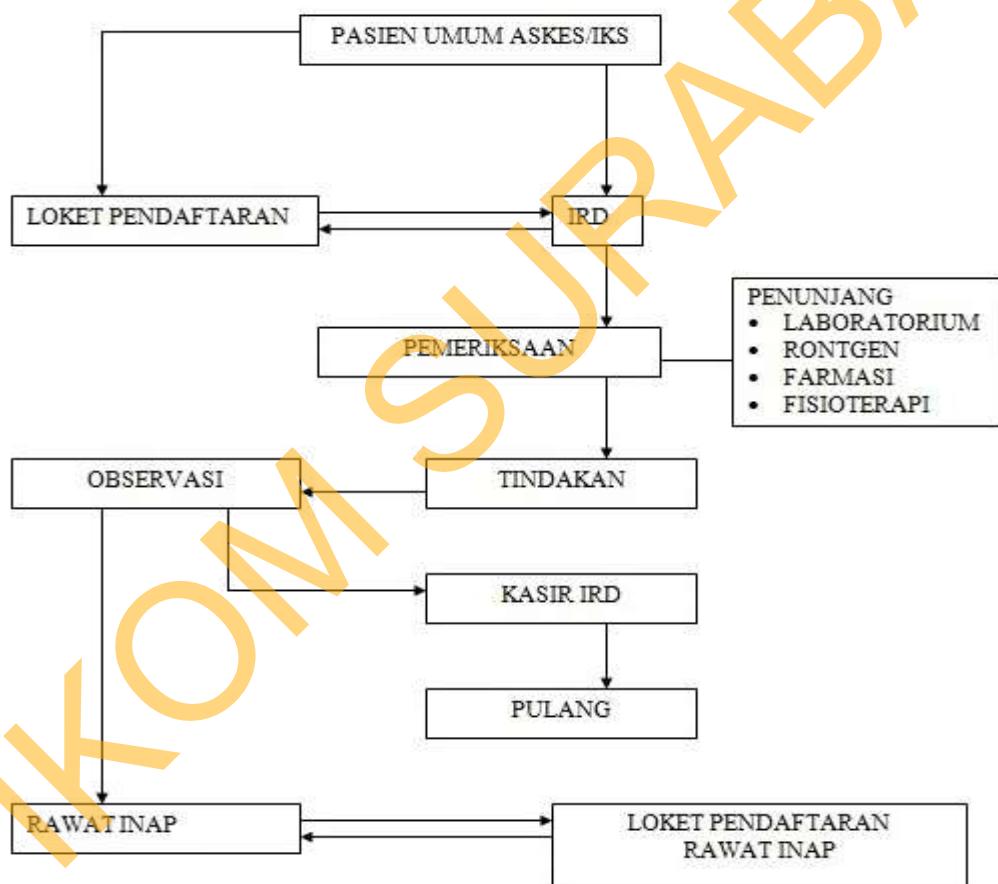
Dari hasil observasi saat datang ke Rumah Sakit yaitu RSUP. Sanglah, Rs. Wangaya dan RS. Bhakti Rahayu. Untuk proses yang berjalan masing-masing rumah sakit sama dan masih menggunakan sistem manual digambarkan dengan diagram aktivitas sebagai berikut pada gambar 3.1.



Jambar 3.1 Diagram Aktivitas pada Sistem yang Sedang Berjalan

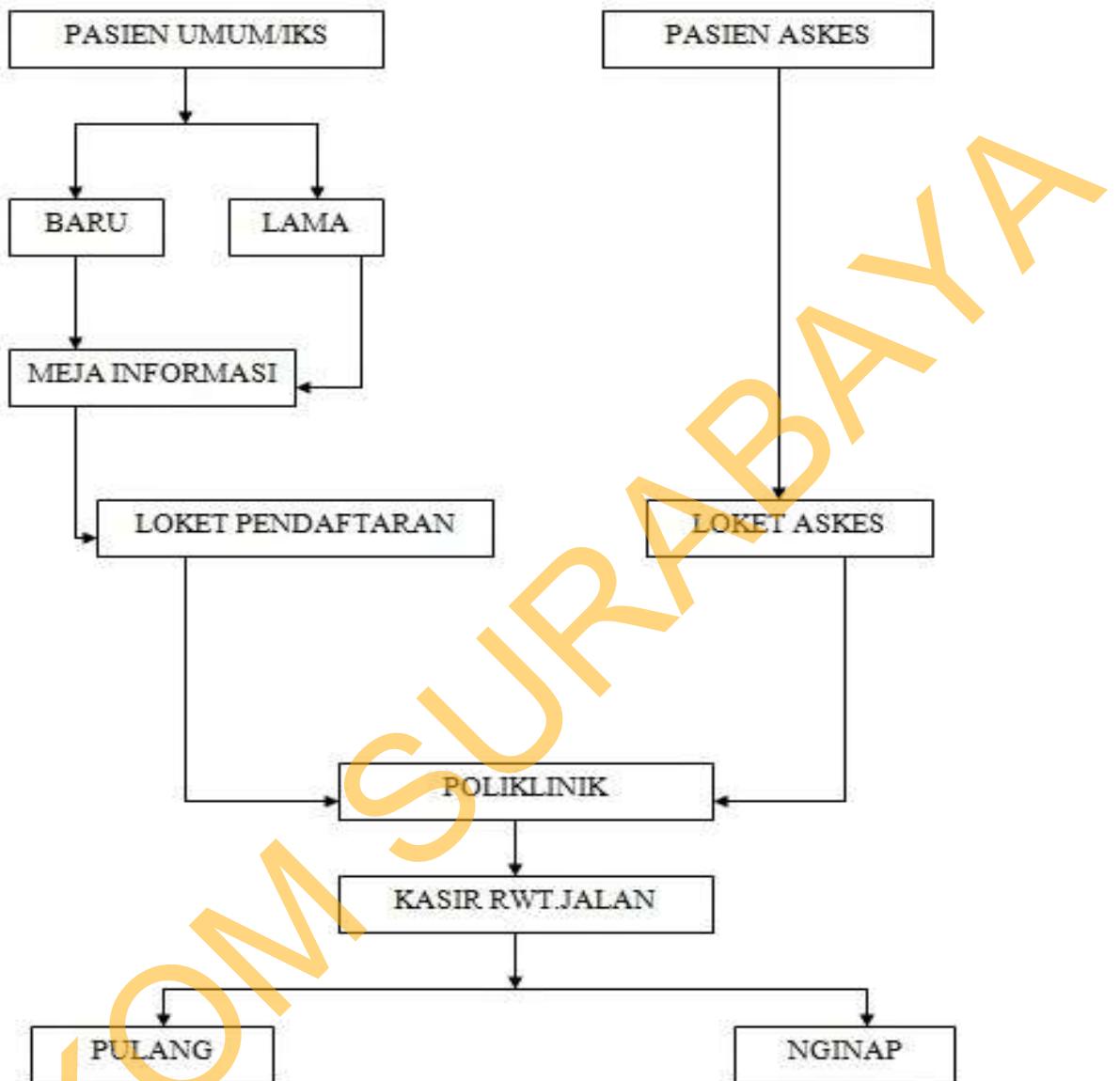
Data rekam medis disini yaitu berkas hasil pemeriksaan dokter yang nantinya disimpan di rak-rak. Penggunaan kertas masih sangat banyak jadi cendrung terjadi kehilangan atau kerusakan data rekam medis. Dengan demikian sangat sulit untuk mengetahui riwayat kesehatan pasien pada rumah sakit tersebut. Bahkan untuk mengetahui penyakit apa saja yang sedang banyak diderita oleh pasien di rumah sakit tersebut.

3.3.1 Alur Pasien IRD



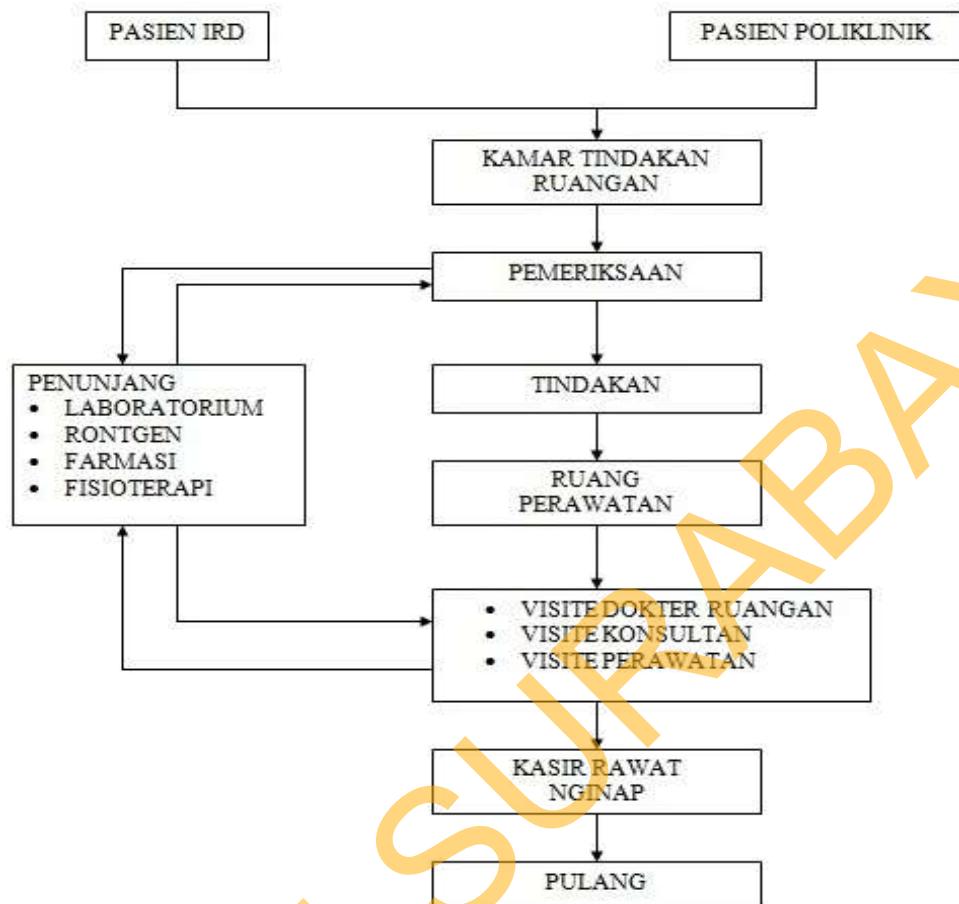
Pada diagram diatas digambarkan mengenai alur pasien untuk tindakan gawat darurat di rumah sakit. Penanganan terhadap pasien diutamakan daripada pendaftaran. Pendaftaran pasien dapat dilakukan bersamaan dengan penanganan medis.

3.3.2 Alur Pasien Rawat Jalan



Pada gambar diatas merupakan alur pasien untuk rawat jalan. Setelah melakukan pendaftaran dan mendapatkan penanganan pasien, jika diperlukan untuk dilakukan kunjungan lagi maka pasien tersebut akan diberikan jadwal untuk kunjungan berikutnya dengan kata lain disebut rawat jalan.

3.3.3 Alur Pasien Rawat Inap



Pada rawat inap, alur yang terjadi di rumah sakit seperti yang digambarkan pada gambar diatas. Pasien setelah mendapatkan pelayanan kesehatan yang dirasa diperlukan untuk rawat inap karena alasan penanganan dan keadaan pasien, maka pasien tersebut akan diarahkan ke rawat inap di ruang perawatan sampai pasien dinyatakan boleh pulang.

3.4 Desain Sistem

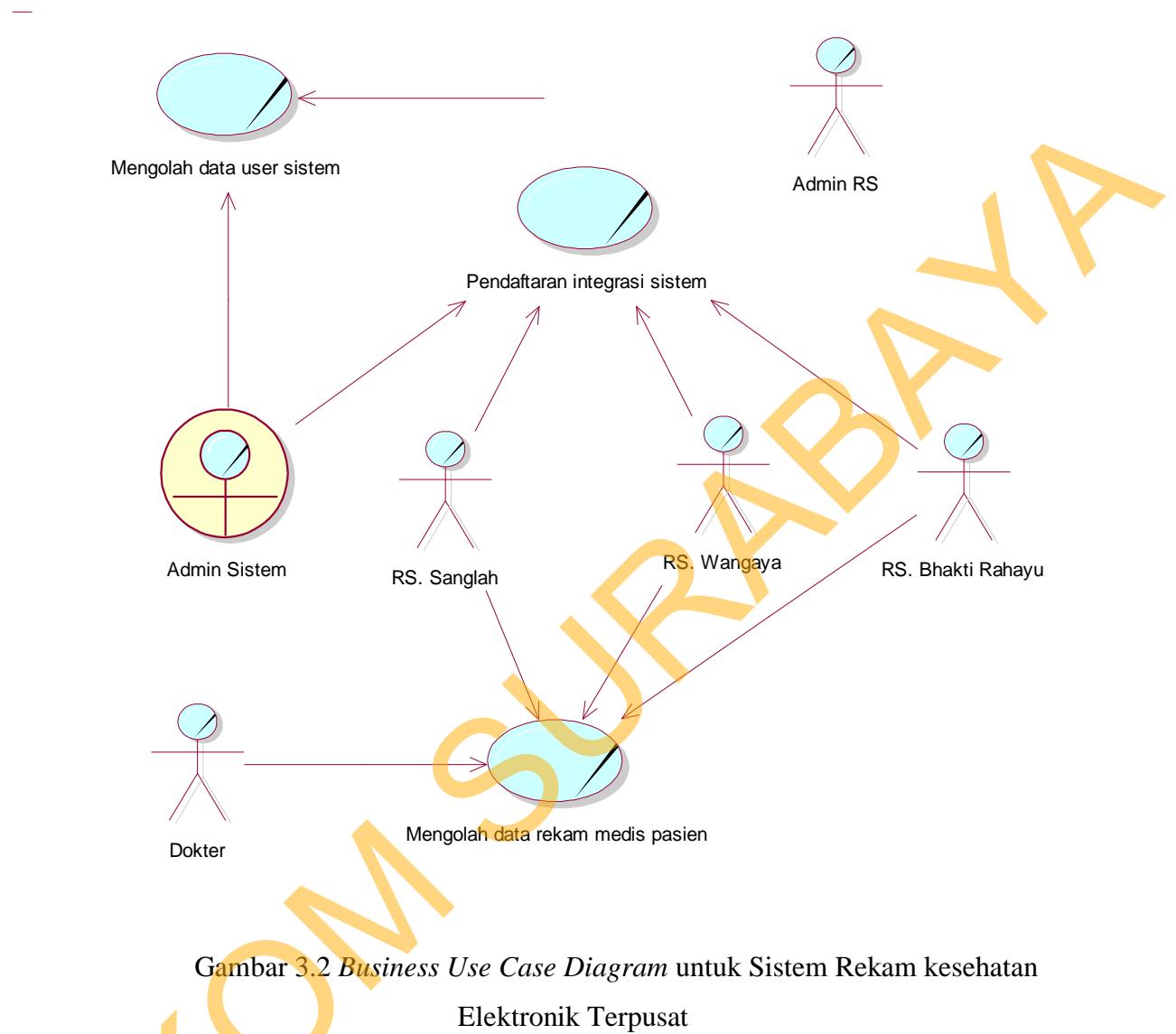
Pada tugas akhir desain sistem informasi ini, menggunakan diagram-diagram UML yang bertujuan agar model yang dibuat mendekatkan realitas dan siap dipakai. Di proyek pengembangan sistem apapun, fokus utama dalam analisis dan

perancangan adalah model. Dengan model kita bisa merepresentasikan sesuatu karena model mudah dan cepat untuk dibuat, bisa digunakan sebagai simulasi untuk mempelajari lebih detail tentang sesuatu. Model bisa dikembangkan sejalan dengan pemahaman kita tentang sesuatu, bisa memberikan penjelasan lebih rinci tentang suatu model dimana model bisa mewakili sesuatu yang nyata maupun yang tidak nyata. Diagram yang berbeda-beda dapat menyatakan tingkatan yang berbeda-beda dalam proses rekayasa perangkat lunak. Diagram-diagram UML yang digunakan adalah sebagai berikut :

3.4.1 *Business Use Case Diagram*

Diagram use case bisnis atau *business use case diagram* menunjukkan interaksi antara use case bisnis (*business use case*), aktor bisnis (*business actor*) dan pekerja bisnis (*business worker*) dalam sebuah organisasi. Diagram ini menggambarkan yang lengkap tentang apa saja yang dilakukan organisasi/perusahaan. Dengan diagram ini dimaksudkan agar dapat memberikan informasi yang jelas tentang proses bisnis secara cepat tanpa menyebabkan kebingungan bagi pembaca. *Business use case diagram* dalam Desain sistem informasi rekam kesehatan elektronik terpusat dapat digambarkan seperti gambar

3.2.



Gambar 3.2 *Business Use Case Diagram* untuk Sistem Rekam kesehatan Elektronik Terpusat

Pada gambar 3.2 ini terdapat empat *business actor* yaitu Admin RS, Dokter, RS. Sanglah, RS. Wangaya, dan RS. Bhakti Rahayu. Dikatakan sebagai bisnis aktor karena pasien merupakan orang atau organisasi luar yang berinteraksi dengan sistem. Terdapat satu *business worker* yaitu Admin Sistem.

Untuk lebih lanjut akan dijelaskan *business actor*, *business worker* dan *business use case* pada sistem informasi ini antara lain :

1. Rumah Sakit

Rumah sakit disini berperan sebagai *business actor*. Rumah sakit melakukan pendaftaran untuk bergabung ke dalam intergrasi sistem. Rumah sakit tersebut antara lain RS. Sanglah, RS. Wangaya, dan RS. Bhakti Rahayu.

2. Dokter

Business actor yang kedua yaitu Dokter. Dokter berperan penting dalam proses mendapatkan data rekam medis mulai dari proses pemeriksaan sampai dengan mengolah data rekam medis. Hanya dokter yang berhak mengisi dan menambahkan data rekam medis seorang pasien.

Dokter disini merupakan dokter yang ada di rumah sakit dengan syarat telah terdaftar pada sistem terintegrasi.

3. Admin RS

Admin RS merupakan *business actor* yang bertugas untuk membantu kegiatan admin sistem dalam mengolah data user dan membantu operator registrasi untuk melengkapi identitas pasien dalam proses pendaftaran. Ruang lingkup kerja seorang admin RS hanya menangani satu rumah sakit di tempat kerjanya saja.

4. Admin Sistem

Admin sistem merupakan *business worker* yang bertugas melakukan pendaftaran user yang bergabung ke dalam integrasi sistem. Selain itu Admin Sistem juga dapat mengolah data user sistem yaitu akan membentuk admin RS di masing-masing rumah sakit yang dilakukan untuk membantu kerjanya.

5. Mengolah Data User Sistem

Bisnis use case ini melibatkan Admin Sistem dan Admin RS. Menggambarkan Admin Sistem dapat melakukan penambahan Admin RS baru, dan juga dapat mengubah status Admin RS menjadi non aktif apabila Admin RS keluar atau berhenti dari rumah sakit atau sistem.

6. Pendaftaran Integrasi Sistem

Disini akan digambarkan Rumah sakit melakukan pendaftaran ke dalam sistem rekam kesehatan elektronik terpusat. Proses pendaftaran akan dilakukan oleh Admin Sistem. Rumah sakit yang melakukan pendaftaran akan dicatat fasilitas yang disediakan dan tenaga medis yang dimiliki.

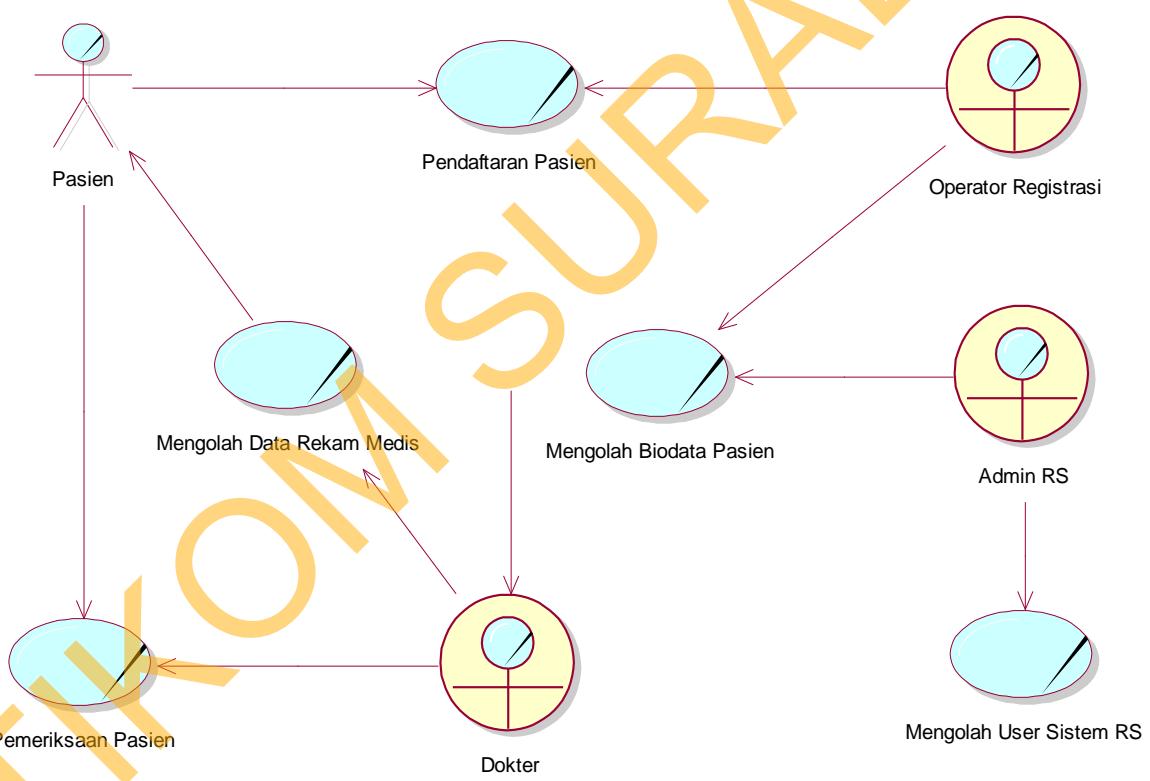
Apabila rumah sakit yang telah tergabung keluar dari sistem maka rumah sakit yang bersangkutan akan di non aktifkan. Hal ini dilakukan agar data rekam kesehatan tetap terjaga kerahasiaannya.

7. Mengolah Data Rekam Kesehatan

Bisnis use case ini merupakan kegiatan yang terjadi dalam rumah sakit. Dokter memegang peranan penting dalam mengisi dan menambahkan data rekam kesehatan setelah melakukan pemeriksaan, diagnosa dan memberikan tindakan medis.

Proses pemeriksaan antara lain menanyakan keluhan-keluhan pasien (anamnesa), memeriksa suhu badan, laju pernapasan, denyut nadi, tekanan darah dan tes laboratorium. Semua hasil pemeriksaan akan dicatat beserta tindakan medis dan obat yang diberikan.

Pada gambar 3.3 berikut yang akan digambarkan merupakan proses yang terjadi dalam rumah sakit. Bagaimana terjadinya proses pendaftaran sampai dengan proses pencatatan rekam kesehatan serta proses-proses yang terjadi di dalam rumah sakit akan digambarkan.



Gambar 3.3 Bisnis Use Case untuk Sistem Rumah Sakit

Pada gambar 3.3 digambarkan terdapat satu *business actor* yaitu Pasien. Sedangkan untuk *business worker* ada tiga antara lain Operator Registrasi, Dokter, dan Admin RS. Untuk lebih jelasnya akan diterangkan *business actor*, *business worker*, dan *business use case* pada sistem di rumah sakit.

1. Pasien

Pasien merupakan satu-satunya *business actor* yang terlibat dalam sistem rumah sakit. Pasien merupakan sumber utama dari data rekam medis. Sebelum mendapatkan pemeriksaan dan tindakan medis pasien harus melakukan pendaftaran terlebih dahulu.

2. Admin RS

Admin RS merupakan *business worker* pada sistem rumah sakit. Berfungsi untuk membantu admin sistem terintegrasi dalam mengolah data user dalam satu rumah sakit dan membantu operator registrasi dalam melengkapi identitas pasien pada saat pendaftaran pasien. Admin RS mendaftarkan operator registrasi dan dokter yang akan terlibat dalam sistem.

3. Operator Registrasi

Operator registrasi bertugas menerima pendaftaran dan mengolah biodata pasien. Operator registrasi juga dapat memeriksa status pasien apakah sudah terdaftar pada sistem atau masih berstatus pasien baru sehingga pasien harus melakukan pendaftaran pasien baru bila belum terdaftar.

4. Dokter

Dokter menangani segala keluhan penyakit pasien dengan syarat pasien telah melakukan pendaftaran dan memiliki nomor rekam kesehatan. Proses pemeriksaan, diagnosa, dan tindakan medis dilakukan sepenuhnya oleh dokter.

Hasil kegiatan dokter terhadap pasien akan dicatat sebagai data rekam kesehatan. Data rekam kesehatan tersebut sangat bersifat rahasia sehingga hanya dokter dan pasien berangkutan yang dapat melihat data rekam medisnya.

5. Pendaftaran Pasien

Operator registrasi menerima proses pendaftaran ketika pasien dating ke loket pendaftaran. Pertama kali pasien akan menyerahkan identitas dan surat rujukan bila ada. Operator registrasi akan melakukan pengecekan apakah pasien yang melakukan pendaftaran merupakan pasien lama atau baru.

6. Mengolah Biodata Pasien

Disini akan digambarkan proses menambahkan dan mengubah biodata pasien. Proses ini dilakukan oleh operator registrasi dan admin RS. Operator memeriksa apakah pasien yang daftar pasien lama atau baru, bila baru pasien tersebut akan diberikan nomor rekam medis baru dan mengisikan identitas pasien. Untuk melengkapi identitas lebih terinci akan dilakukan oleh admin RS. Sedangkan apabila pasien yang melakukan pendaftaran merupakan pasien lama, operator registrasi tidak menambahkan identitas lagi hanya menambahkan tanggal pendaftaran saja.

7. Mengolah User Sistem di RS

Bisnis use case ini menggambarkan admin RS yang diberikan kepercayaan untuk menambahkan user baru yaitu dokter dan operator registrasi dan mengubah status user sistem di satu rumah sakit yang ditangani. User-user yang telah terdaftar pada sistem barlah bias melakukan akses terhadap identitas pasien dan rekam kesehatan.

Apabila ada user yang berhenti atau keluar dari rumah sakit, maka status user akan dirubah menjadi tidak aktif.

8. Pemeriksaan Pasien

Setelah melakukan pendaftaran pasien akan mendapatkan pemeriksaan yaitu proses diagnosa dan diberikan tindakan medis. Semua proses pemeriksaan dilakukan oleh dokter di poliklinik tujuan yang terkait dengan keluhan pasien.

9. Mengolah Data Rekam Medis

Mengolah data rekam medis disini dimaksudkan untuk mengisi dan menambahkan data rekam medis seorang pasien. Dokter memiliki peranan untuk menambahkan data hasil pemeriksaan terhadap pasien ke data rekam kesehatan yang ditangani. Data rekam medis bersifat rahasia jadi yang boleh mengetahui hanya dokter dan pasien yang bersangkutan. Data rekam medis tidak dapat diubah ataupun dihapus sehingga datanya bersifat berkelanjutan.

3.4.2 *Activity Diagram*

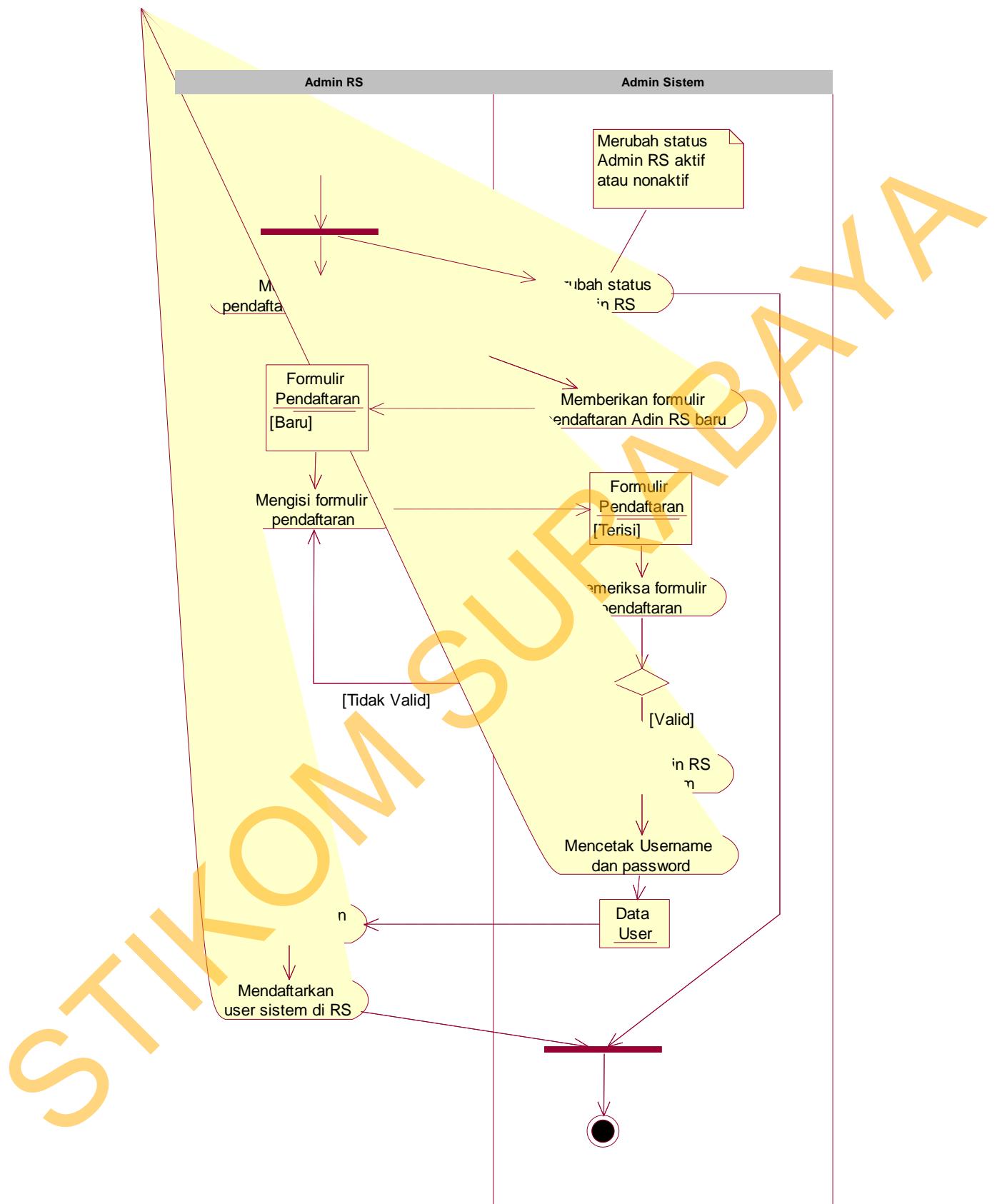
Activity diagram menggambarkan aliran fungsionalitas sistem yang terjadi pada tiga rumah sakit yaitu RSUP. Sanglah, RS. Wangaya, dan RS. Bhakti Rahayu. Diagram aktivitas adalah sebuah cara untuk memodelkan aliran kerja dari use case bisnis dalam bentuk grafik. Pada tahap pemodelan bisnis, aktivitas yang digunakan untuk menunjukkan aliran kerja bisnis (*business workflow*). Dalam use case diagram dapat juga digunakan untuk menggambarkan aliran kejadian (*flow of event*). Diagram ini menunjukkan langkah-langkah di dalam aliran kerja, titik-titik keputusan di dalam aliran kerja dan objek-objek yang digunakan dalam aliran kerja.

Activity diagram dalam perancangan dan desain sistem informasi rekam kesehatan elektronik terpusat untuk tiga rumah sakit di Kota Madya Denpasar digambarkan sebagai berikut :

A. Activity Diagram untuk Bisnis Use Case Mengolah Data User Sistem

Activity diagram mengolah data user sistem yang digambarkan pada gambar 3.4 dilakukan oleh admin sistem yang mana akan membentuk admin RS untuk membantu kegiatannya. Proses dimulai dari admin RS yang akan mendaftar mengisi formulir pendaftaran. Setelah diserahkan kepada admin sistem, akan diperiksa lebih dulu. Jika telah divalid, admin RS akan didaftarkan sebagai user admin RS pada sistem terintegrasi.

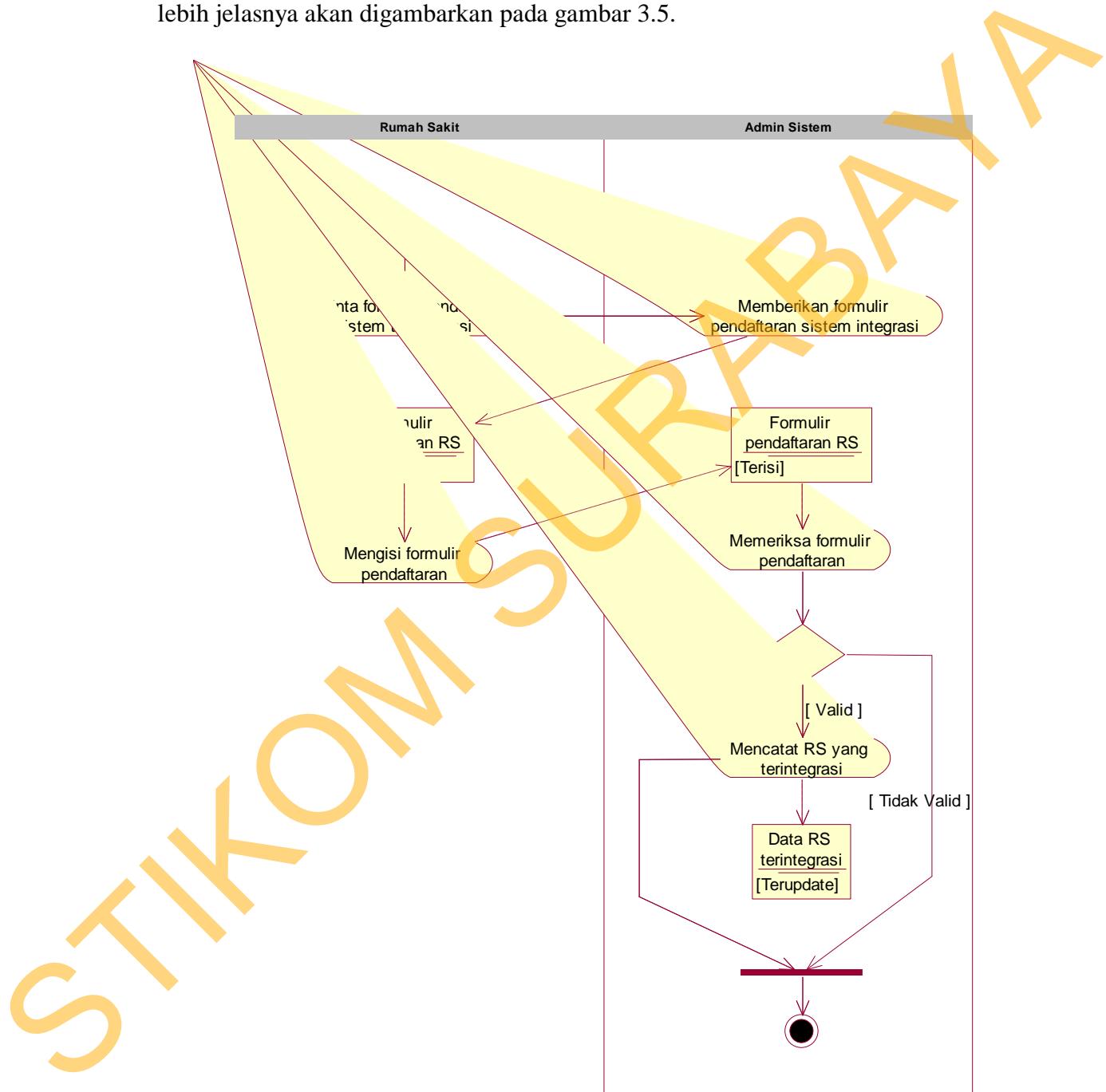
Admin RS memiliki tugas membantu admin sistem dalam mendaftarkan user di satu rumah sakit yang ditanganiya saja. Semua user di rumah sakit yang akan mendaftarkan diri pada sistem terpusat akan dilayani oleh admin RS. Apabila admin RS berhenti atau keluar dari sistem, Admin sistem akan mengubah status admin RS menjadi tidak aktif. Semua kegiatan yang dilakukan mengenai user sistem sepenuhnya dilakukan oleh admin sistem.



Gambar 3.4 *Activity Diagram* untuk Bisnis Use Case Mengolah Data User Sistem

B. Activity Diagram untuk Bisnis Use Case Pendaftaran Integrasi Sistem

Selain menangani pengolahan data user admin RS, admin sistem juga dapat menerima pendaftaran integrasi sistem untuk sebuah rumah sakit. Untuk lebih jelasnya akan digambarkan pada gambar 3.5.



Gambar 3.5 *Activity Diagram* untuk Bisnis Use Case

Pendaftaran Integrasi Sistem

Dari gambar 3.5 terdapat dua *swimline* yaitu Rumah Sakit dan Admin sistem. Admin sistem akan melayani segala sesuatu tentang pendaftaran rumah sakit ke sistem rekam kesehatan elektronik terpusat. Identitas rumah sakit, fasilitas yang disediakan, daya tampung dan tenaga medis yang dimiliki akan dicatat ke dalam sistem. Untuk pendaftaran disesuaikan dengan tipe rumah sakit ada.

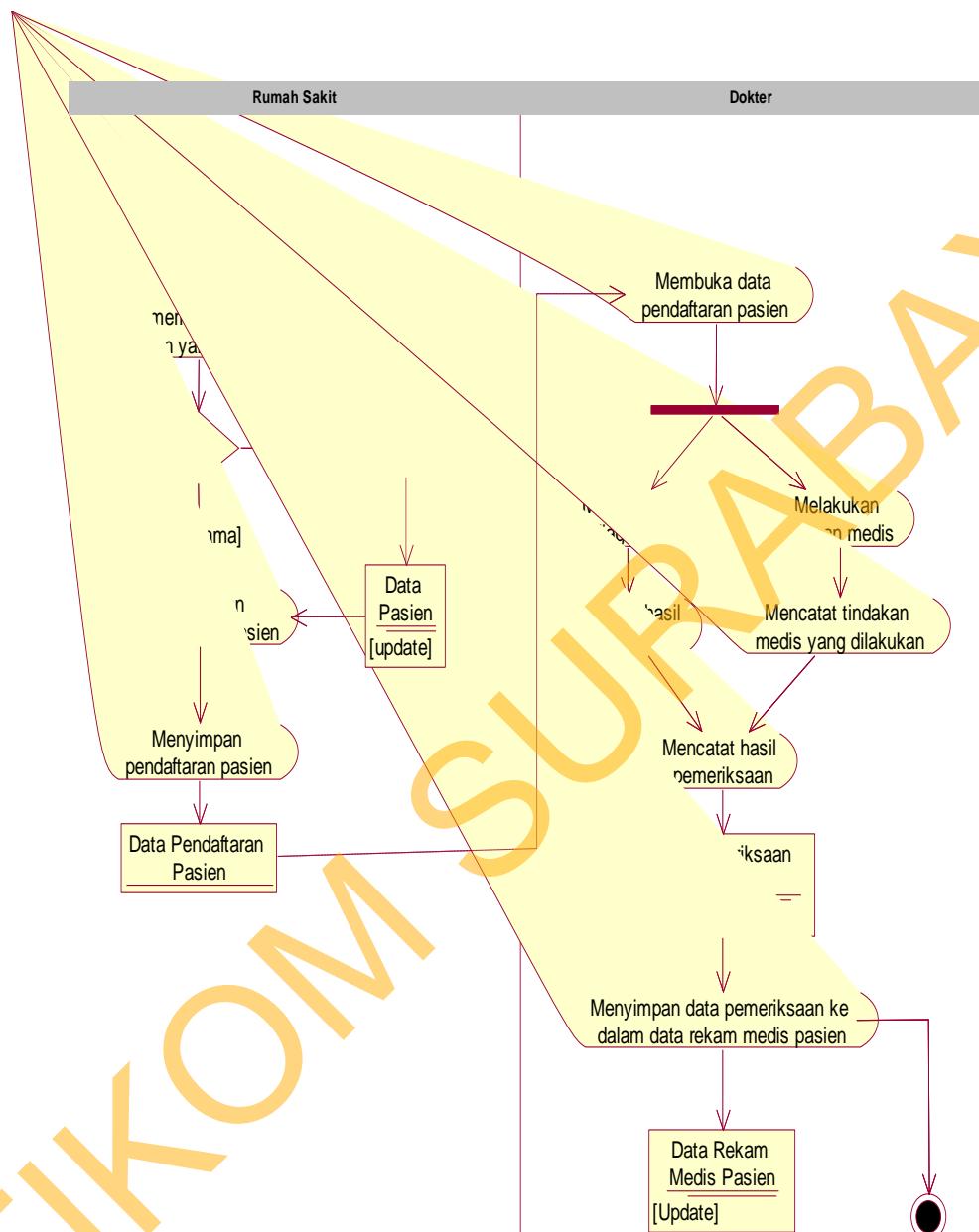
C. *Activity Diagram* untuk Bisnis Use Case Mengolah Data Rekam kesehatan

Pada gambar 3.6 terdapat dua *swimline* yaitu dokter dan rumah sakit. Dokter yang berada dalam suatu rumah sakit yang melakukan proses pemeriksaan, diagnosa, dan tindakan medis akan menambahkan semua kegiatan yang dilakukan terhadap pasien ke dalam data rekam kesehatan.

Pada rumah sakit digambarkan kegiatan awal pasien yang mendaftar ke dalam sistem. Kemudian setelah proses pendaftaran dilakukan maka akan dilakukan pemeriksaan. Pemeriksaan disini yaitu melakukan diagnosa dan memberikan tindakan medis terhadap hasil diagnosa yang dilakukan. Seluruh kegiatan dokter tersebut akan dicatat dan ditambahkan pada laporan pemeriksaan dan dicatat pada data rekam kesehatan.

Sebelum melakukan penyimpanan ke database terpusat, data rekam medis akan diperiksa terlebih dahulu untuk menghindari kesalahan. Karena data rekam medis tidak dapat diubah ataupun dihapus.

Diagram aktivitas dalam sistem yang ada di rumah sakit digambarkan sebagai berikut :



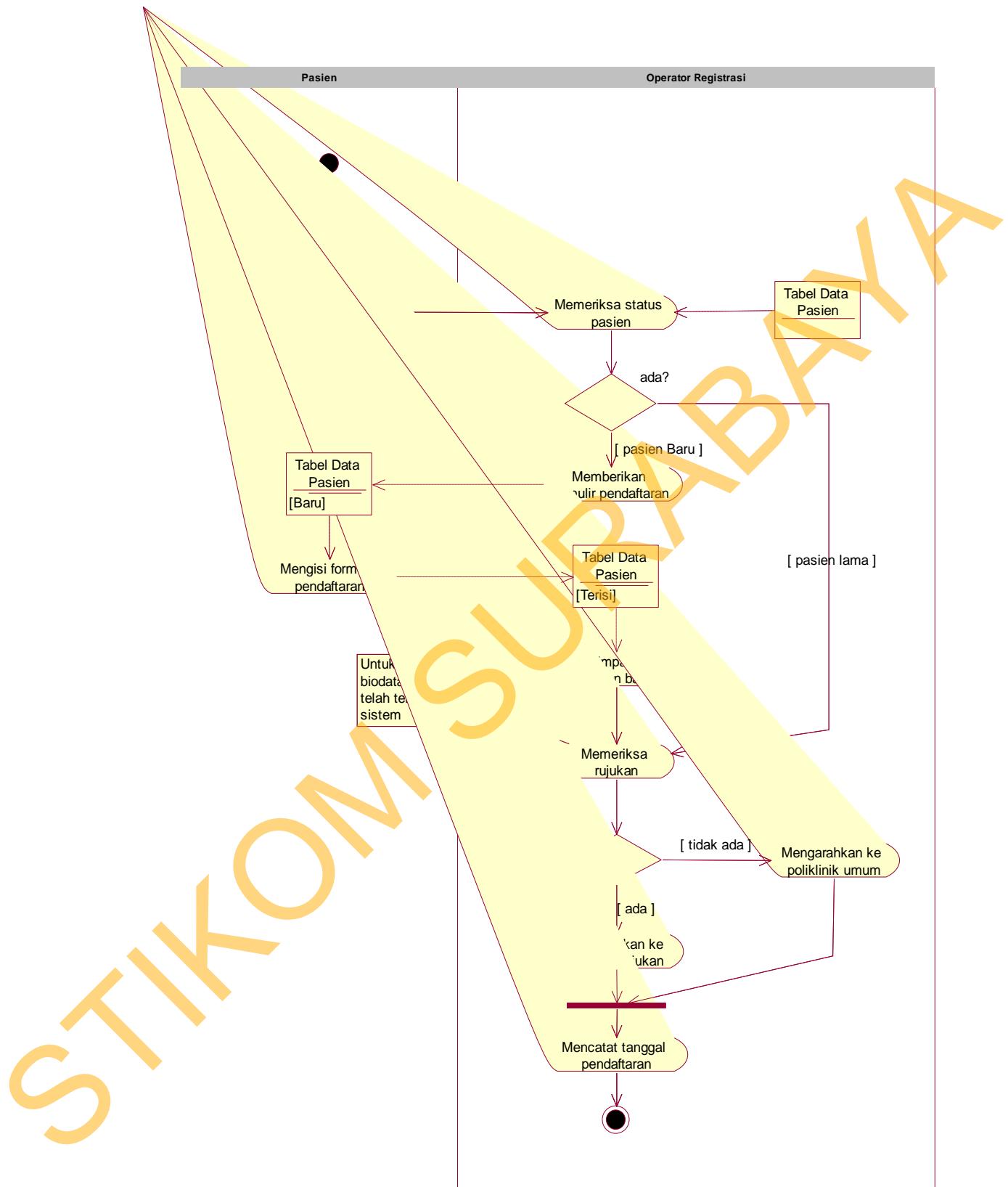
Gambar 3.6 *Activity Diagram* untuk Bisnis Use Case Mengolah Data Rekam kesehatan

D. Activity Diagram untuk Bisnis Use Case Pendaftaran Pasien

Pada gambar 3.7 digambarkan ada dua *swimlane* yaitu pasien dan operator registrasi. Proses dimulai dari pasien yang melakukan pendaftaran mendatangi loket pendaftaran. Pasien menyerahkan identitas kepada operator registrasi. Di operator registrasi akan diperiksa apakah pasien merupakan pasien alama atau pasien baru. Jika pasien merupakan pasien baru maka pasien akan diberikan formulir pendaftaran yang kemudian disikan identitasnya.

Setelah pasien menyerahkan formulir pendaftaran, operator registrasi akan mencatat biodata pasien ke dalam sistem. Setelah terdaftar pada sistem pasien akan ditanyakan surat rujukannya begitupun untuk pasien lama yang datang langsung ditanyakan rujukannya. Jika ada rujukan maka pasien akan diarahkan ke poliklinik rujukan dan apabila tidak ada rujukan pasien akan diarahkan ke poliklinik umum untuk mendapatkan pemeriksaan.

Pada saat pendaftaran pasien, tanggal pendaftaran akan dicatat oleh sistem sehingga apabila terjadi kekeliruan dapat ditangani. Untuk lebih jelasnya kita dapat melihat gambar 3.7.

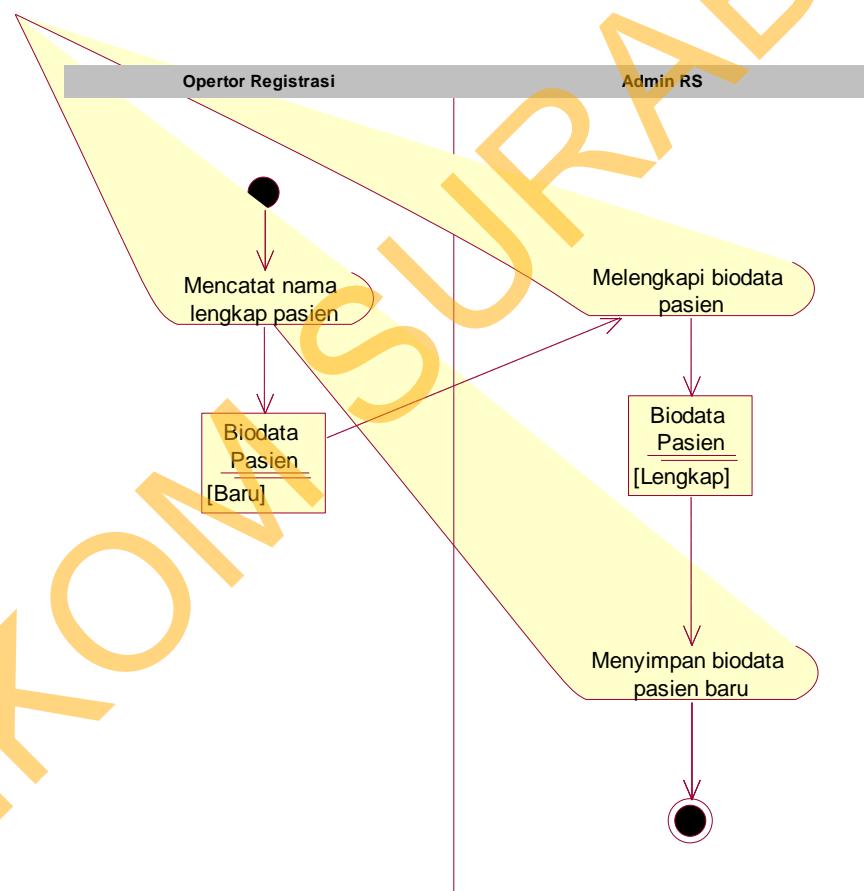


Gambar 3.7 *Activity Diagram* untuk Bisnis Use Case Pendaftaran Pasien

E. Activity Diagram untuk Bisnis Use Case Mengolah Biodata Pasien

Pada gambar 3.8 terdapat dua *swimlane* yaitu operator registrasi dan admin RS. Proses yang terjadi adalah operator registrasi yang menerima pendaftaran pasien akan mencatat identitas pasien yang melakukan pendaftaran. Biodata yang dicatat masih belum lengkap untuk kemudian dilengkapi oleh admin RS dan kemudian disimpan ke dalam database terpusat.

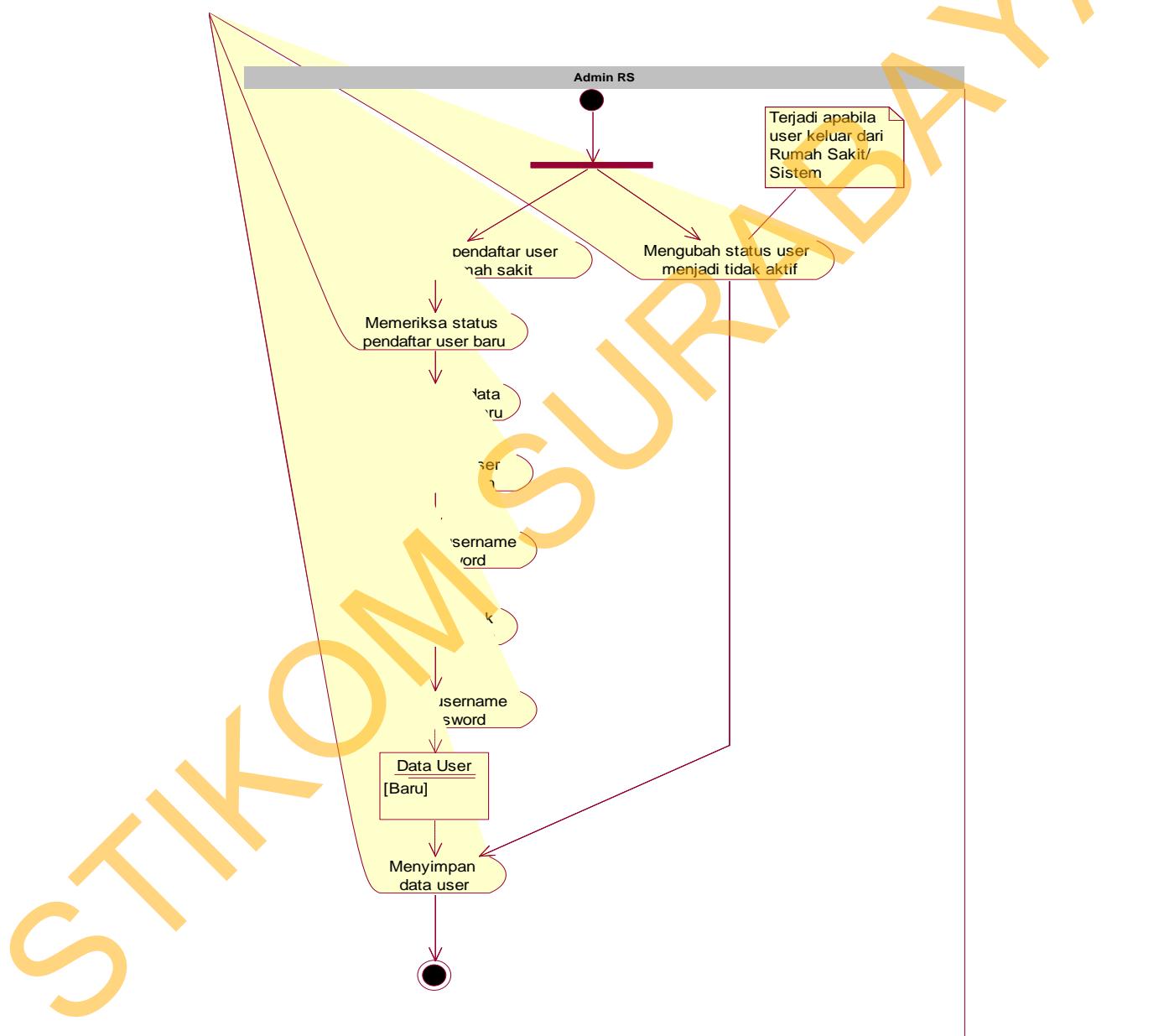
Untuk lebih jelasnya proses mengolah biodata pasien dapat dilihat pada gambar 3.8.



Gambar 3.8 *Activity Diagram* untuk Bisnis Use Case Mengolah Biodata Pasien

F. Activity Diagram untuk Bisnis Use Case Mengolah User Sistem di RS

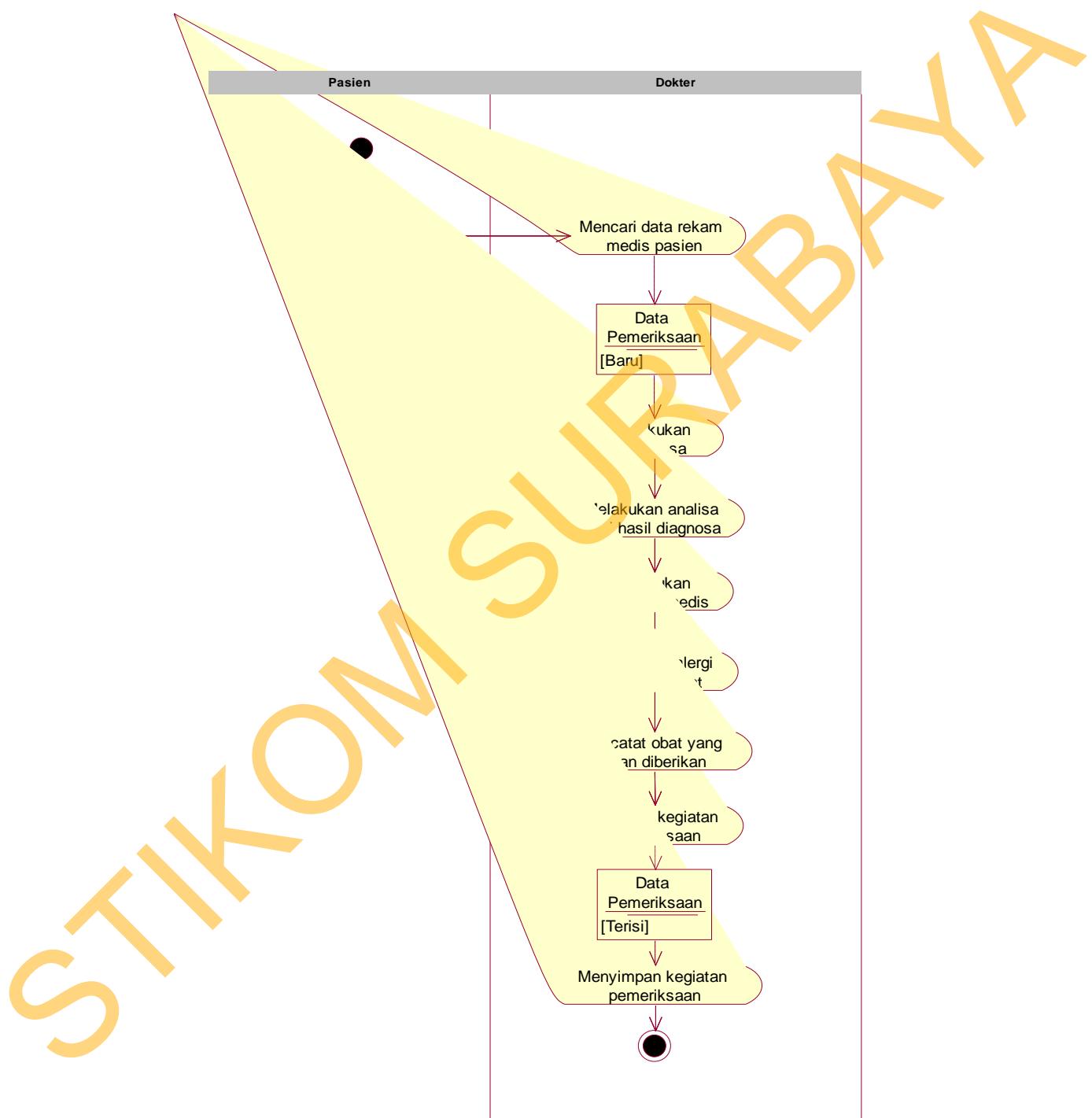
Aktivitas yang terjadi pada mengolah user sistem RS dilakukan oleh admin RS. Mulai dari proses menambahkan user sistem baru di rumah sakit sampai dengan mengubah status menjadi non aktif karena user sistem tersebut berhenti atau keluar dari sistem. Untuk lebih jelasnya dapat dilihat gambar 3.9.



Gambar 3.9 *Activity Diagram* untuk Bisnis Use Case Mengolah User Sistem di RS

G. Activity Diagram untuk Bisnis Use Case Pemeriksaan Pasien

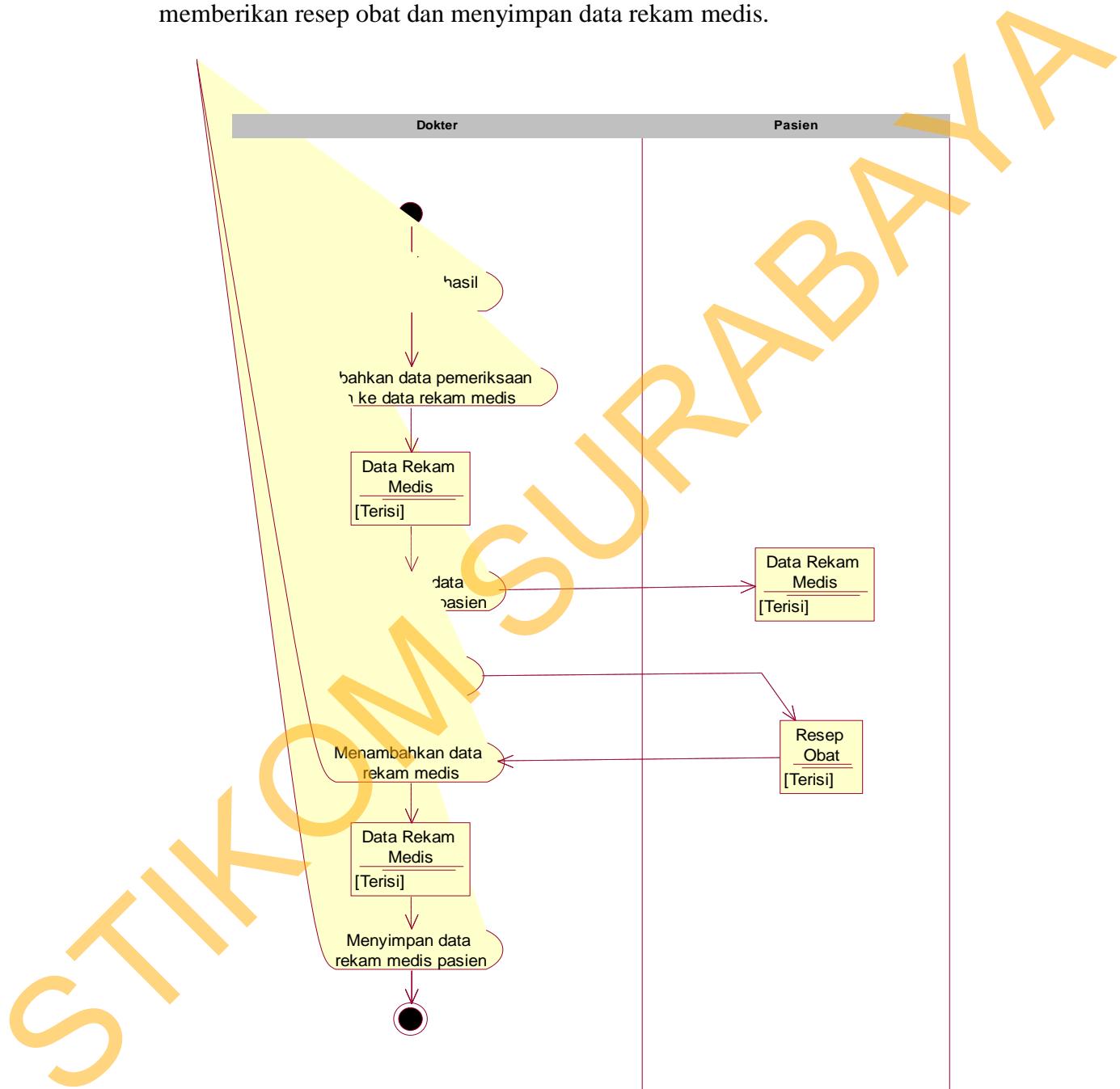
Pada gambar 3.10 akan digambarkan proses pemeriksaan terhadap pasien. Mulai dari proses diagnosa sampai dengan penyimpanan data rekam medis ke database terpusat.



Gambar 3.10 *Activity Diagram* untuk Bisnis Use Case Pemeriksaan Pasien

H. Activity Diagram untuk Bisnis Use Case Mengolah Data Rekam Medis

Pada gambar 3.11 digambarkan proses mengolah data rekam kesehatan. Kegiatannya antara lain menambahkan hasil pemeriksaan ke data rekam medis, memberikan resep obat dan menyimpan data rekam medis.



Gambar 3.11 Diagram Aktivitas untuk Bisnis Use Case Mengolah Data Rekam Medis

3.4.3 Use Case Diagram

Use case diagram menggambarkan interaksi antara use case dan actor. Use case diagram menggambarkan ruang lingkup sistem yang dibangun untuk sistem informasi rekam kesehatan elektronik terpusat di Kodya Denpasar. Use case juga bisa juga menggambarkan bagaimana seseorang sebagai pengguna berinteraksi dengan sistem. Use case memberikan spesifikasi fungsi-fungsi yang ditawarkan oleh sistem dari perspektif user, sedangkan aktor adalah sebuah peran yang bisa dimainkan oleh pengguna dalam interaksi dengan sistem dapat berupa orang atau apa saja yang berhubungan dengan sistem yang sedang dibangun (bisa di luar organisasi ataupun di dalam organisasi).

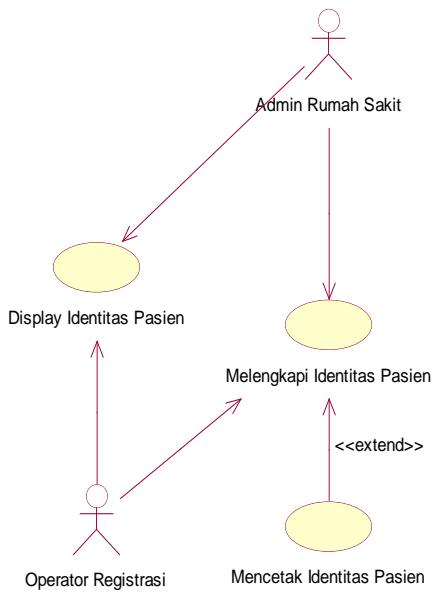
Sebelum menggambarkan diagram *use case* tersebut, dilakukan sebuah pemetaan *business use case* ke *use case* sistem dalam sebuah table. Pemetaan tersebut dapat dilihat pada table dibawah 3.2

Tabel 3.2 Pemetaan *Business Use Case* ke *Use Case*.

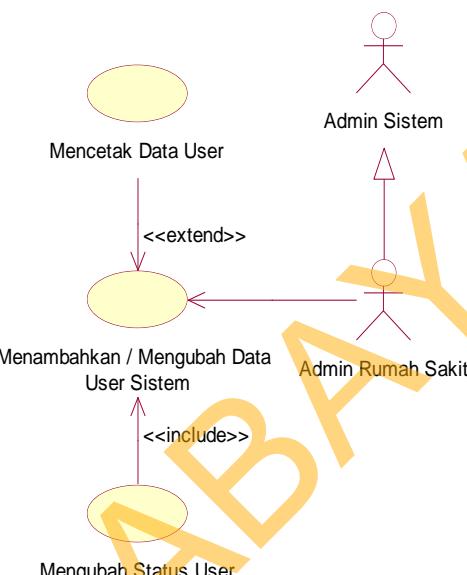
Bisnis Use Case	Use Case Sistem
Melakukan Login	Melakukan Login
Mengolah user sistem	Menambahkan/mengubah data user sistem Mengubah status user Mencetak data user
Pendaftaran integrasi sistem	Pendaftaran/perubahan integrasi sistem Menambahkan/mengubah data rumah sakit Mengubah status RS
Mengolah data rekam medis	Display data rekam kesehatan Menambahkan data rekam kesehatan Mencetak hasil pemeriksaan pasien
Pendaftaran pasien	Menyerahkan identitas (ID) Menyerahkan nomer RM Melakukan pendaftaran pasien baru Memeriksa status pasien Menambahkan pendaftaran pasien Memeriksa rujukan

	Mencetak nomer antrean dan dokter tujuan
Mengolah identitas pasien	Display identitas pasien
	Melengkapi identitas pasien
	Mencetak identitas pasien
Pemeriksaan pasien	Menyerahkan nomer antrean dan ID
	Melakukan pemeriksaan
	Melakukan tindakan medis
	Mencetak resep obat
	Mencetak hasil pemeriksaan
	Menambahkan data ream medis pasien

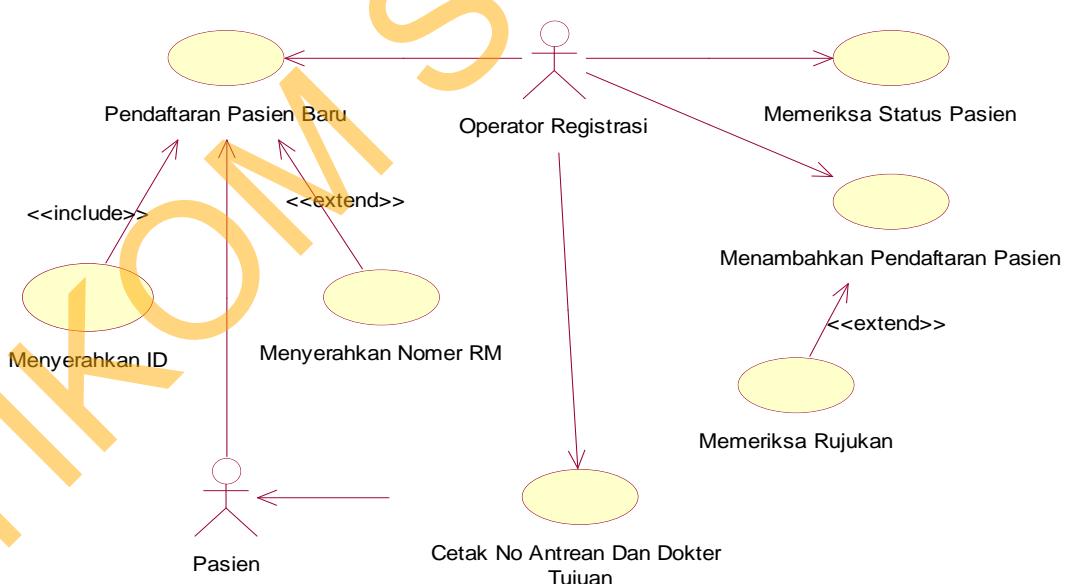
Dari tabel pemetaan diatas, dapat dilihat bahwa 8 *business use case* yang ada akan memiliki 24 use case sistem. Dan use case akan digambarkan dalam gambar 3.12 yaitu use case diagram sistem informasi rekam kesehatan elektronik terpusat di Kota Madya Denpasar.



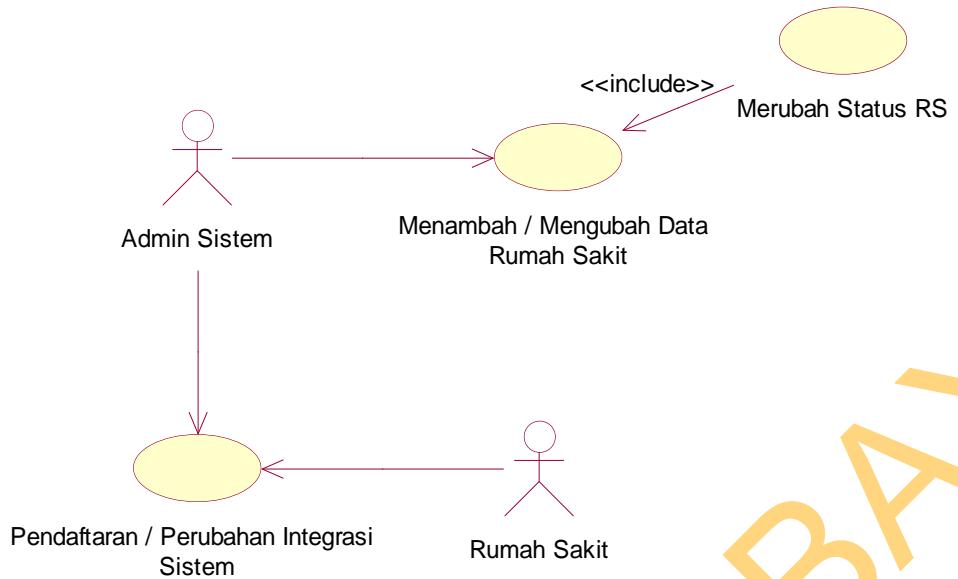
Gambar 3.12 Use Case Mengolah
Identitas Pasien



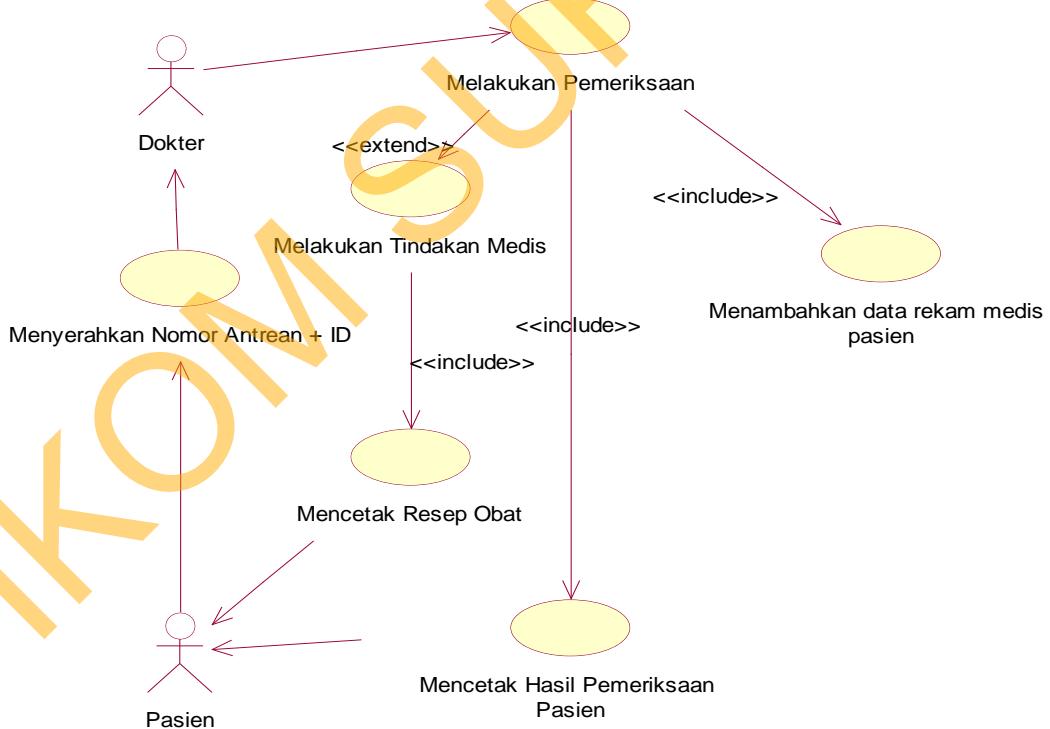
Gambar 3.13 Use Case Mengolah
User Sistem



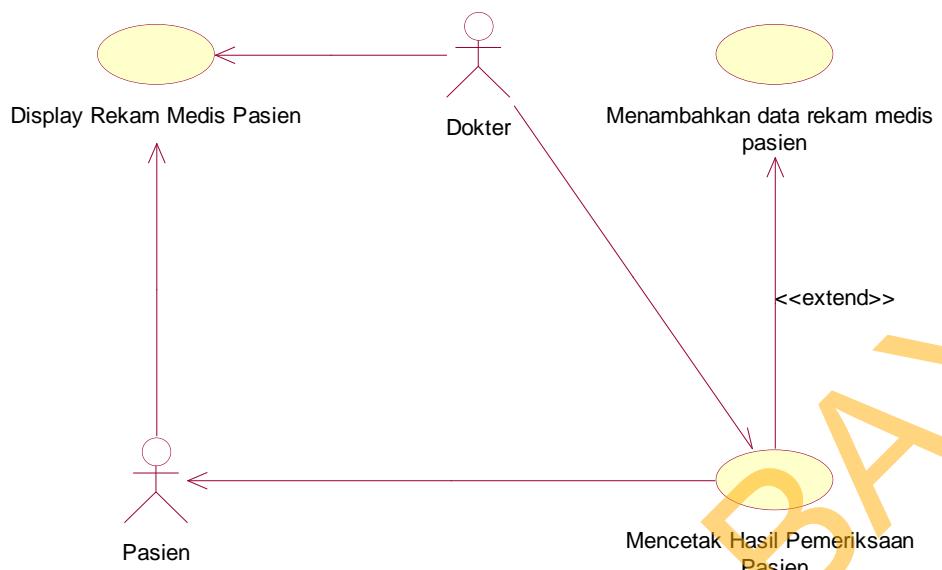
Gambar 3.14 Use Case Pendaftaran Integrasi Sistem



Gambar 3.15 Use Case Pendaftaran Pasien



Gambar 3.16 Use Case Pemeriksaan Pasien



Gambar 3.17 Use Case Mengolah Data Rekam kesehatan



Gambar 3.18 Use Case Melakukan Login

Use case diagram sistem informasi rekam kesehatan elektronik terpusat yang tampak pada gambar 3.12 sampai dengan gambar 3.18. Pada diagram use case ini terdapat enam aktor antara lain dokter, pasien, operator registrasi, rumah sakit, admin rumah sakit, dan admin sistem. Penjelasan interaksi antara actor dengan use case yang terdapat pada use case diagram adalah sebagai berikut :

1. Dokter

Dokter bertugas untuk melakukan pemeriksaan terhadap keluhan-keluhan seorang pasien yang datang ke rumah sakit. Di rumah sakit dokter di wajibkan mengisi dan menambahkan data rekam kesehatan setelah melakukan pemeriksaan agar seluruh kegiatan yang dilakukan oleh seorang dokter terhadap pasien dapat tercatat.

2. Pasien

Pasien merupakan aktor yang menjadi sumber data rekam medis. Pasien yang sakit memerlukan penanganan kesehatan ke rumah sakit. Untuk mendapatkan pelayanan kesehatan seorang pasien harus melakukan pendaftaran terlebih dahulu pada loket pendaftaran. Setelah terdaftar pada sistem pasien akan diarahkan menuju poliklinik terkait atau rujukan yang dituju. Pasien akan diwawancara keluhannya apa untuk kemudian didiagnosa agar bisa dilakukan penanganan kesehatan. Semua kegiatan dan jawaban pasien akan dicatat oleh dokter.

3. Operator registrasi

Operator registrasi bertugas menerima pendaftaran pasien yang ingin mendapatkan pelayanan kesehatan. Untuk pasien lama, operator registrasi akan mencari identitas pasien berdasarkan nomer rekam medisnya dan menambahkan tanggal pendaftarannya untuk selanjutnya diarahkan ke poliklinik yang dituju. Untuk pasien baru, operator registrasi meminta identitas pasien untuk didaftarkan ke dalam sistem dan mendapatkan nomer rekam medis. Setelah data tersimpan dan status pasien telah terdaftar, pasien dilanjutkan ke poliklinik untuk mendapatkan pelayanan kesehatan.

4. Rumah Sakit

Rumah sakit bertugas mendaftarkan diri ke dalam sistem rekam medis terpusat. Rumah sakit melakukan pendaftaran dengan mengisi formulir pendaftaran untuk kemudian diserahkan kepada admin sistem. Admin sistem akan memeriksa kelengkapan pendaftaran apabila sudah lengkap rumah sakit akan didaftarkan ke dalam integrasi sistem.

5. Admin Rumah Sakit

Admin RS bertugas untuk mendaftarkan user-user yang akan menggunakan sistem yang ada dirumah sakit yang ditangani. User-user yang didaftarkan antara lain operator registrasi dan dokter. Selain bertugas membantu admin sistem dalam mendaftarkan user ke dalam integrasi sistem, admin RS juga membantu operator registrasi pada proses pendaftaran untuk melengkapi identitas pasien yang tidak bisa sepenuhnya dilakukan oleh operator registrasi. Hal ini akan membantu proses pendaftaran akan lebih cepat karena operator registrasi hanya akan mendaftarkan data pasien dengan mencatat identitas utama diantaranya nama, tempat lahir, tanggal lahir, alamat, dan sebagainya. Untuk selanjutnya umur dan keterangan lain dibantu oleh admin RS.

6. Admin Sistem

Admin sistem merupakan admin pusat yang bertugas untuk menerima pendaftaran rumah sakit yang akan bergabung ke dalam integrasi sistem. Segala perubahan untuk user sistem dilakukan sepenuhnya oleh admin sistem. Untuk membantu dalam mengolah user sistem, admin pusat membentuk admin RS yang akan diposisikan di masing-masing rumah sakit yang terdaftar pada integrasi sistem. Tugas admin RS disini untuk mendaftarkan user sistem yang ada di rumah sakit.

Dalam gambar 3.12 sampai dengan gambar 3.18 juga terdapat beberapa use case antara lain :

1. Login

Use case ini akan dilakukan oleh semua user yang akan berinteraksi dengan sistem informasi ini. Setiap user akan mengakses sistem informasi rekam kesehatan elektronik terpusat kemudian akan tampil halaman login. User

menginputkan username dan password kemudian akan dilakukan proses validasi apakah username dan password sudah sesuai.

2. Display Identitas Pasien

User akan menampilkan identitas pasien yang dicari berdasarkan identitas atau nomer rekam medis (apabila telah terdaftar sebelumnya pada Sistem Rekam kesehatan Elektronik Terpusat). Sistem akan menampilkan identitas pasien pada layar.

3. Melengkapi Identitas Pasien

Operator registrasi, admin RS dan petugas terkait akan membantu melengkapi identitas pasien baru yang melakukan pendaftaran di rumah sakit yang menjadi tempat tujuan pasien tersebut. Identitas pasien akan dilengkapi berdasarkan salinan atau photocopy identitas yang diserahkan pada saat pendaftaran.

4. Mencetak Identitas Pasien

Setelah identitas dilengkapi, data identitas pasien akan dicetak di tempat pendaftaran. Pencetakan data identitas pasien merupakan optional dimana dapat dilakukan saat tertentu apabila diperlukan untuk keperluan administrasi dan sebagainya dengan sepengetahuan pasien yang bersangkutan.

5. Menambahkan/Mengubah Data User Sistem

Pada use case ini dilakukan untuk penambahan user sistem yang menggunakan Sistem Rekam kesehatan Elektronik Terpusat. Selain itu juga melakukan perubahan data user sistem apabila terjadi kesalahan penginputan atau perubahan data user. Admin sistem memiliki otoritas penuh untuk melakukan peran ini dibantu oleh Admin Rumah Sakit di masing-masing rumah sakit.

6. Mengubah Status User

Apabila user yang keluar dari Sistem Rekam kesehatan Elektronik Terpusat, status user akan diganti menjadi nonaktif tanpa melakukan penghapusan data user. Perubahan tanpa penghapusan status ini dilakukan agar untuk ke depannya apabila terjadi masalah dengan user yang keluar dapat diketahui histori yang terjadi sebelumnya.

7. Mencetak Data User

Use case ini merupakan use case optional atau pilihan dimana dapat dilakukan cetak data user kapan saja diperlukan. Namun untuk user sistem yang baru melakukan pendaftaran, setelah melakukan pendaftaran akan langsung dicetakkan untuk arsip user itu sendiri.

8. Menambahkan/Mengubah Data Rumah Sakit

Pada saat pendaftaran integrasi sistem, data rumah sakit yang melakukan pendaftaran akan ditambahkan ke dalam Sistem Rekam kesehatan Elektronik Terpusat. Data-data yang dimaksud yaitu identitas pasien, fasilitas yang dimiliki rumah sakit dan sebagainya. Apabila terjadi perubahan data rumah sakit, akan dilakukan juga pada use case ini.

9. Pendaftaran/Perubahan Integrasi Sistem

Proses pendaftaran integrasi sistem dimulai dengan pengajuan permohonan oleh pihak rumah sakit yang melakukan pendaftaran. Apabila rumah sakit memenuhi kelengkapan pendaftaran, rumah sakit akan ditambahkan ke dalam tabel rumah sakit yang tergabung dalam sistem terintegrasi.

10. Mengubah Status Rumah Sakit

Use case ini menggambarkan apabila terjadi perubahan status rumah sakit dimana rumah sakit yang bersangkutan sedang melakukan rehabilitasi atau perbaikan fasilitas, bahkan apabila rumah sakit tersebut keluar dari Sistem Rekam kesehatan Elektronik Terpusat status akan diperbahari sesuai dengan keadaan yang terjadi. Perubahan status ini dilakukan untuk mencatat dan mengetahui histori apabila diperlukan di kemudian hari.

11. Pendaftaran Pasien Baru

Use case ini menggambarkan dimana seorang pasien dating ke loket pendaftaran untuk melakukan pendaftaran dan registrasi agar mendapatkan pelayanan kesehatan di rumah sakit yang menjadi tujuannya.

12. Menyerahkan Identitas (ID)

Untuk pasien baru yang melakukan pendaftaran, petugas akan meminta identitas pasien. Identitas dapat berupa KTP, SIM, dan sebagainya. Untuk anak-anak dapat menyerahkan identitas berupa akta kelahiran.

13. Menyerahkan Nomer Rekam Medis (RM)

Pada use case ini menggambarkan kejadian apabila pasien yang datang ke loket pendaftaran merupakan pasien lama yang telah terdaftar pada sistem sehingga telah memiliki nomer rekam medis (RM). Setelah pasien datang ke loket dan menyerahkan nomer RM, maka akan dilakukan registrasi ulang oleh petugas operator registrasi sehingga pasien akan terdaftar dan akan mendapat pelayanan kesehatan di rumah sakit yang bersangkutan.

14. Cetak Nomer Antrean dan Dokter Tujuan

Setelah melakukan pendaftaran operator yang menerima pendaftaran akan mencetakkan nomer antrean dan dokter yang akan dituju sesuai dengan keluhan pasien untuk segera mendapatkan pelayanan kesehatan.

15. Memeriksa Status Pasien

Operator registrasi akan melakukan pemeriksaan status pasien yang melakukan pendaftaran agar diketahui pasien yang datang merupakan pasien lama atau pasien baru dengan memeriksa data pasien pada sistem.

16. Menambahkan Pendaftaran Pasien

Setelah data pasien ditemukan, data pasien akan diaktifkan pada sistem agar segera mendapat pelayanan kesehatan. Proses ini merupakan registrasi ulang agar data pasien dapat diakses juga nantinya oleh dokter sebagai bukti telah melakukan registrasi. Apabila seorang pasien belum melakukan registrasi maka data pasien tidak dapat diakses dan diupdate oleh dokter yang memeriksa.

17. Memeriksa Rujukan

Pada saat registrasi ulang apabila seorang pasien yang datang dengan membawa surat rujukan dari klinik, puskesmas atau rumah sakit lainnya, petugas akan memeriksa rujukan dan untuk kemudian melampirkan pada data pendaftaran pasien. Rujukan akan mempermudah dan mempercepat dokter dalam melakukan diagnosa dan melakukan pemeriksaan.

18. Menyerahkan Nomer Antrean dan Identitas

Use case ini menggambarkan pasien telah mendatangi dokter yang dituju sesuai dengan keluhannya. Pasien akan menyerahkan nomer antrean dan identitasnya untuk memastikan bahwa pasien telah melakukan registrasi agar dokter dapat membuka dan memperbarui data pemeriksaan pasien.

19. Melakukan Pemeriksaan

Pada use case ini dokter akan melakukan pemeriksaan terhadap pasien. Pemeriksaan tersebut meliputi proses anamnesa (wawancara keluhan pasien), pemeriksaan suhu badan, tekanan darah, kecepatan denyut nadi, keadaan pernafasan dan sebagainya. Setelah itu dokter akan melakukan diagnosa dan menyimpulkan penyakit pasien berdasarkan keluhan dan pemeriksaan yang telah dilakukan.

20. Melakukan Tindakan Medis

Pada saat pemeriksaan apabila diperlukan, dokter akan melakukan beberapa tindakan medis di tempat praktek dengan memberikan suntikan (*injection*) atau obat pencegahan. Tindakan medis selanjutnya dokter akan membuatkan resep obat untuk ditebus di apotek untuk penanganan di rumah. Dari hasil diagnosa dokter juga akan dapat diketahui apakah pasien tersebut diperlukan untuk rawat inap ataupun rawat jalan untuk mendapatkan penanganan selanjutnya.

21. Mencetak Resep Obat

Setelah melakukan proses pemeriksaan, pasien akan menuliskan dan mencetakkan obat sesuai dengan pemeriksaan dan hasil diagnosa yang telah dilakukan. Kemudian pasien dapat menebus obat di apotek yang dituju.

22. Mencetak Hasil Pemeriksaan Pasien

Setelah mencatat seluruh hasil pemeriksaan ke dalam sistem, dokter akan mencetakkan hasil pemeriksaan dan diberikan kepada pasien untuk disimpan. Hasil pemeriksaan tersebut akan sangat berguna apabila nantinya seorang pasien mengalami keluhan penyakit yang sama maka dengan cepat dokter dapat melakukan penanganan.

23. Menambahkan Data Rekam kesehatan

Hasil pemeriksaan yang telah dicatat, akan ditambahkan ke dalam data rekam kesehatan yang berisi hasil anamnesa, pemeriksaan fisik pasien, diagnosa, tindakan medis yang dilakukan serta juga melampirkan resep obat yang telah diberikan. Sehingga data rekam kesehatan tersebut lengkap.

24. Display Rekam kesehatan

Use case ini hanya dapat dilakukan oleh dokter. Rekam medis merupakan data yang bersifat sangat rahasia. Dokter akan memasukkan kode rekam kesehatan yang akan di munculkan ke layar. Maka sistem akan menampilkan data rekam medis yang diinginkan.

3.4.4 *Flow Of Event*

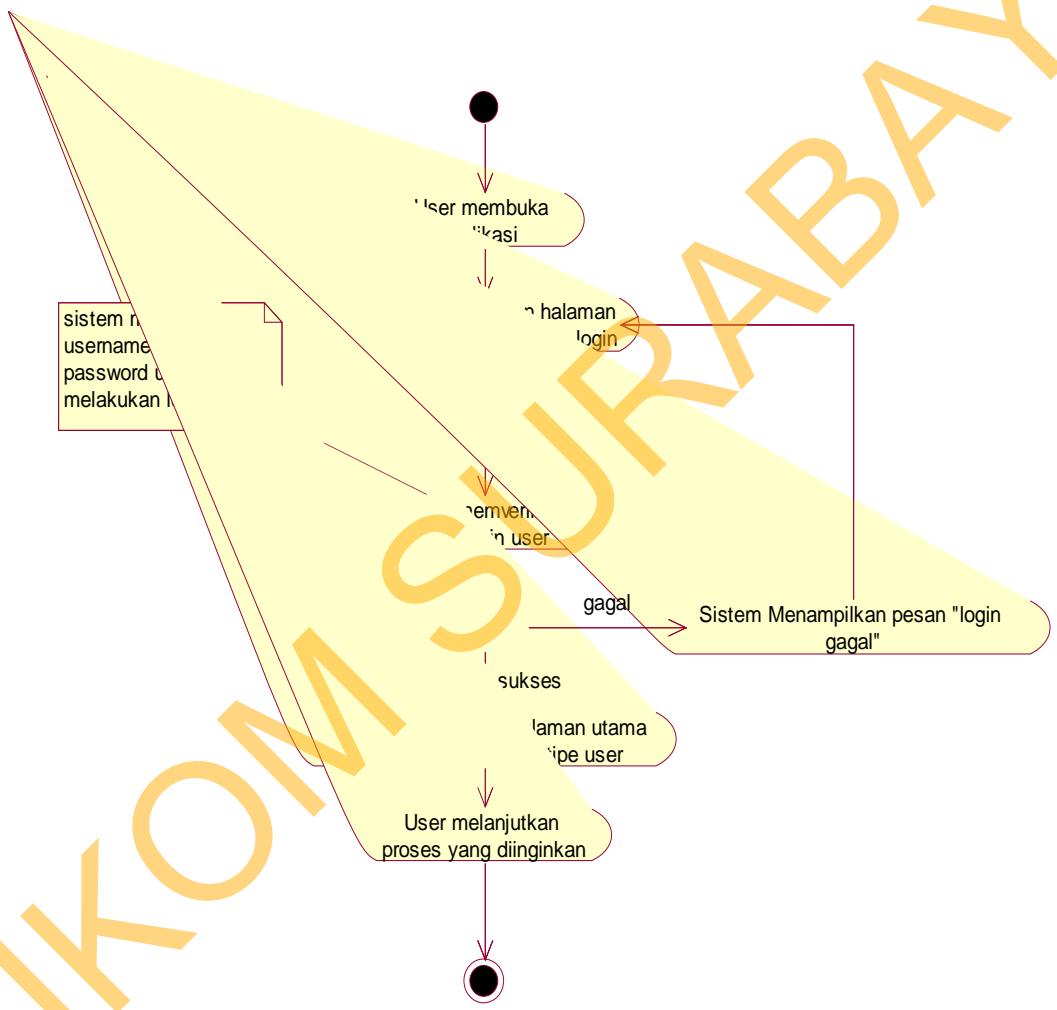
Dalam membangun sebuah sistem diperlukan diagram yang menjelaskan detail aktivitas yang terjadi untuk setiap use case yang ada. *Flow of event* ini digunakan untuk menggambarkan aliran kejadian pada use case dan menguraikan secara detail apa yang user lakukan dan apa yang sistem itu sendiri akan lakukan.

Dalam *flow of event* ini dapat diasumsikan bahwa aktivitas yang ada menggambarkan otomatisasi sistem informasi. Bagaimanapun juga, pembuatan *flow of event* tidak berkaitan dengan apakah sistem akan dibangun nantinya, baik java, PHP, ASP, .NET. Tujuandari pembuatan *flow of event* ini adalah menggambarkan apa yang sistem akan lakukan, bukan bagaimana sistem akan melakukannya.

Flow of event yang ada pada sistem informasi ini akan digambarkan dengan menggunakan *activity* diagram. Model *activity* bisa digunakan untuk mewakili secara grafis aliran kejadian (*flow of event*). Karena desain sistem informasi ini merupakan rancangan untuk sistem informasi yang terintegrasi, maka ada penggunaan kata-kata dan istilah terpusat dan integrasi sistem. *Flow of event* yang ada antara lain sebagai berikut :

A. *Flow Of Event* Melakukan Login

Aktivitas login akan dilakukan oleh semua user yang berinteraksi dengan sistem informasi ini. Pada gambar 3.19 dibawah ini aktivitas akan dimulai ketika user membuka aplikasi Sistem Informasi Rekam kesehatan Elektronik Terpusat kemudian sistem akan menampilkan halaman selamat datang dan form login.



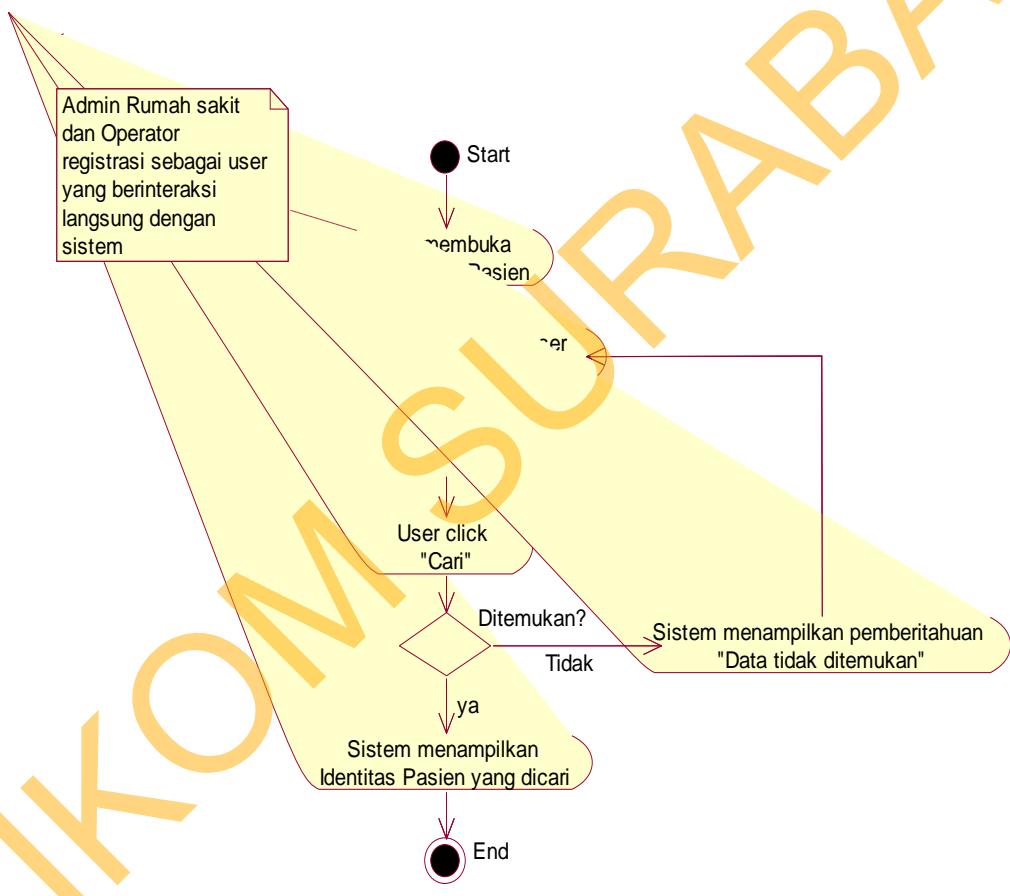
Gambar 3.19 *Flow Of Event* Melakukan Login

User selanjutnya menginputkan username dan password kemudian sistem akan melakukan validasi apakah *username* dan *password* sesuai. Apabila *username* dan *password* sesuai, sistem akan menampilkan halaman selanjutnya sesuai dengan tipe user yang melakukan login sehingga user dapat melanjutkan proses selanjutnya. Apabila *username* dan *password* tidak sesuai maka sistem

akan menampilkan pesan “login gagal” dan sistem akan menampilkan form login agar user memasukkan kembali *username* serta *password*.

B. *Flow Of Event Display Identitas Pasien*

Display identitas pasien merupakan use case yang dilakukan oleh operator registrasi dan admin rumah sakit untuk mengetahui detail identitas pasien yang ingin ditampilkan. Pada gambar 3.20 aktivitas yang dilakukan dimulai dengan membuka menu data pasien pada sistem. Pada menu data pasien user membuka identitas pasien dengan melakukan pencarian berdasarkan kode pasien.

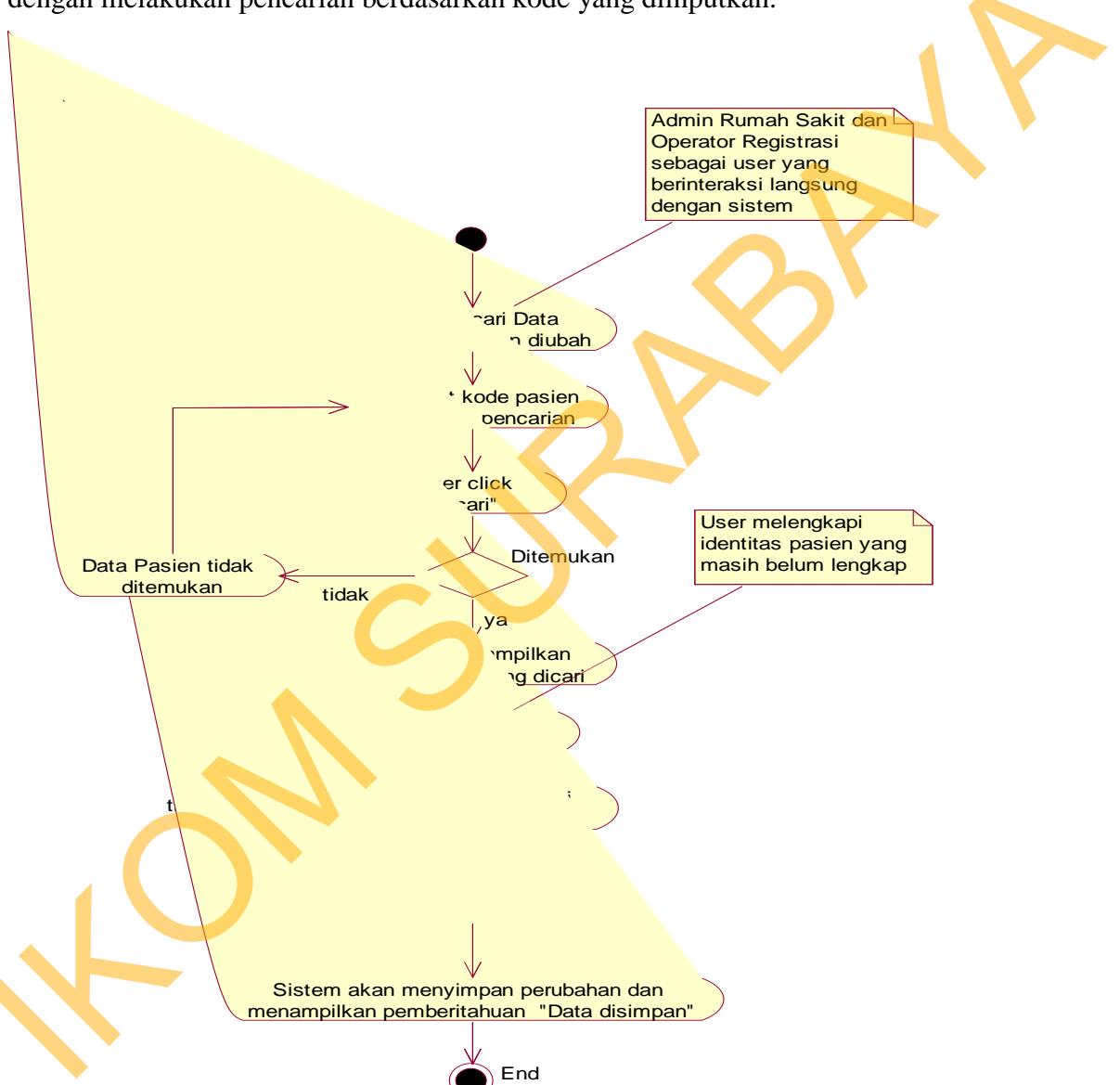


Gambar 3.20 *Flow Of Event Display Identitas Pasien*

Setelah memasukkan kode pasien, sistem akan melakukan pencarian data identitas pasien sesuai dengan kode yang dimasukkan. Apabila data pasien ditemukan, sistem akan menampilkan identitas pasiennya sedangkan apabila data pasien tidak ditemukan maka sistem akan menampilkan pesan “Data Tidak Ditemukan”.

C. *Flow Of Event Melengkapi Identitas Pasien*

Pada use case ini yang berinteraksi langsung dengan sistem adalah operator registrasi dan juga admin rumah sakit. Pada gambar 3.21 aktivitas yang terjadi dimulai dari user yang membuka data identitas pasien terlebih dahulu dengan melakukan pencarian berdasarkan kode yang diinputkan.



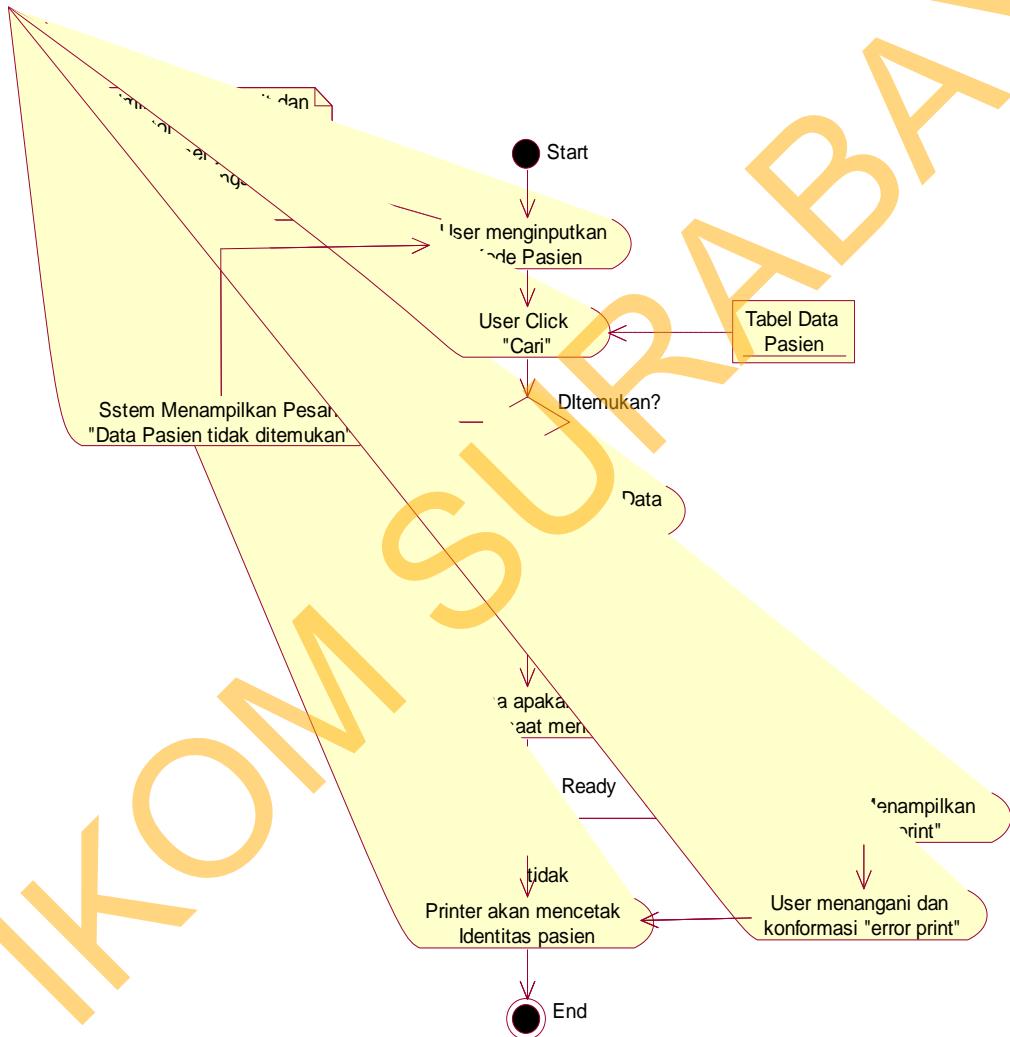
Gambar 3.21 *Flow Of Event Melengkapi Identitas Pasien*

Setelah data ditemukan, user memilih menu edit data pasien untuk menambahkan data yang masih kurang dan juga memperbaiki apabila terjadi kesalahan pengetikan atau inputan. Apabila perubahan telah selesai dilakukan, user menyimpan perubahan dan penambahan sehingga data pasien menjadi

lengkap, kemudian sistem menyimpan ke dalam database data pasien dan menampilkan pesan “Data Disimpan”.

D. *Flow Of Event* Mencetak Identitas Pasien

Mencetak identitas pasien dapat dilakukan oleh operator registrasi dan admin rumah sakit. Data identitas pasien yang akan dicetak ditampilkan dahulu kemudian user memilih menu cetak.



Gambar 3.22 *Flow Of Event* Mencetak Identitas Pasien

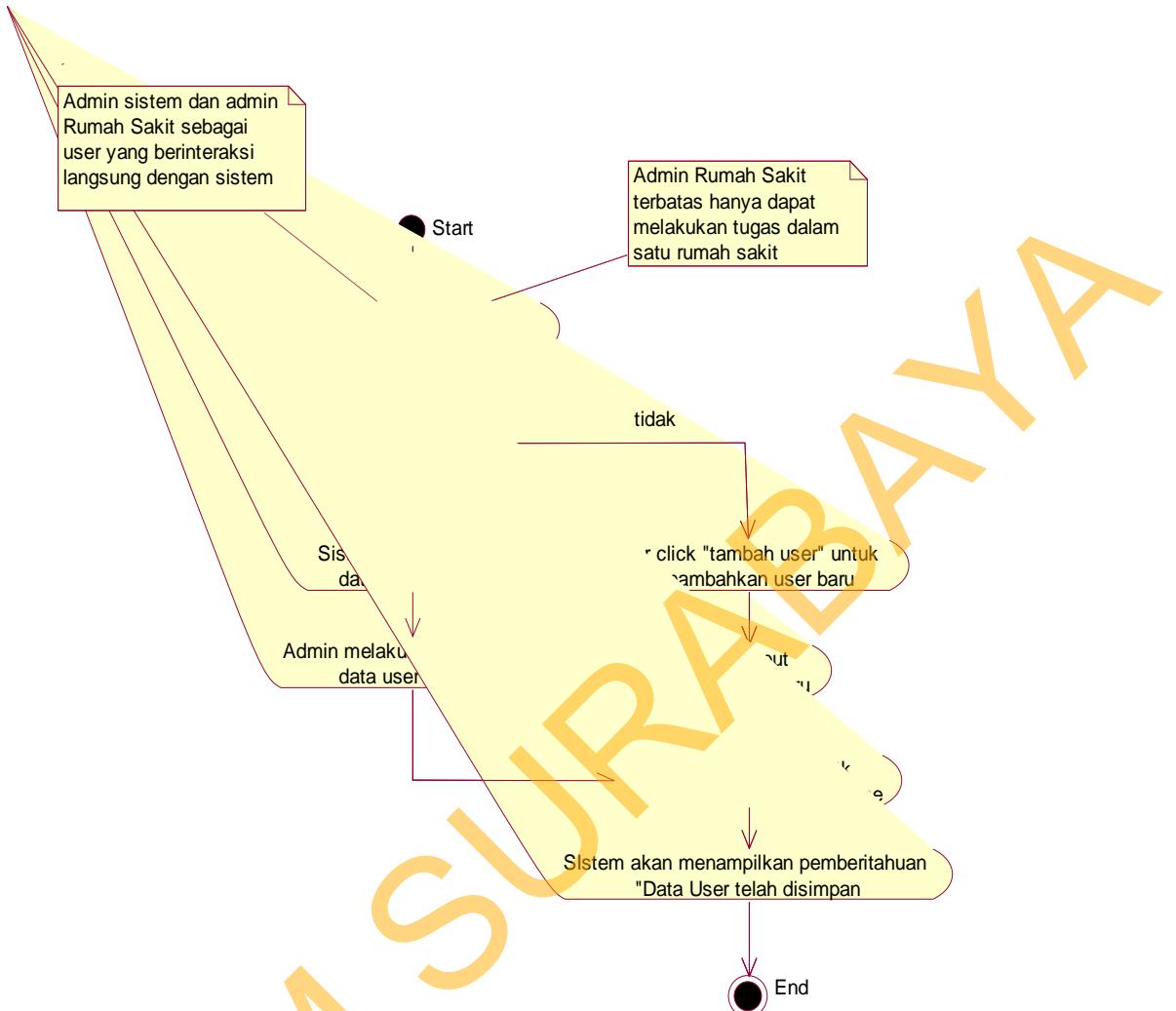
Pada gambar 3.22 aktivitas setelah user memilih menu cetak adalah memeriksa data yang akan dicetak dan juga memeriksa perangkat pencetak apakah sudah siap mencetak atau belum. Apabila telah siap data akan dicetak sesuai dengan pengaturan yang dilakukan user.

E. *Flow Of Event* Menambahkan/Mengubah Data User Sistem

Use case ini menggambarkan proses penambahan atau perubahan data user sistem. Admin sistem dan admin RS memiliki peranan dalam proses ini namun kinerja admin RS terbatas hanya dalam satu rumah sakit yang ditanganinya saja. Admin sistem atau admin RS akan membuka menu data user sistem untuk melakukan perubahan atau penambahan data user. Pertama akan dilakukan pencarian user sistem dengan memasukkan identitas user. Jika ditemukan sistem akan menampilkan data user sistem yang dicari untuk kemudian dilakukan proses selanjutnya. Apabila proses perubahan atau perbaikan selesai dilakukan admin sistem atau admin RS mengklik tombol simpan dan sistem akan menyimpan perubahan tersebut ke dalam database.

Untuk penambahan user sistem yang baru, admin sistem dan admin RS memilih menu tambah user. Identitas user baru akan diinputkan ke dalam sistem sesuai dengan petunjuk. Setelah semua data user terisi, admin sistem atau admin RS menyimpan penambahan user tersebut dengan menekan tombol simpan. Sistem akan menambahkan data user ke dalam database.

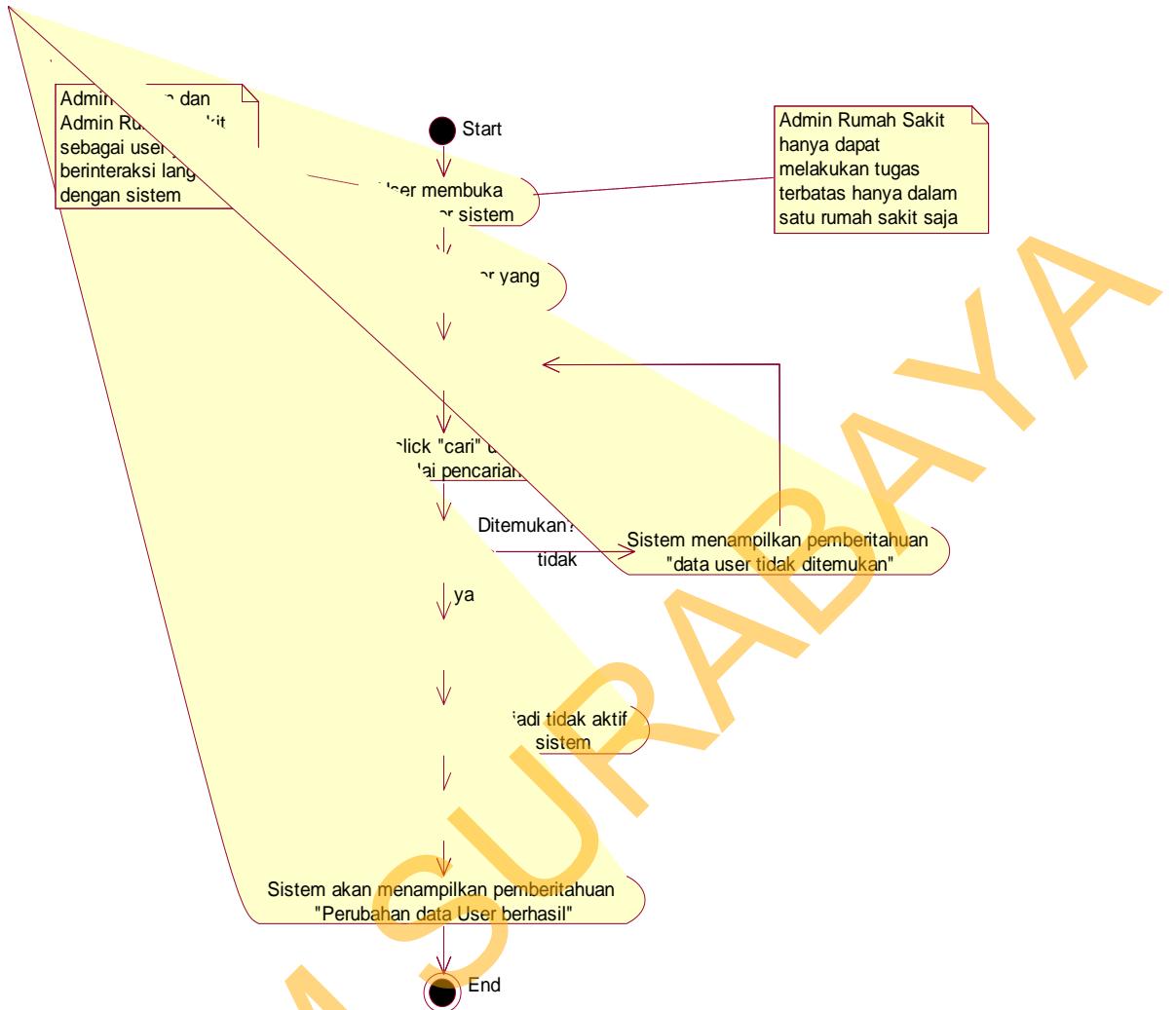
Pada gambar 3.23 di bawah ini akan digambarkan aktivitas perubahan atau penambahan user sistem.



Gambar 3.23 *Flow Of Event* Mengubah atau Menambahkan Data User Sistem

F. *Flow Of Event* Mengubah Status User Sistem

Pada use case ini digambarkan alur kejadian dengan diagram aktivitas yaitu perubahan status user sistem. Aktor yang berperan dalam proses ini adalah admin sistem atau admin RS. Namun admin RS hanya menangani satu rumah sakit saja. Aktivitas dimulai oleh user (admin Sistem/admin RS) membuka menu data user sistem. Kemudian user melakukan pencarian data user yang diinginkan pada menu pencarian dengan menginputkan kode user. apabila data user tidak ditemukan, sistem akan menampilkan pemberitahuan bahwa data user tidak ditemukan, begitu juga sebaliknya apabila data user ditemukan maka sistem akan menampilkan data user yang diinginkan.



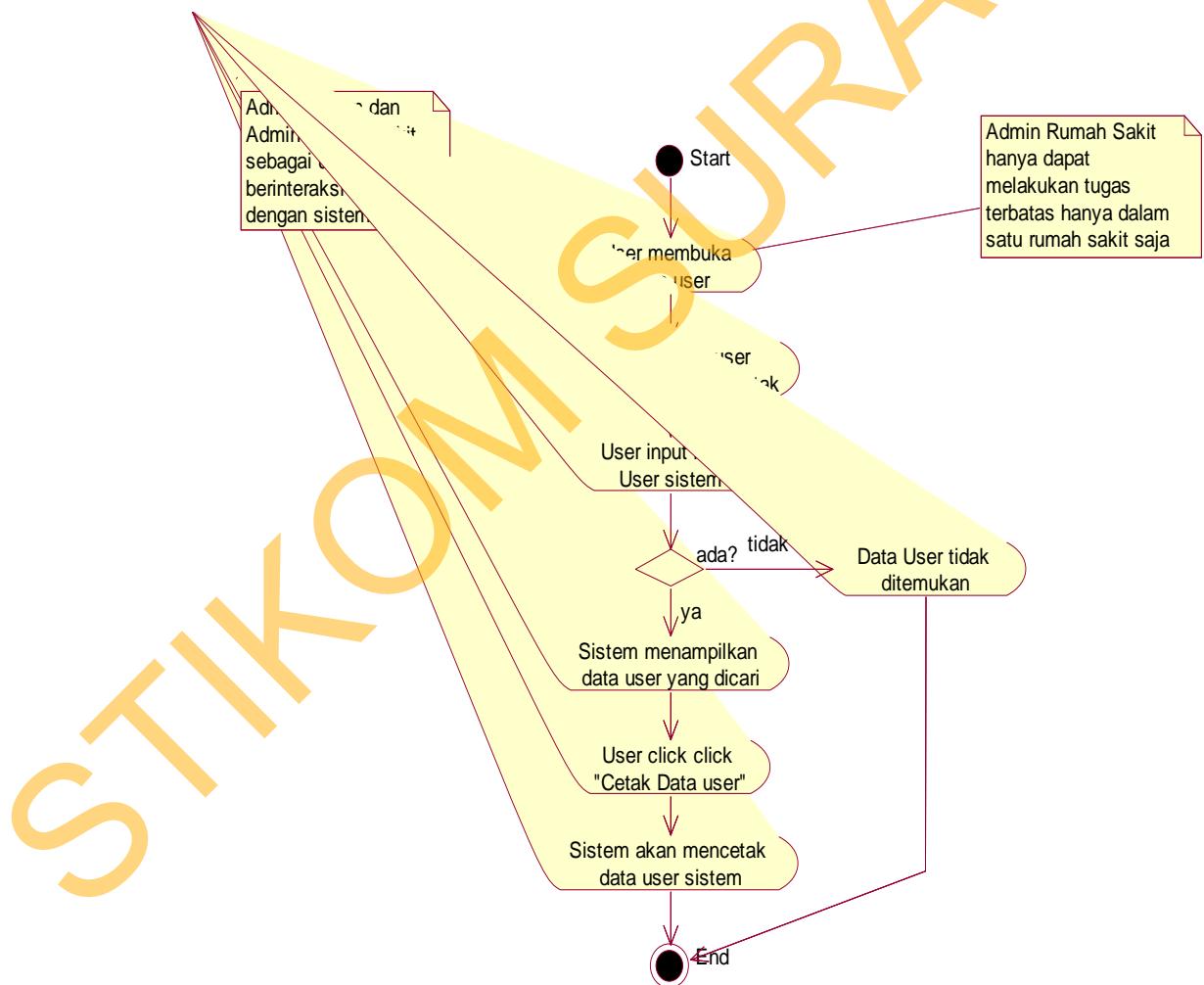
Gambar 3.24 *Flow Of Event* Mengubah Status Data User Sistem

Pada gambar 3.24 aktivitas yang terjadi selanjutnya yaitu perubahan status user yang ditampilkan menjadi tidak aktif. Perubahan status ini dilakukan untuk memutus akses user yang telah keluar dari sistem untuk menghindari hal-hal yang tidak diinginkan. Setelah perubahan selesai dilakukan maka user menyimpan perubahan tersebut dengan memilih tombol simpan. Sistem akan menyimpan perubahan yang terjadi ke dalam database dan menampilkan pemberitahuan perubahan data user berhasil.

G. *Flow Of Event* Mencetak Data User

Use case ini menggambarkan aktivitas mencetak data user sistem. Sama halnya dengan mengubah data user sistem, actor yang berperan disini adalah admin sistem atau admin RS. Kegiatan yang terjadi akan digambarkan dalam diagram aktivitas pada gambar 3.25. Aktivitas dimulai dari user membuka data pasien yang akan dicetak dengan melakukan pencarian pada sistem. Setelah data user ditemukan dan ditampilkan maka user memilih menu cetak data user. Sebelum dicetak sistem akan ditampilkan preview data pasien untuk melakukan pengaturan alat pencetak dan sebagainya. Apabila sudah sesuai, sistem akan melakukan pencetakan data user sistem.

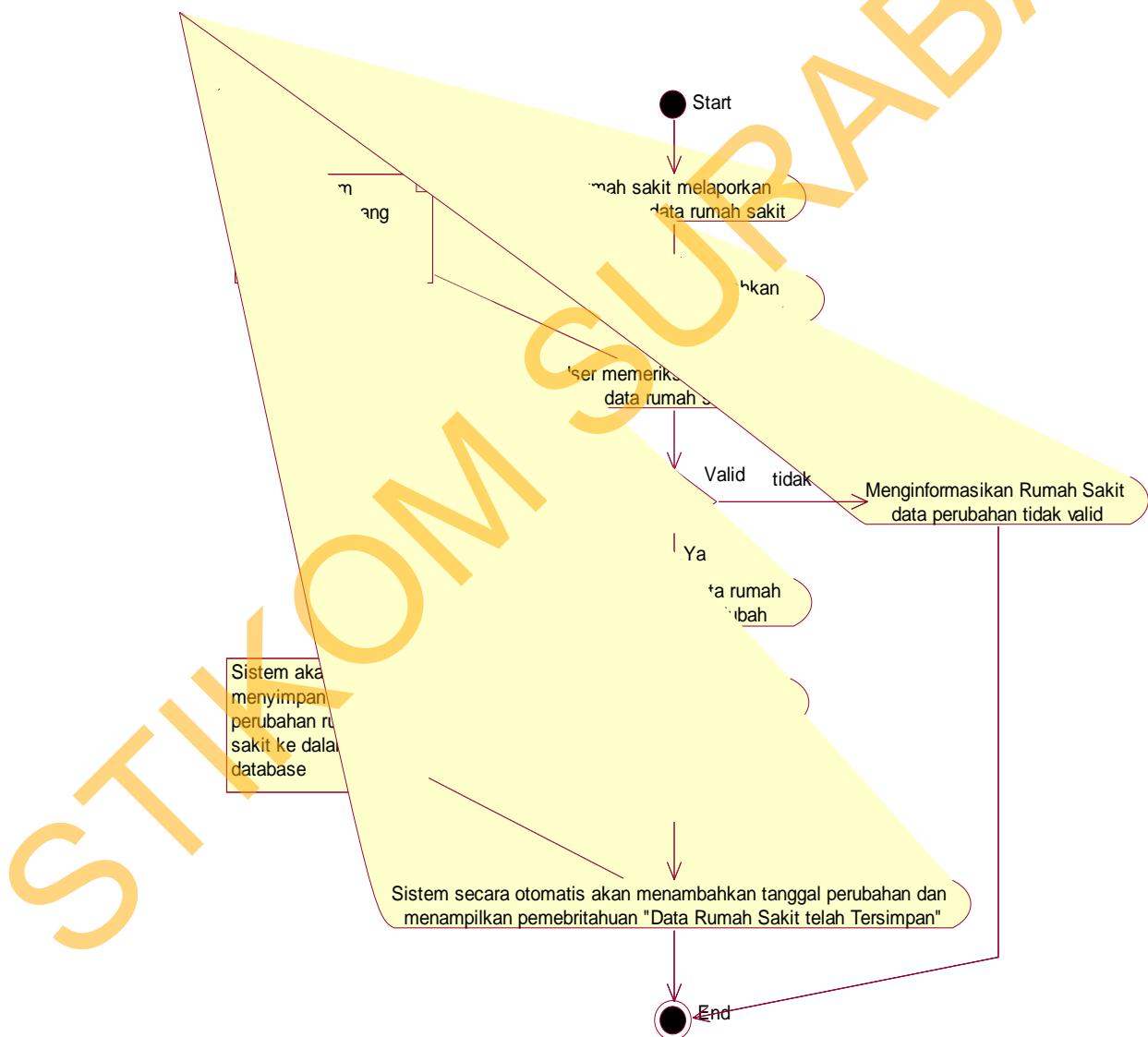
Di bawah ini akan ditampilkan gambar *flow of event* mencetak data user sistem.



Gambar 3.25 *Flow Of Event* Mencetak Data User Sistem

H. Flow Of Event Menambahkan /Mengubah Data Rumah Sakit

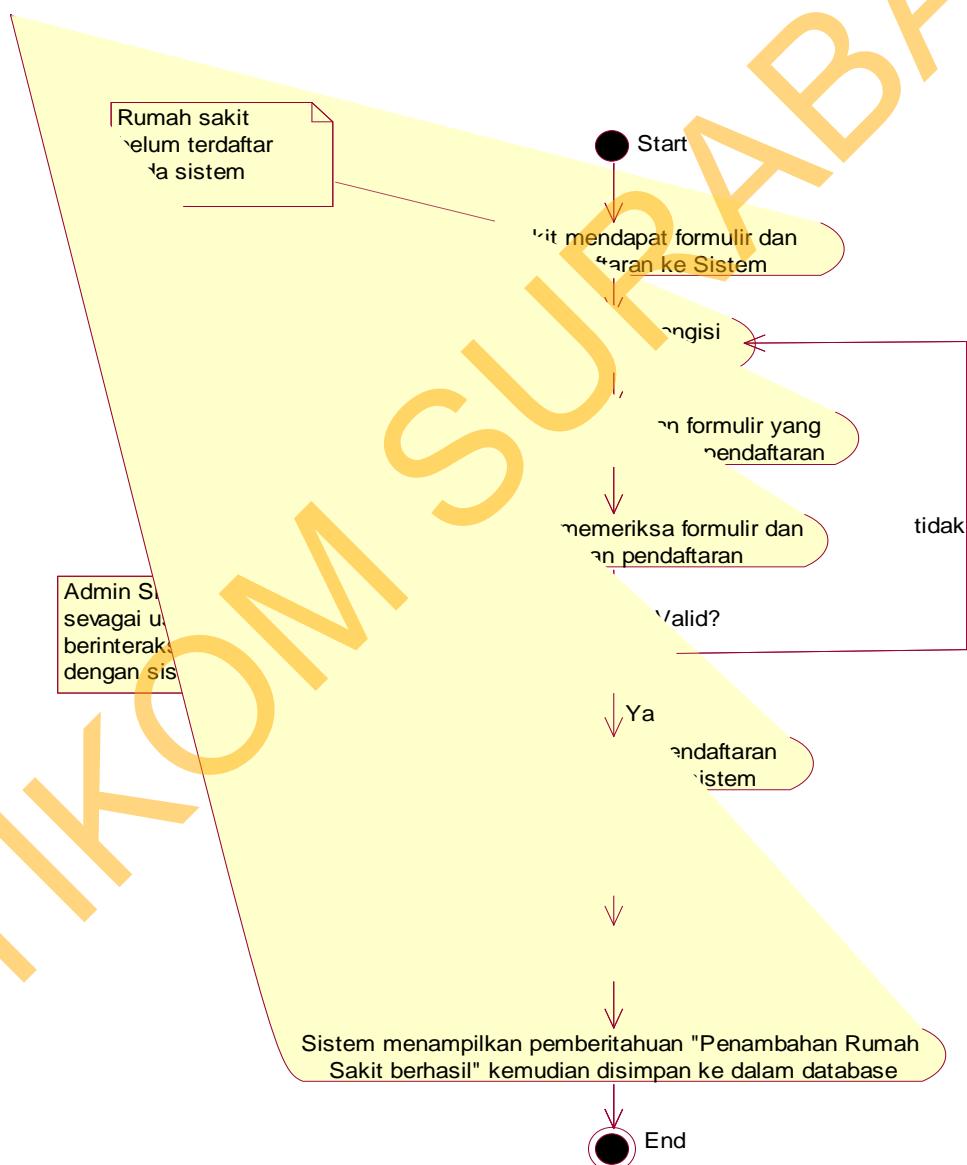
Use case ini menggambarkan proses menambahkan atau mengubah data rumah sakit yang telah bergabung dalam integrasi sistem. Penambahan yang dimaksud disini seperti penambahan fasilitas atau infrastruktur rumah sakit. Admin sistem akan menampilkan data rumah sakit yang akan dirubah. Kemudian perubahan yang diajukan oleh rumah sakit akan dicatat ke dalam database. Perubahan data rumah sakit ini sebelumnya telah dilakukan pemeriksaan kebenarannya dengan yang terjadi dilapangan. Semua perubahan akan disimpan ke dalam database. Pada gambar 3.26 akan digambarkan aktivitas yg terjadi pada kegiatan perubahan data rumah sakit.



Gambar 3.26 Flow Of Event Menambahkan/Mengubah Data Rumah Sakit

I. *Flow Of Event* Pendaftaran/Perubahan Integrasi Sistem

Pada use case ini menggambarkan kegiatan pendaftaran rumah sakit ke dalam integrasi sistem. Aktor yang berperan disini adalah admin sistem. Aktivitas pertama yang dilakukan dimulai dari sebuah rumah sakit yang mengajukan usulan untuk bergabung ke dalam integrasi sistem. Usulan akan diperiksa oleh admin sistem apakah telah memenuhi syarat yang sesuai. Admin sistem akan memberikan formulir pendaftaran untuk diisi oleh pihak rumah sakit yang melakukan pendaftaran beserta lampiran-lampiran yang harus dipenuhi.



Gambar 3.27 *Flow Of Event* Pendaftaran/Perubahan Integrasi Sistem

Pada gambar 3.27 akan digambarkan mengenai aktivitas selanjutnya yang terjadi pada pendaftaran integrasi sistem. Apabila formulir dan prasyarat telah dipenuhi, admin sistem akan menambahkan rumah sakit ke dalam integrasi sistem. Namun apabila rumah sakit yang melakukan pendaftaran belum memenuhi syarat, akan diberikan pemberitahuan bahwa rumah sakit belum memenuhi syarat-syarat tertentu sesuai dengan kekurangan yang belum dipenuhi.

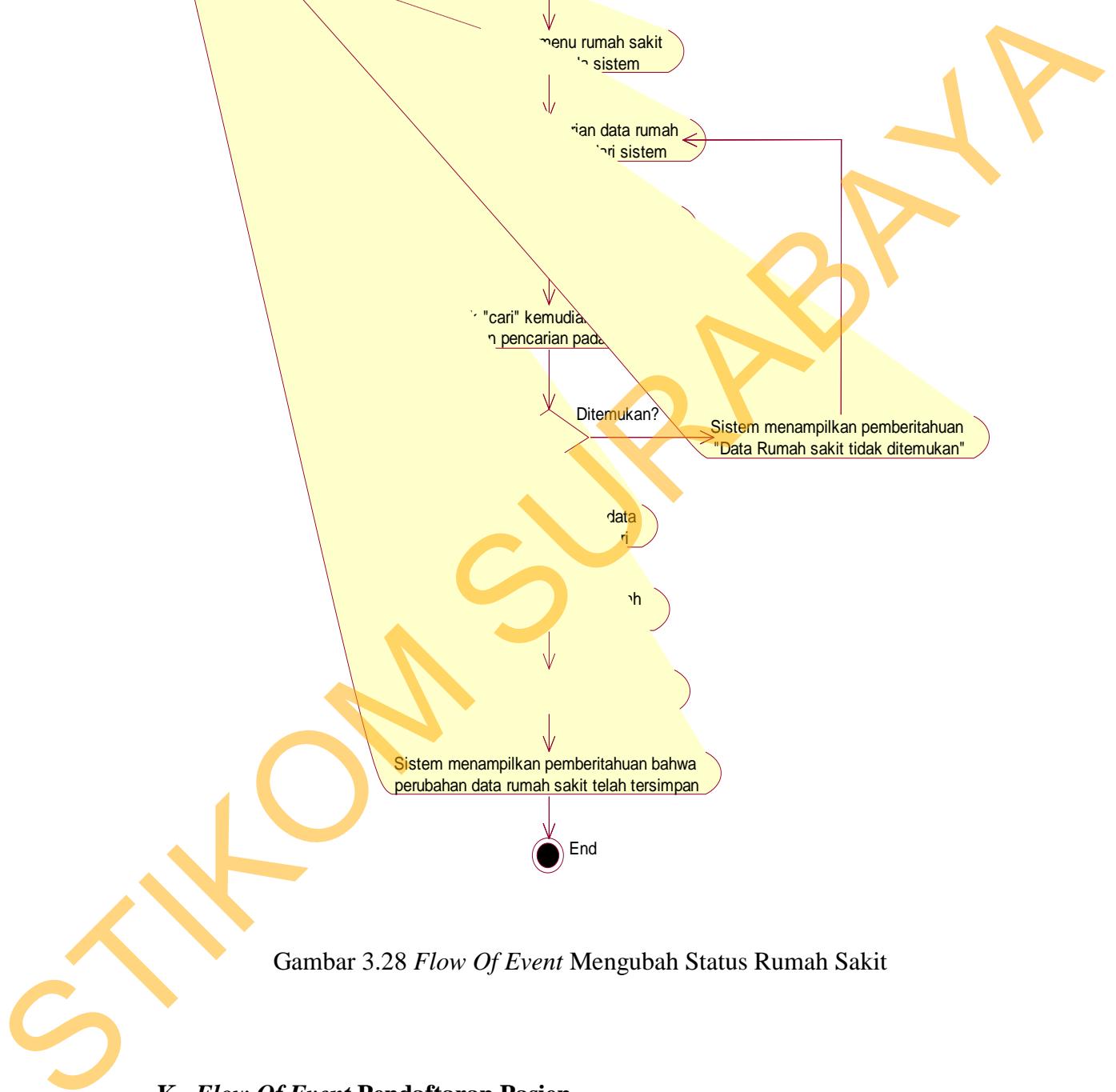
J. Flow Of Event Mengubah Status Rumah Sakit

Aktivitas yang terjadi pada use case ini adalah mengubah status rumah sakit. Use case ini dimaksudkan untuk menangani apabila ada rumah sakit yang keluar dari sistem. Maka status keberadaan rumah sakit di non-aktifkan dari sistem terintegrasi. Aktivitas ini dilakukan oleh admin sistem.

Rumah sakit yang keluar dari sistem akan ditampilkan datanya terlebih dahulu. User yaitu admin sistem akan membuka data rumah sakit yang tergabung dalam integrasi sistem. Data rumah sakit yang akan diubah akan dicari berdasarkan kode yang dimiliki rumah sakit melalui menu pencarian. Setelah data ditemukan, admin sistem akan memeriksa dan mengubah status rumah sakit menjadi tidak aktif. Status rumah sakit ini diubah agar tidak terjadi hal-hal yang tidak diinginkan karena menyangkut data rekam kesehatan yang bersifat sangat rahasia.

Pihak rumah sakit yang telah keluar dari sistem, tidak dapat lagi mengakses sistem. Namun dari sistem riwayat-riwayat yang telah terjadi antara sistem dan rumah sakit sebelumnya masih dapat diketahui dimana artinya sistem masih dapat mengetahui *history* yang terjadi sebelum rumah sakit keluar dari sistem.

Pada gambar 3.28 akan digambarkan proses-proses pengubahan status rumah sakit yang telah keluar dari sistem.

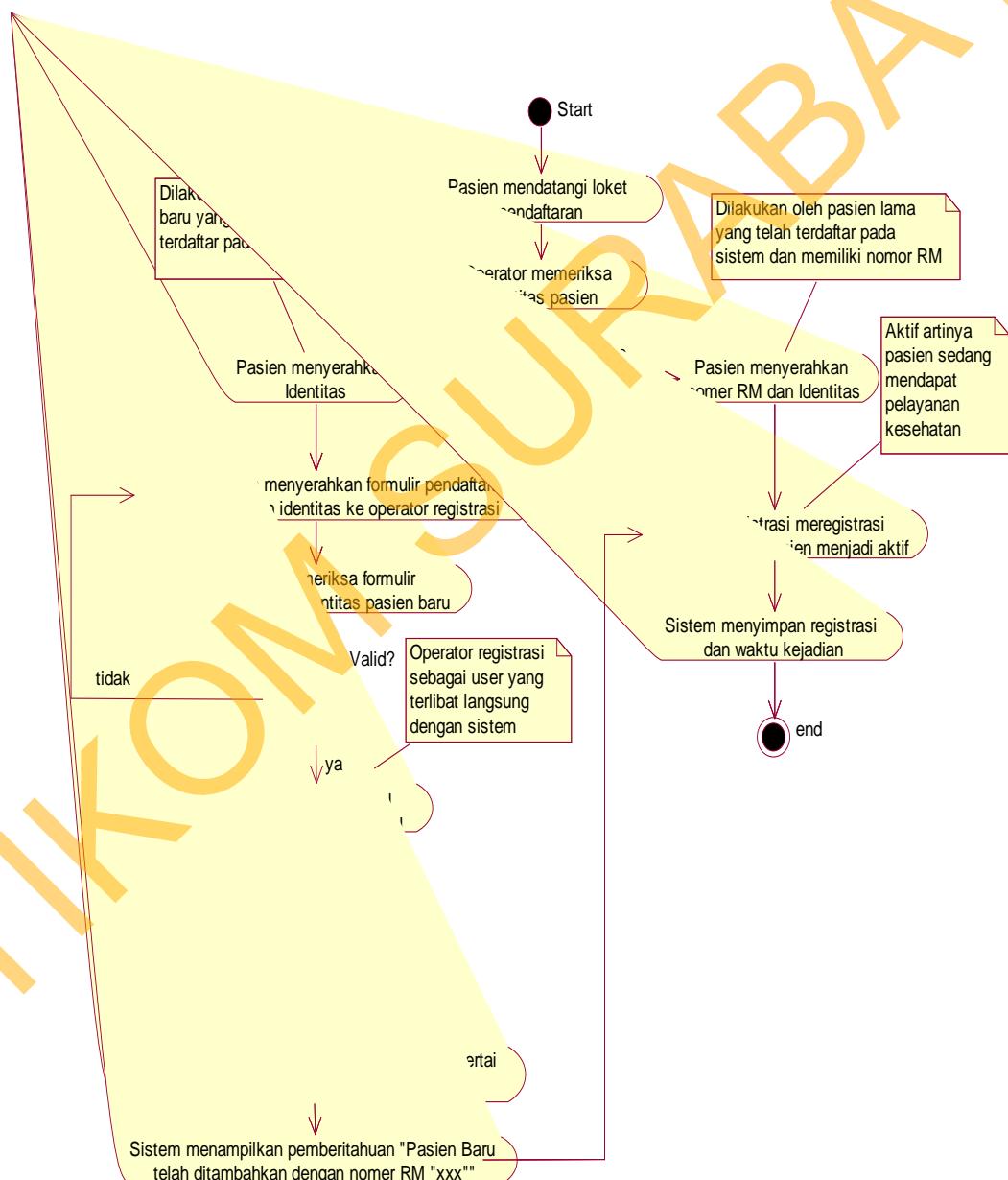


Gambar 3.28 *Flow Of Event* Mengubah Status Rumah Sakit

K. *Flow Of Event* Pendaftaran Pasien

Pendaftaran pasien baru dilakukan oleh operator registrasi. Use case ini menggambarkan seorang pasien datang ke sebuah rumah sakit yang telah bergabung ke integrasi sistem untuk mendapatkan pelayanan kesehatan. Aktivitas

yang terjadi dimulai dari pasien datang ke rumah sakit melakukan pendaftaran dengan menyerahkan identitas. Operator registrasi rumah sakit akan memeriksa pasien berdasarkan identitasnya apakah telah terdaftar pada sistem atau belum. Apabila belum terdaftar, pasien akan mendapatkan formulir pendaftaran pasien baru untuk diisi. Setelah formulir diisi, kemudian diserahkan kembali ke operator registrasi untuk diperiksa valid atau tidaknya. Formulir yang telah terisi diserahkan dengan melampirkan identitas pasien.

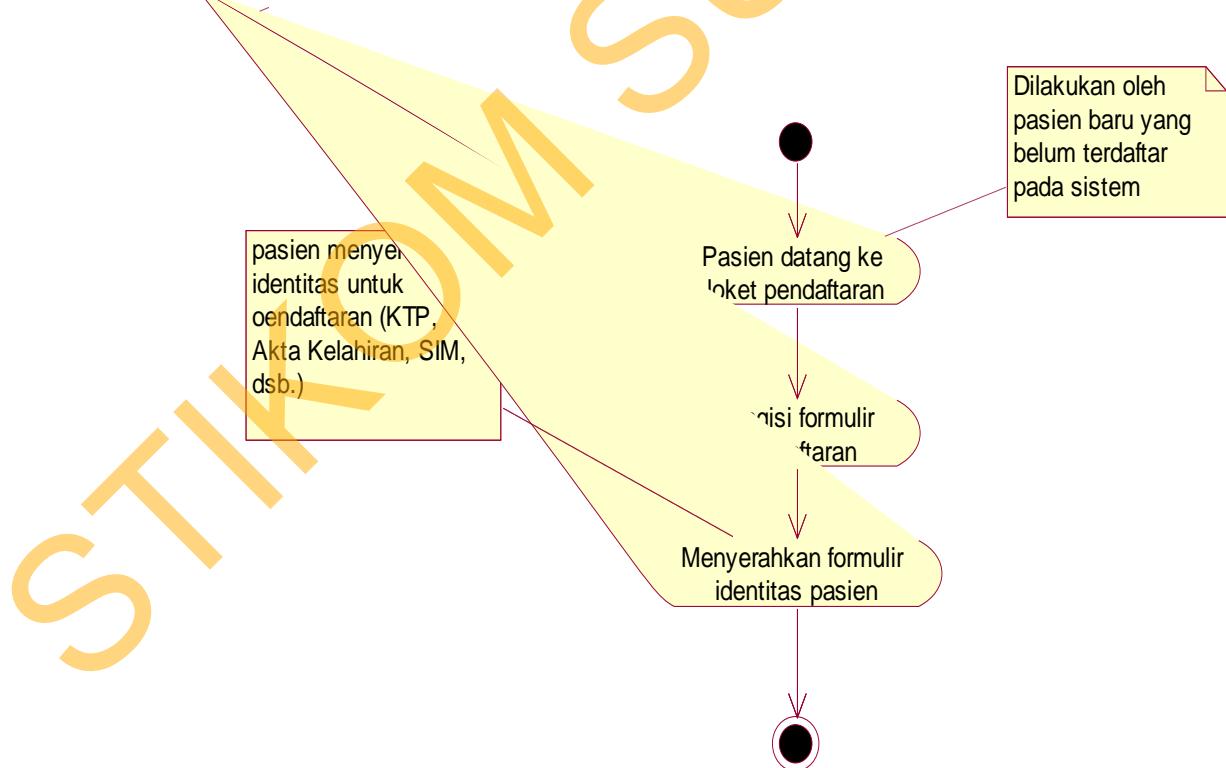


Gambar 3.29 *Flow Of Event* Pendaftaran Pasien

Berdasarkan formulir yang telah valid operator registrasi akan menambahkan pasien baru ke dalam sistem. Setelah data pendaftaran pasien baru disimpan, sistem akan memberikan nomer rekam medis kepada pasien. Pada gambar 3.29 akan digambarkan urutan proses yang terjadi selanjutnya. Setelah data pasien tersimpan dalam sistem, operator akan meregistrasikan pasien agar sistem mencatat bahwa pasien sedang mendapatkan pelayanan kesehatan di rumah sakit yang dituju pada waktu dan tanggal kejadian. Untuk pasien lama yang datang ke loket dengan menyerahkan nomor rekam medis, operator akan meregistrasikan pasien untuk mendapatkan pelayanan kesehatan sesuai dengan keluhan dan tujuan pasien.

L. *Flow Of Event* Menyerahkan Identitas

Pada use case ini terjadi apabila seorang pasien baru yang melakukan pendaftaran belum terdaftar dalam sistem. Jadi terlebih dahulu pasien akan mendaftarkan diri melalui loket pendaftaran dengan menyerahkan identitas yang dimiliki oleh pasien. Untuk lebih jelasnya akan digambarkan pada gambar 3.29.

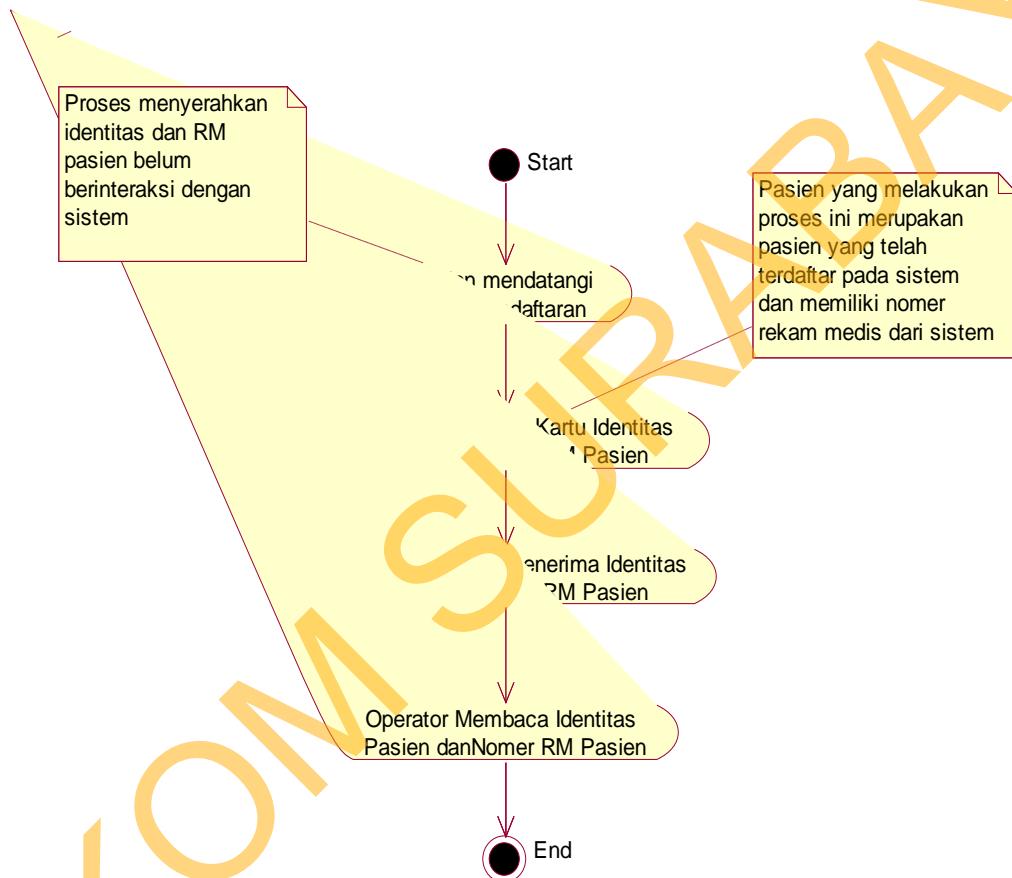


Gambar 3.30 *Flow Of Event* Menyerahkan Identitas

M. Flow Of Event Menyerahkan Nomer Rekam Medis (RM)

Use case ini berlangsung apabila pasien yang melakukan pendaftaran pada loket merupakan pasien yang telah terdaftar pada sistem. Dengan artian pasien sebelumnya pernah mendapat pelayanan kesehatan dan telah terdaftar di penyedia layanan kesehatan yang telah bergabung pada sistem terintegrasi.

Aktivitas yang terjadi pada *flow of event* ini akan digambarkan pada gambar 3.31. Aktivitas yang terjadi hanya penyerahan dokumen bukan transaksi.

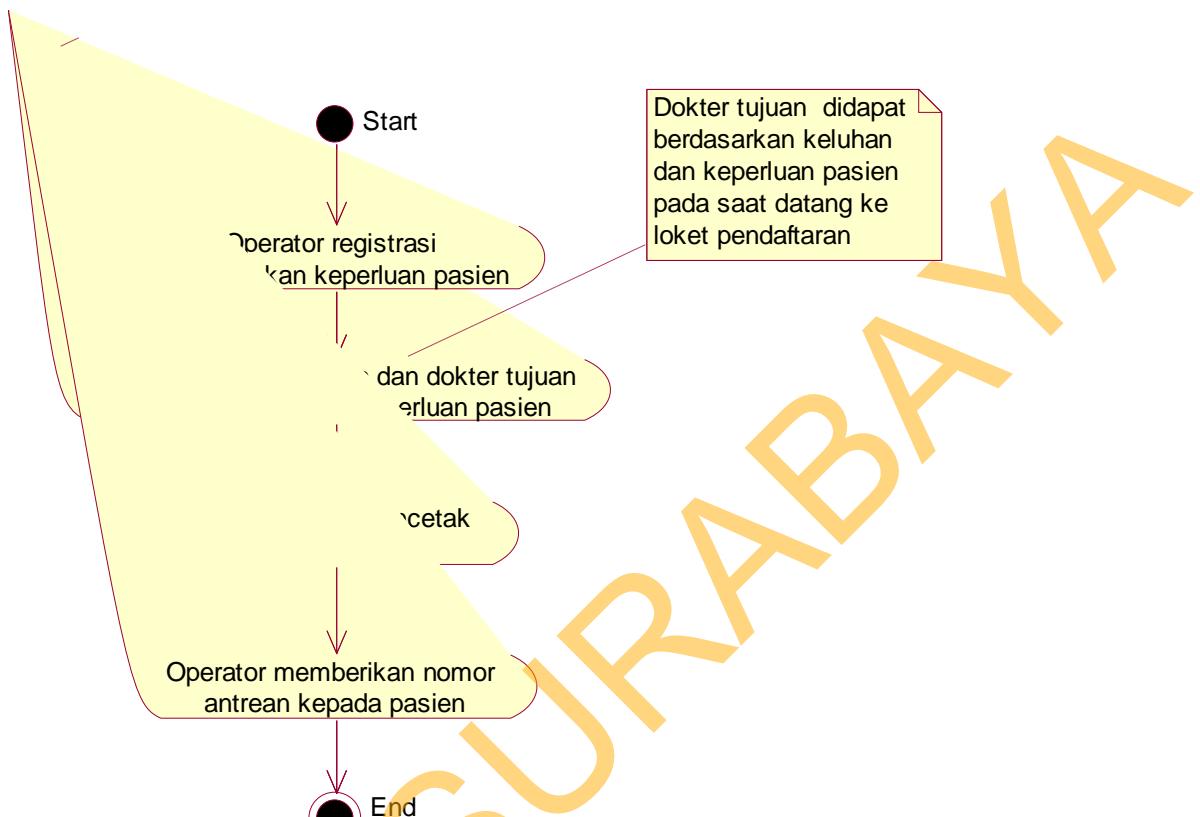


Gambar 3.31 *Flow Of Event* Menyerahkan Nomer Rekam Medis (RM)

N. Flow Of Event Cetak Nomer Antrean dan Dokter Tujuan

Operator registrasi setelah melakukan registrasi ulang pasien ke dalam sistem, akan menanyakan keluhan dan keperluan pasien sehingga memerlukan pelayanan kesehatan di rumah sakit yang bersangkutan. Berdasarkan keterangan pasien, operator akan mencetak nomer antrean dan dokter yang akan dituju selanjutnya. Nomor antrean dan dokter tujuan diserahkan kepada pasien yang

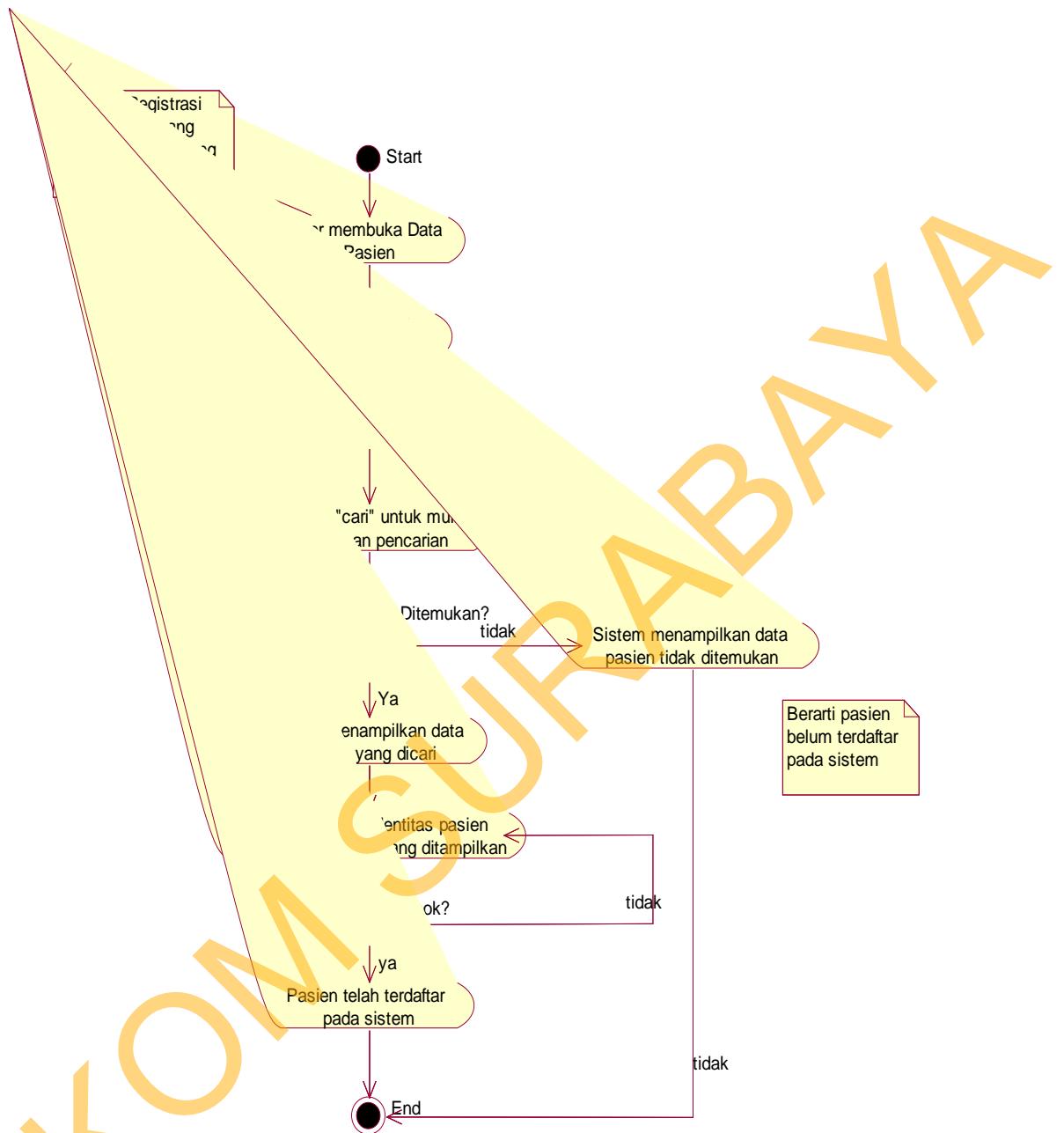
melakukan pendaftaran. Proses mencetak nomor antrean akan digambarkan pada gambar 3.32.



Gambar 3.32 *Flow Of Event* Mencetak Nomor Antrean dan Dokter yang Dituju

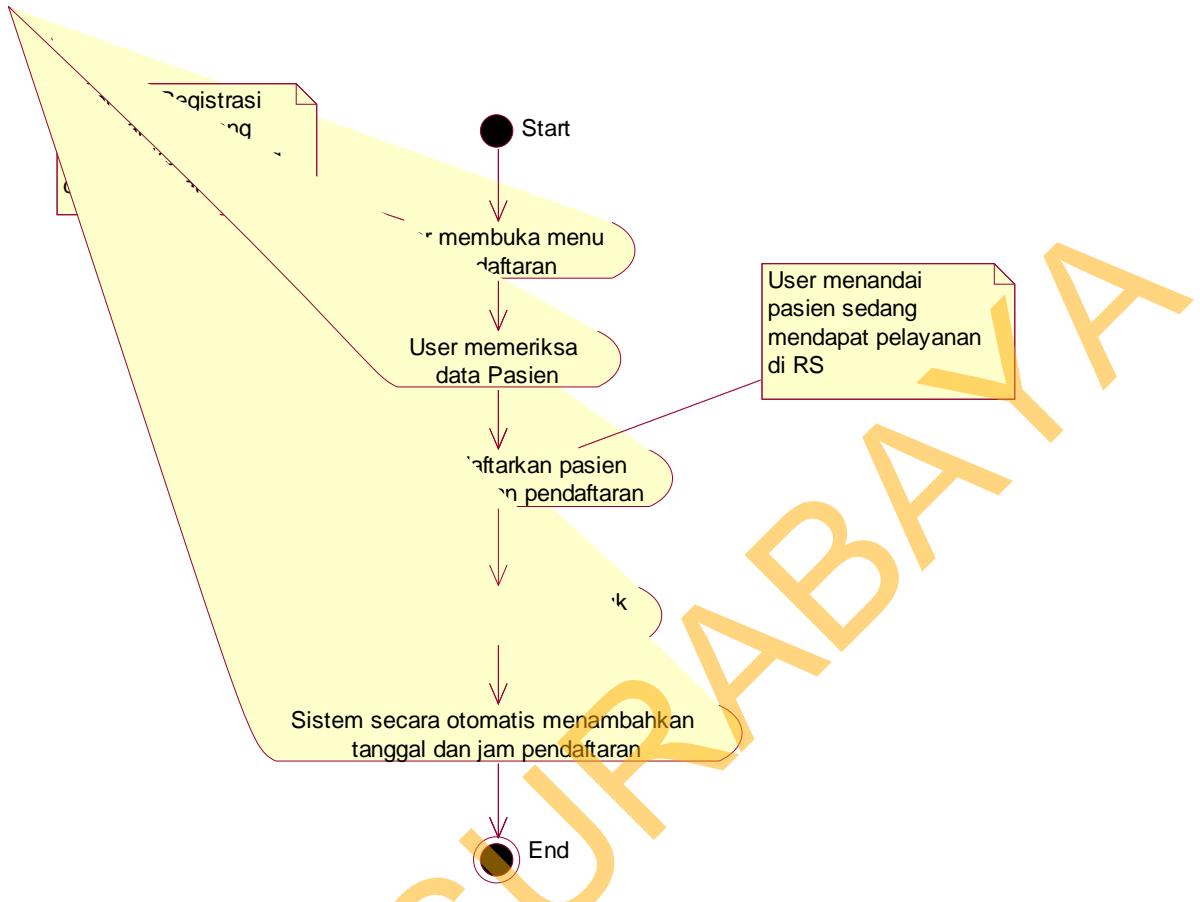
O. *Flow Of Event* Memeriksa Status Pasien

Pada use case ini akan digambarkan *flow of event* mengenai pemeriksaan status pasien yang datang ke loket pendaftaran. Operator registrasi membuka menu data pasien dan memasukkan identitas atau nomer rekam kesehatan untuk memeriksa status pasien pada menu pencarian data pasien. Apabila data ditemukan, sistem akan menampilkan data pasien yang dicari untuk kemudian dicocokkan dengan identitas yang diserahkan oleh pasien. Apabila identitas cocok, berarti pasien telah terdaftar pada sistem. Namun sebaliknya apabila data pasien tidak ditemukan pada saat pencarian data pasien, berarti pasien belum terdaftar pada sistem sehingga perlu didaftarkan sebagai pasien baru. Untuk lebih jelasnya akan digambarkan dengan diagram aktivitas 3.33 berikut.

Gambar 3.33 *Flow Of Event* Memeriksa Status Pasien

P. *Flow Of Event* Menambahkan Pendaftaran Pasien

Pasien yang telah terdaftar pada sistem, akan diregistrasi oleh operator registrasi agar pasien tercatat pada sistem sedang mendapatkan pelayanan kesehatan. Sehingga nanti data pasien dapat diakses oleh dokter yang akan menangani pasien.



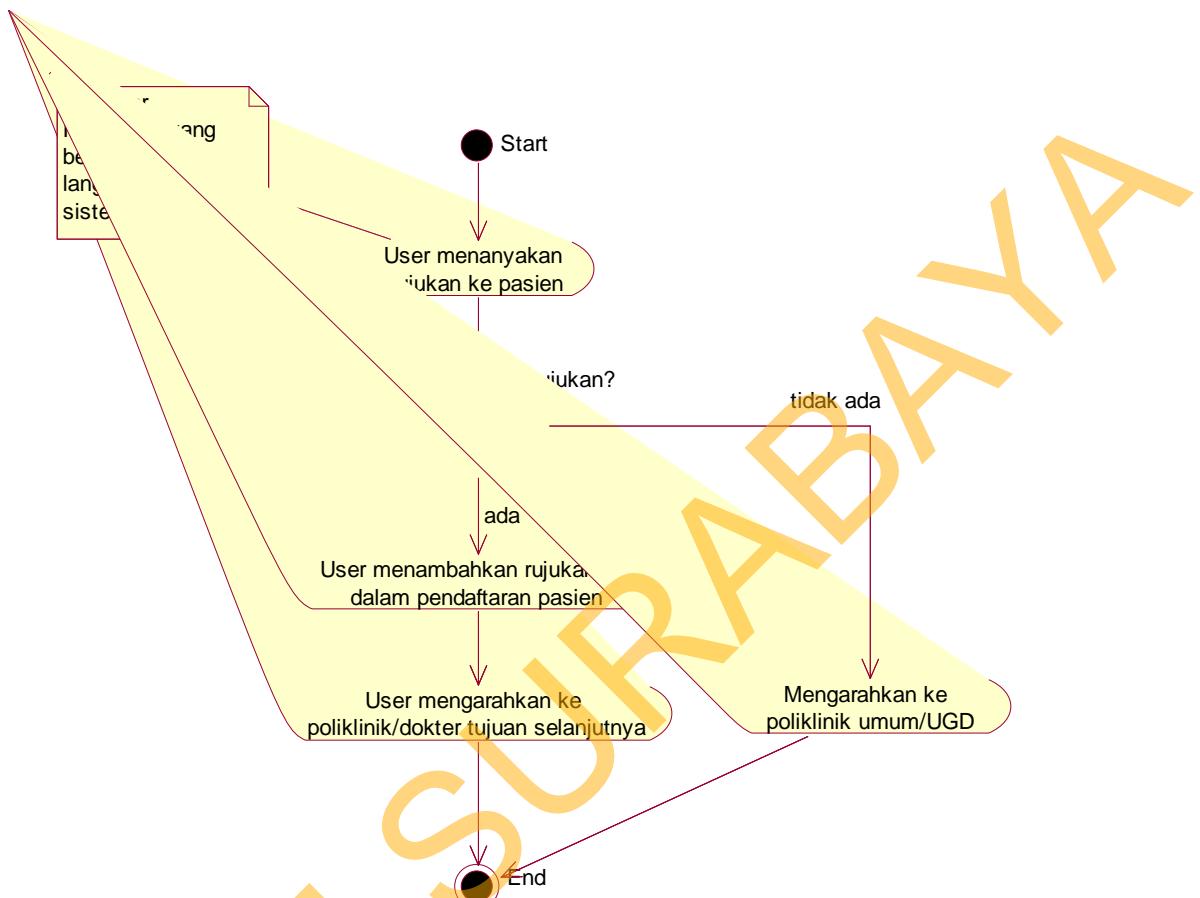
Gambar 3.34 *Flow Of Event* Menambahkan Pendaftaran Pasien

Pada gambar 3.34 aktivitas selanjutnya yang dilakukan oleh operator registrasi yaitu menyimpan registrasi pasien. Sistem akan menyimpan proses tersebut secara otomatis akan menambahkan waktu dan tanggal kejadian setelah user menekan tombol simpan.

Q. *Flow Of Event* Memeriksa Rujukan

Operator yang menangani pendaftaran pasien, selain menanyakan keluhan dan tujuan pasien datang ke rumah sakit, juga menanyakan apakah pasien datang membawa surat rujukan dari puskesmas atau penyedia layanan kesehatan lainnya. Apabila pasien membawa surat rujukan, setelah registrasi, pasien akan diarahkan ke poliklinik/dokter tujuan selanjutnya sesuai dengan rujukan yang dibawa pasien. Namun apabila pasien datang tanpa membawa surat rujukan, operator registrasi akan menambahkan adanya rujukan ke dalam pendaftaran dan mengarahkan pasien ke poliklinik umum/UGD sesuai dengan keluhan pasien. Berdasarkan

keluhannya pasien akan diarahkan ke penanganan selanjutnya. Untuk memperjelas dapat kita lihat pada gambar 3.35.

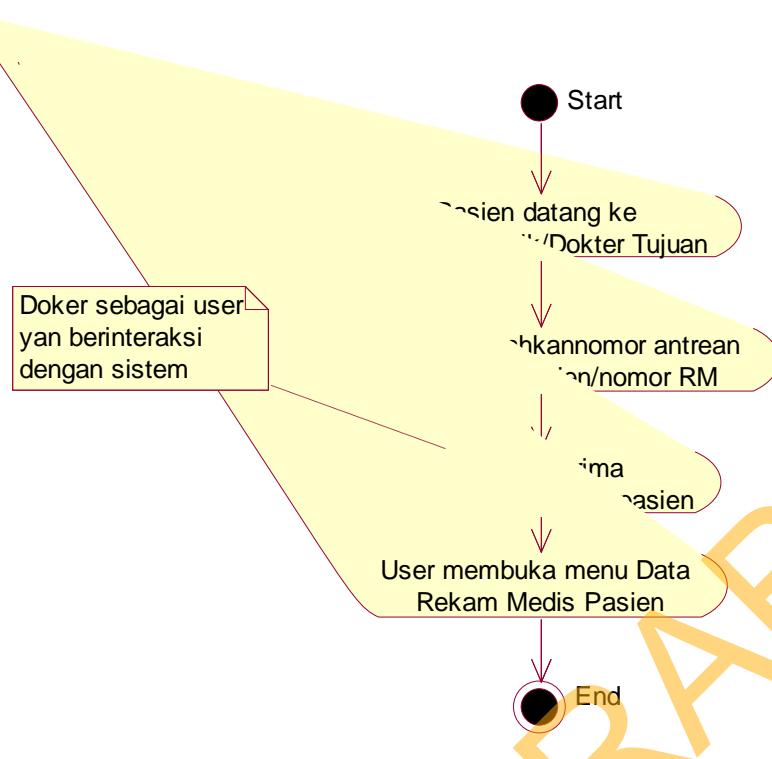


Gambar 3.35 *Flow Of Event* Memeriksa Rujukan

R. *Flow Of Event* Menyerahkan Nomor Antrean dan Identitas

Pada *flow of event* ini akan menggambarkan aktivitas yang terjadi pada use case menyerahkan nomor antrean dan identitas. Pasien yang telah melakukan registrasi dan memeliki nomer antrean datang ke poliklinik atau dokter tujuannya. Pasien menyerahkan nomer antrean dan identitasnya kepada dokter. Dokter kemudian membuka data rekam kesehatan berdasarkan identitas atau nomor rekam medis yang diserahkan.

Pada gambar 3.36 digambarkan urutan proses yang terjadi. Use case dan *flow of event* ini hanya menggambarkan aliran dokumen dari pasien ke dokter dimana dokumen berupa nomor antrean dan identitas/nomor ekam medis pasien, bukan merupakan sebuah transaksi.



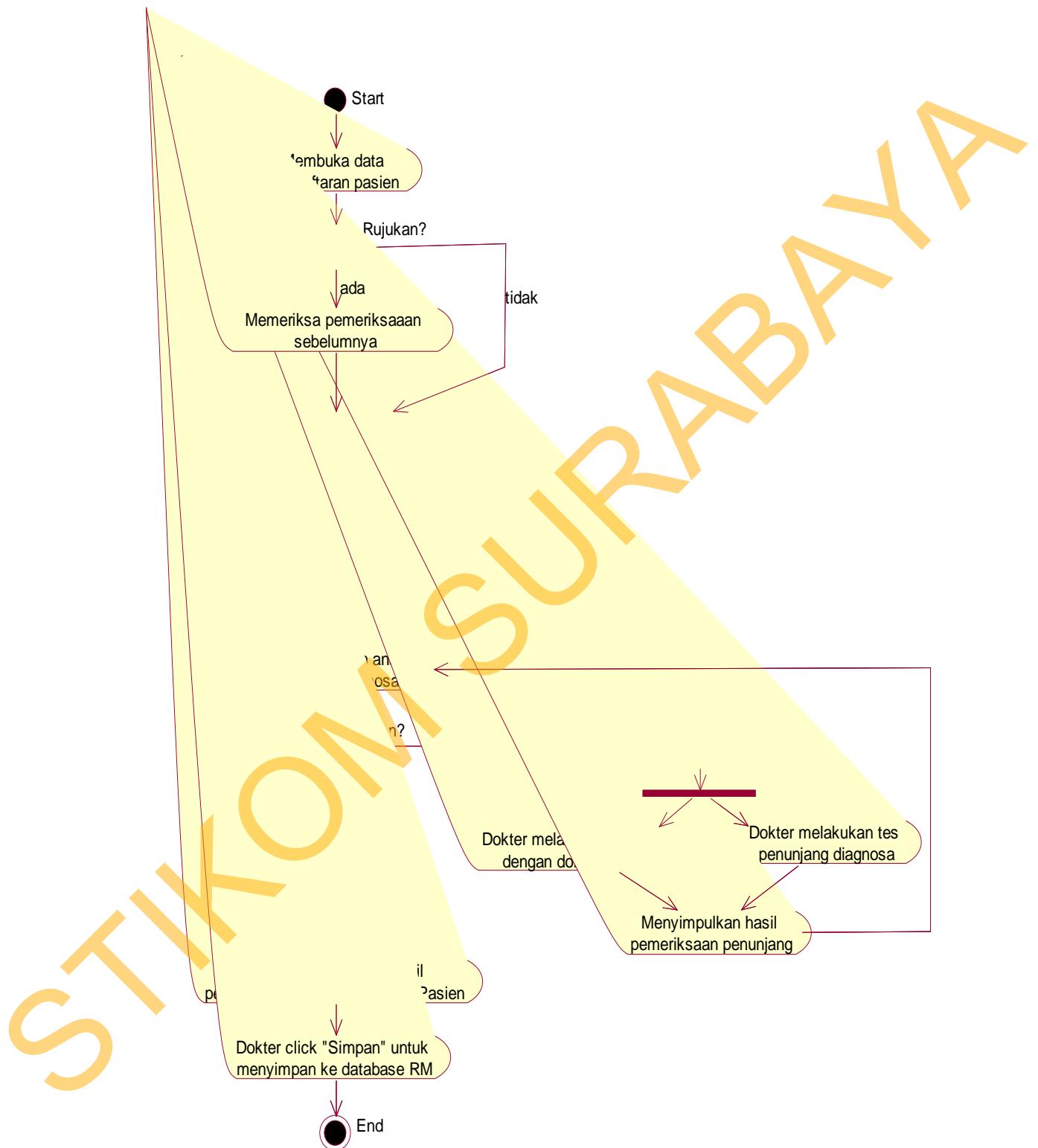
Gambar 3.36 *Flow Of Event* Menyerahkan Nomor Antrean dan Identitas

S. *Flow Of Event* Melakukan Pemeriksaan

Dokter sebagai aktor yang terlibat langsung dengan sistem sebelum melakukan pemeriksaan akan memeriksa pendaftaran pasien. Apabila ada rujukan, dokter akan memeriksa hasil pemeriksaan di penyedia layanan kesehatan yang merujuk pasien. Berdasarkan pemeriksaan sebelumnya dokter akan melanjutkan pemeriksaan pasien. Dokter akan melakukan anamnesa yaitu wawancara kepada pasien mengenai keluhan-keluhan yang dirasakan. Setelah proses anamnesa dirasa cukup, dokter melanjutkan ke pemeriksaan fisik pasien meliputi pemeriksaan suhu badan, tekanan darah, kecepatan pernafasan (*Respiration Rate*), dan kecepatan denyut nadi/jantung. Berdasarkan anamnesa dan pemeriksaan fisik dokter akan melakukan proses diagnosa terhadap penyakit yang diderita pasien. Apabila dokter masih belum yakin dengan hasil pemeriksaan dan diagnosa, dokter akan berkonsultasi dengan dokter spesialis ataupun melakukan tes penunjang untuk memastikan hasil diagnosa yaitu tes laboratorium.

Dokter akan kembali melakukan diagnosa untuk menyimpulkan penyakit yang diderita oleh pasien. Hasil pemeriksaan akan dicatat ke dalam data rekam

kesehatan kemudian disimpan ke dalam database. Untuk lebih jelasnya dapat dilihat pada gambar 3.37.



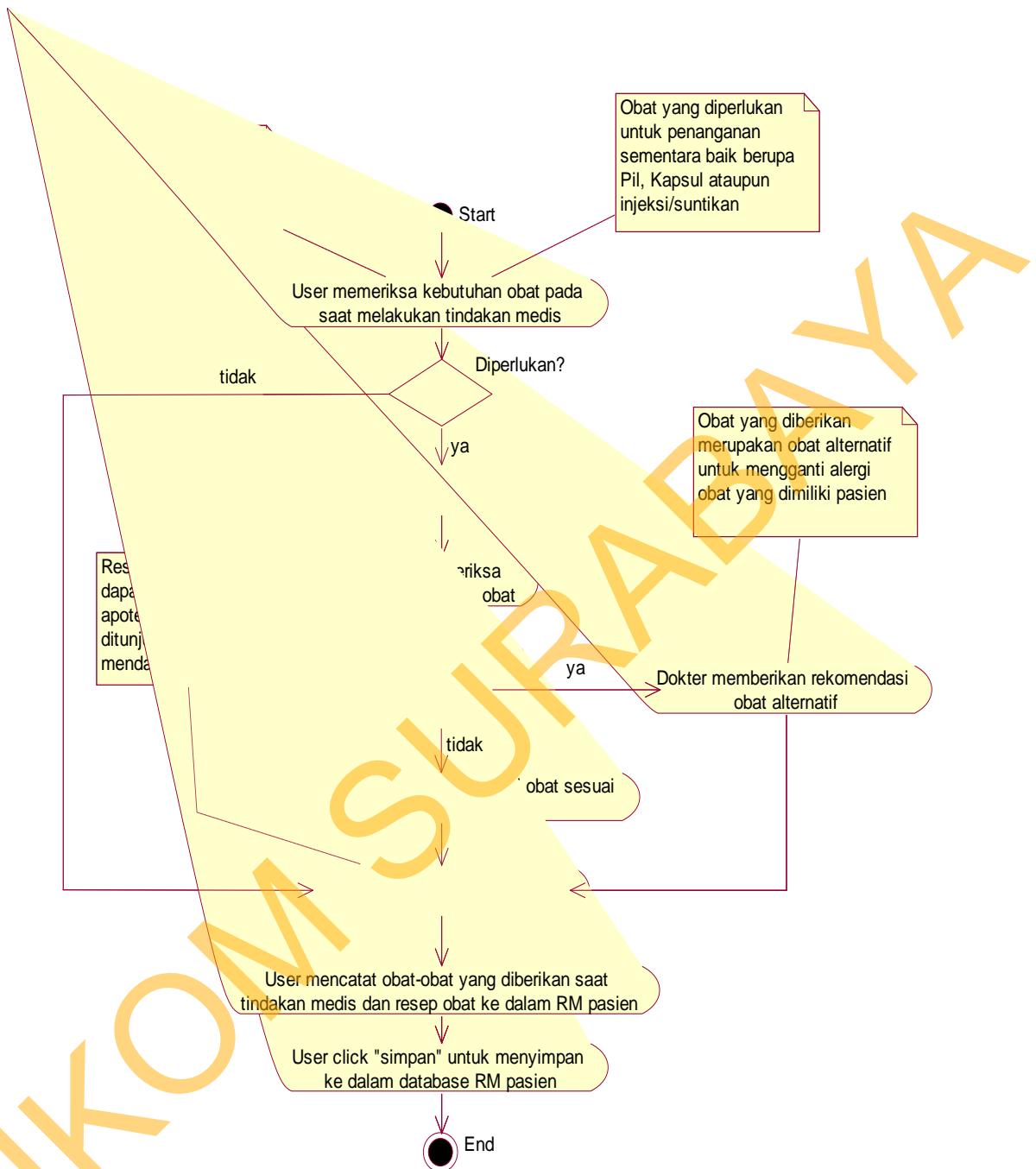
Gambar 3.37 *Flow Of Event Melakukan Pemeriksaan*

T. *Flow Of Event* Melakukan Tindakan Medis

Pada *flow of event* ini akan digambarkan aktivitas-aktivitas yang terjadi pada use case melakukan tindakan medis. Pada saat pemeriksaan pasien, setelah pemeriksaan fisik dokter akan melakukan tindakan medis apabila diperlukan. Tindakan medis tersebut antara lain memberikan suntikan (*injection*), ataupun pil untuk menangani sementara saat pemeriksaan. Sebelum memberikan tindakan medis, dokter akan membuka rekam kesehatan yang bersangkutan untuk memeriksa apakah pasien memiliki alergi terhadap obat tertentu. Apabila dari hasil pemeriksaan pasien memiliki riwayat alergi terhadap obat, maka pasien diberikan obat alternatif lain namun berfungsi sama dengan obat utama namun dengan campuran yang sedikit berbeda.

Setelah pemeriksaan dan pemberian tindakan medis sementara dalam penanganannya, dokter akan mencatat dan memberikan resep obat untuk pengobatan selanjutnya di rumah atau rawat inap. Resep obat tersebut dapat ditebus di apotek yang telah ditentukan oleh rumah sakit. Daftar resep obat yang diberikan kepada pasien, disimpan oleh dokter ke dalam data rekam kesehatan yang bersangkutan dan sistem akan menyimpan ke dalam database.

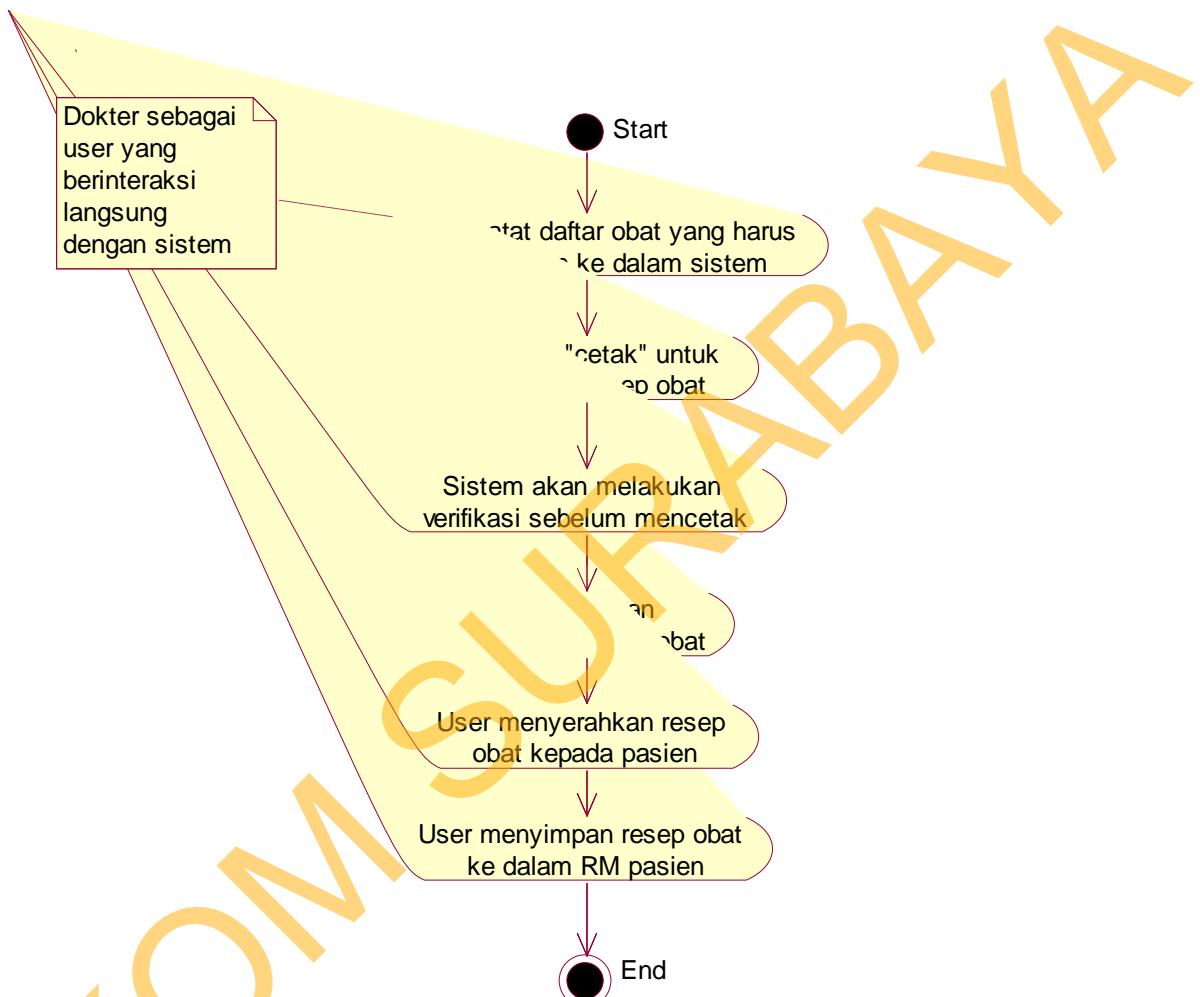
Pada gambar 3.38 akan digambarkan urutan aktivitas yang terjadi pada saat dokter melakukan tindakan medis ke pasien. Untuk lebih jelasnya dapat dilihat pada gambar 3.38 dibawah ini.

Gambar 3.38 *Flow Of Event* Melakukan Tindakan Medis

U. *Flow Of Event* Mencetak Resep Obat

Dokter merupakan aktor yang berperan dalam pemeriksaan, diagnosa, tindakan medis termasuk memberikan resep obat. Setelah pemeriksaan pasien, dokter akan mencatat resep obat yang akan diberikan kepada pasien. Resep obat

akan dicetak melalui mesin pencetak dan memberikannya kepada pasien. Dokter akan menambahkan resep obat yang diberikan kepada pasien ke dalam data rekam kesehatan yang bersangkutan. Untuk urutan aktivitasnya dapat dilihat pada gambar 3.39.



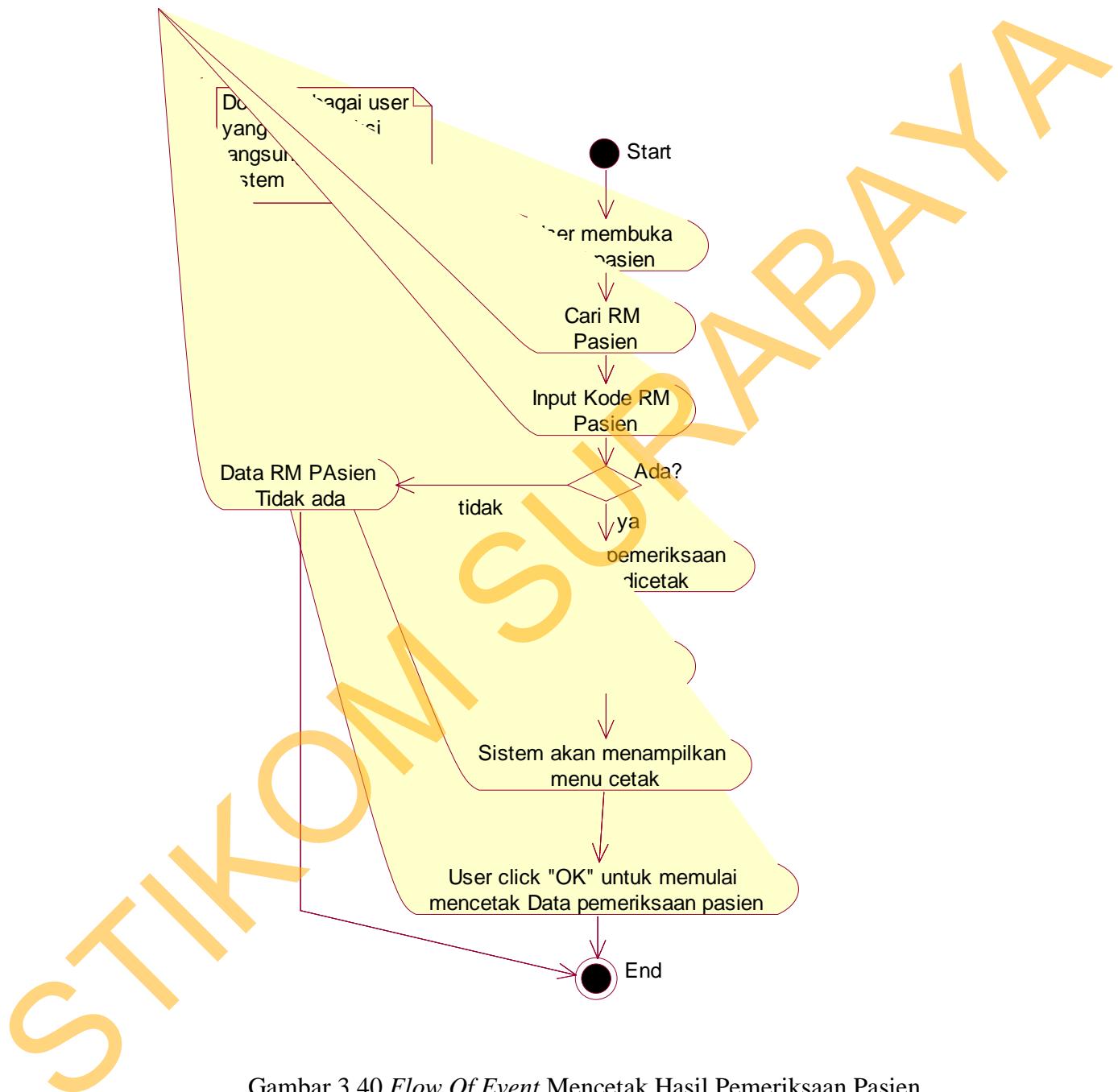
Gambar 3.39 *Flow Of Event* Mencetak Resep Obat

V. *Flow Of Event* Mencetak Hasil Pemeriksaan Pasien

User yang terlibat langsung dengan sistem disini adalah dokter. Dokter akan membuka data rekam kesehatan. Dokter akan memasukkan nomor rekam kesehatan pada menu pencarian. Apabila ditemukan sistem akan menampilkan data rekam kesehatan yang bersangkutan. Dokter akan memilih hasil pemeriksaan sesuai dengan yang dibutuhkan, karena data rekam kesehatan terdiri dari banyak hasil pemeriksaan. Data pemeriksaan tersebut dicetak apabila diperlukan oleh

dokter untuk pembelajaran atau pasien untuk arsip. Setelah dokter memilih tombol cetak, mesin pencetak akan mencetak hasil pemeriksaan yang dipilih.

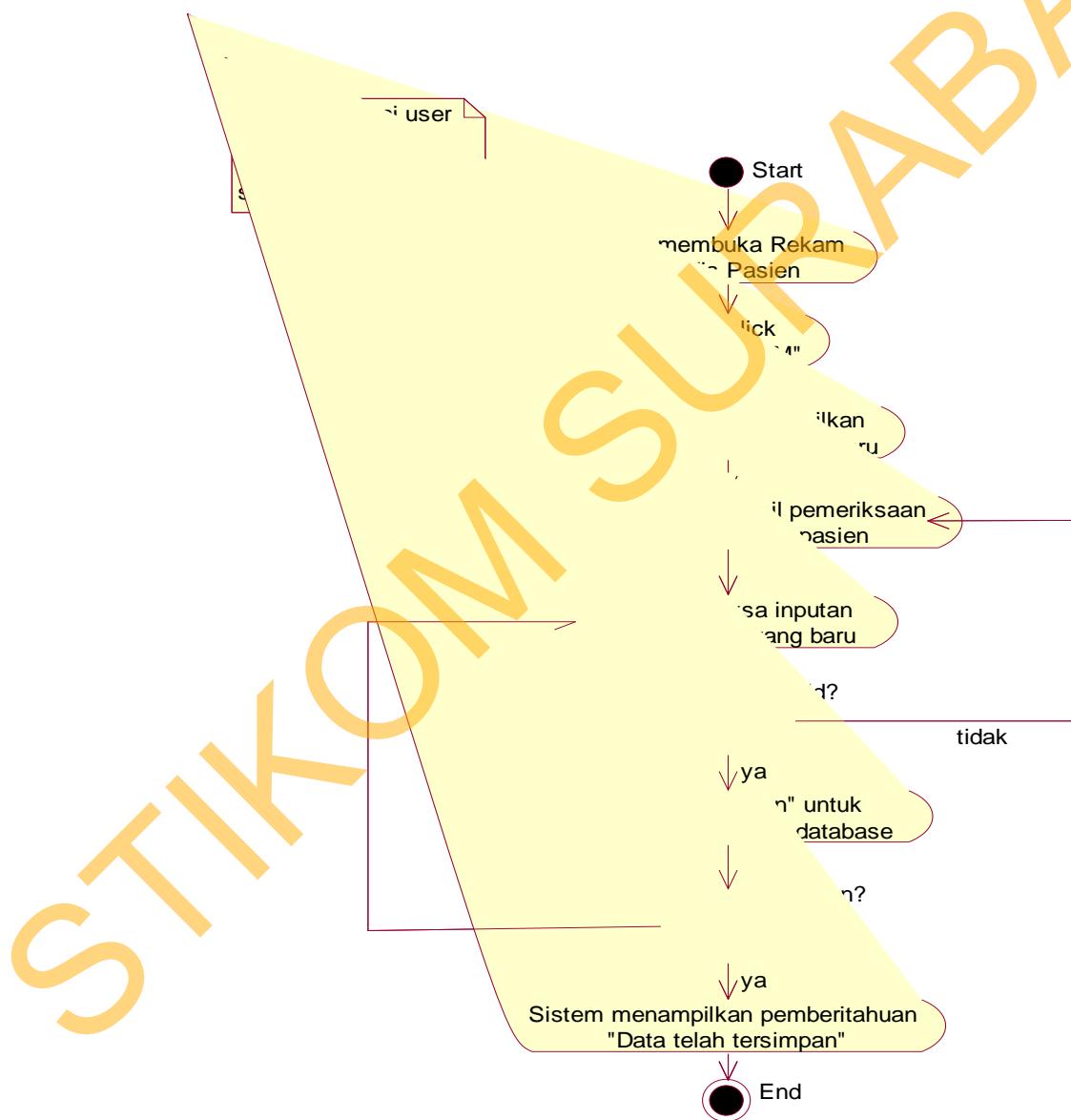
Pada gambar 3.40 akan digambarkan urutan proses percetakan hasil pemeriksaan.



Gambar 3.40 *Flow Of Event* Mencetak Hasil Pemeriksaan Pasien

W. *Flow Of Event* Menambahkan Data Rekam kesehatan

Pada *flow of event* ini akan digambarkan aktivitas yang terjadi dalam proses menambahkan data rekam kesehatan. Aktor yang berinteraksi langsung dengan sistem adalah dokter. Dokter membuka data rekam kesehatan yang diperiksa berdasarkan kode rekam medis yang diserahkan oleh pasien. Dokter kemudian memilih menu tambah data rekam kesehatan untuk menambahkan hasil pemeriksaan dan tindakan medis pada saat pasien telah diperiksa. Sebelum menyimpan, dokter akan memeriksa apakah semua kegiatan atau pemeriksaan telah dicatat ke dalam sistem.



Gambar 3.41 *Flow Of Event* Menambahkan Data Rekam kesehatan

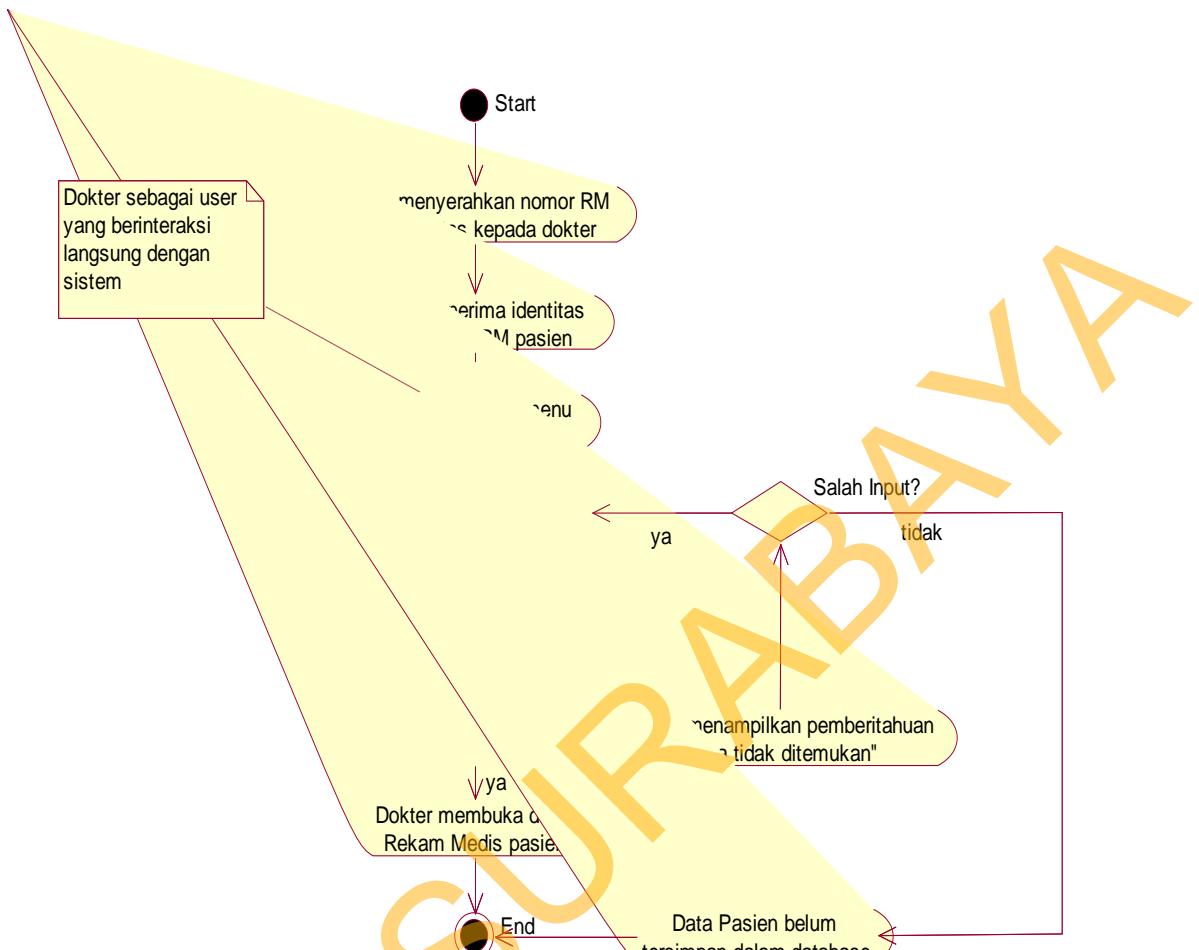
Pada gambar 3.41 akan menggambarkan *flow of event* menambahkan data rekam kesehatan. Selanjutnya data yang sudah valid akan disimpan kedalam database oleh dokter. Setelah tombol simpan dipilih oleh dokter, data hasil pemeriksaan pasien telah ditambahkan ke dalam data rekam kesehatan. Sistem akan menampilkan pemberitahuan bahwa data telah disimpan ke dalam database.

X. Flow Of Event Display Rekam kesehatan

Pada *flow of event* ini akan menampilkan aktivitas yang terjadi pada se case display data rekam kesehatan. Dokter sebagai user atau aktor yang berinteraksi langsung dengan sistem, jadi hanya dokter yang dapat mengakses data rekam medis seorang pasien. Untuk menampilkan data rekam kesehatan, harus berdasarkan ijin dan persetujuan dari pasien yang bersangkutan.

Pada proses menampilkan data rekam kesehatan, dimulai dengan pasien menyerahkan kode rekam medisnya kepada dokter. Dokter akan memasukkan kode tersebut ke dalam menu pencarian pasien. Apabila data tidak ditemukan sistem akan menampilkan pemberitahuan bahwa data rekam kesehatan tidak ditemukan, sehingga dokter akan memeriksa apakah pada saat memasukkan kode rekam kesehatan terjadi kesalahan atau data rekam kesehatan yang dicari memang belum tercatat pada sistem.

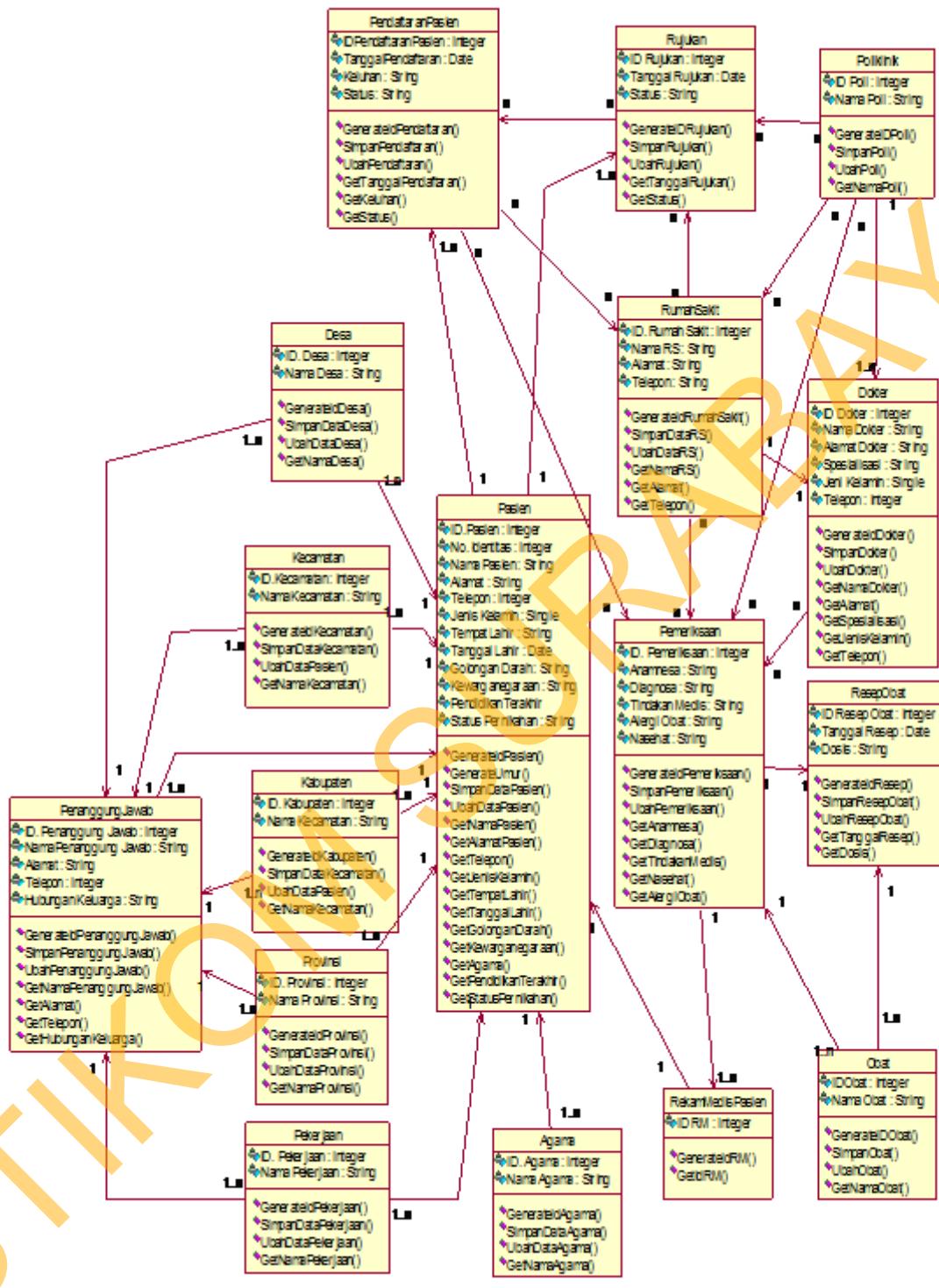
Namun apabila sebaliknya, data rekam kesehatan yang dimaksud ditemukan, sistem akan menampilkan data rekam medis tersebut ke layar. Untuk lebih jelasnya dapat kita lihat pada gambar 3.42.



Gambar 3.42 Flow Of Event Display Data Rekam kesehatan

3.4.5 Class Diagram Untuk Entity Bisnis

Class biasanya mendefinisikan objek-objek bisnis, class seperti ini biasanya mendefinisikan model database dari suatu aplikasi. Atas dasar itulah class seperti ini sering disebut dengan *class entity* karena mewakili objek database. Pada gambar 3.43 menunjukkan class entity yang ada pada sistem informasi rekam kesehatan elektronik terpusat, pada gambar 3.42 juga menampilkan multiplicity yang mengindikasikan adanya sejumlah objek pada sebuah asosiasi class.



Gambar 3.43 Class Diagram Untuk Entity Bisnis

Class yang ada memiliki atribut dan operasinya masing-masing, yang akan digunakan untuk menjelaskan bagaimana setiap class berperan dalam sistem, class-class yang ada dijelaskan sebagai berikut :

1. Class Pasien

Class Pasien memiliki atribut IDPasien, No. Identitas, NamaPasien, Alamat, Telepon, JenisKelamin, TempatLahir, TanggalLahir, GolonganDarah, Kewarganegaraan, Agama, PendidikanTerakhir, dan StatusPernikahan. Sedangkan metode atau operasi dari class ini antara lain GenerateIdPasien(), GenerateUmur(), SimpanDataPasien(), UbahDataPasien(), GetNamaPasien(), GetAlamatPasien(), GetTelepon(), GetJenisKelamin(), GetTempatLahir(), GetTanggalLahir(), GetGolonganDarah, GetKewarganegaraan(), GetAgama(), GetPendidikanTerakhir(), dan GetStatusPernikahan(). Class Pasien ini memiliki asosiasi dengan class Desa, Kecamatan, Kabupaten, Provinsi, Agama, Pekerjaan, PendaftaranPasien, Rujukan, dan PenanggungJawab yang memiliki multiplicity one-to-many dimana many pada sisi Desa, Kecamatan, Kabupaten, Provinsi, Pekerjaan, Agama, PendaftaranPasien, Rujukan, dan PenanggungJawab dan one pada sisi Pasien, artinya satu atau banyak Desa, Kecamatan, Kabupaten, Provinsi, Pekerjaan, Agama, PendaftaranPasien, Rujukan, dan PenanggungJawab memiliki satu Pasien. Sedangkan terdapat juga asosiasi one-to-one antara class Pasien dengan RekamMedisPasien yang artinya satu pasien memiliki satu rekam medis.

2. Class Penanggung Jawab

Class PenanggungJawab memiliki atribut IDPenanggungJawab, NamaPenanggungJawab, Alamat, Telepon, dan HubunganKeluarga. Sedangkan metode atau operasi dari class ini antara lain GenerateIdPenanggungJawab(),

SimpanPenanggungJawab(), UbahPenanggungJawab(), GetNamaPenanggungJawab(), GetAlamat(), GetTelepon(), dan GetHubunganKeluarga. Class PenanggungJawab ini memiliki asosiasi dengan class Desa, Kecamatan, Kabupaten, Provinsi, dan Pekerjaan yang memiliki multiplicity one-to-many dimana many pada sisi Desa, Kecamatan, Kabupaten, Provinsi, dan Pekerjaan, dan one pada sisi PenanggungJawab, artinya satu atau banyak Desa, Kecamatan, Kabupaten, Provinsi, dan Pekerjaan memiliki satu PenanggungJawab. Sedangkan terdapat juga asosiasi many-to-one antara class PenanggungJawab dengan Pasien yang artinya banyak PenanggungJawab memiliki satu Pasien.

3. Class Desa

Class Desa memiliki atribut IDDesa dan NamaDesa. Sedangkan metode atau operasi dari class ini antara lain GenerateIdDesa(), SimpanDesa(), UbahDesa() dan GetNamaDesa(). Class ini memiliki asosiasi dengan class Pasien dan class PenanggungJawab dengan multiplicity many to one, satu pada sisi Desa dan many pada sisi Pasien dan PenanggungJawab yang mana memiliki arti satu desa memiliki banyak pasien dan banyak penanggung jawab.

4. Class Kecamatan

Class Kecamatan memiliki atribut IDKecamatan dan NamaKecamatan. Sedangkan metode atau operasi dari class ini antara lain GenerateIdKecamatan(), SimpanKecamatan(), UbahKecamatan() dan GetNamaKecamatan(). Class ini memiliki asosiasi dengan class Pasien dan class PenanggungJawab dengan multiplicity many to one, satu pada sisi Kecamatan dan banyak pada sisi Pasien dan PenanggungJawab yang mana memiliki arti satu kecamatan memiliki banyak pasien dan banyak penanggung jawab.

5. Class Kabupaten

Class Kabupaten memiliki atribut IDKabupaten dan NamaKabupaten. Sedangkan metode atau operasi dari class ini antara lain GenerateIdKabupaten(), SimpanKabupaten(), UbahKabupaten() dan GetNamaKabupaten(). Class ini memiliki asosiasi dengan class Pasien dan class PenanggungJawab dengan multiplicity many to one, satu pada sisi Kabupaten dan many pada sisi Pasien dan PenanggungJawab yang mana memiliki arti satu kabupaten memiliki banyak pasien dan banyak penanggung jawab.

6. Class Provinsi

Class Provinsi memiliki atribut IDProvinsi dan NamaProvinsi. Sedangkan metode atau operasi dari class ini antara lain GenerateIdProvinsi(), SimpanProvinsi(), UbahProvinsi() dan GetNamaProvinsi(). Class ini memiliki asosiasi dengan class Pasien dan class PenanggungJawab dengan multiplicity many to one, satu pada sisi Provinsi dan banyak pada sisi Pasien dan PenanggungJawab yang mana memiliki arti satu provinsi memiliki banyak pasien dan banyak penanggung jawab.

7. Class Pekerjaan

Class Pekerjaan memiliki atribut IDPekerjaan dan NamaPekerjaan. Sedangkan metode atau operasi dari class ini antara lain GenerateIdPekerjaan(), SimpanPekerjaan(), UbahPekerjaan() dan GetNamaPekerjaan(). Class ini memiliki asosiasi dengan class Pasien dan class PenanggungJawab dengan multiplicity many to one, satu pada sisi pekerjaan dan banyak pada sisi Pasien dan PenanggungJawab yang mana memiliki arti satu pekerjaan memiliki banyak pasien dan banyak penanggung jawab.

8. Class Agama

Class Agama memiliki atribut IDAgama dan NamaAgama. Sedangkan metode atau operasi dari class ini antara lain GenerateIdAgama(), SimpanAgama(), UbahAgama() dan GetNamaAgama(). Class ini memiliki asosiasi dengan class Pasien dengan multiplicity many to one, satu pada sisi agama dan banyak pada sisi Pasien yang mana memiliki arti satu agama memiliki banyak.

9. Class Pendaftaran Pasien

Class Pekerjaan memiliki atribut IDPendaftaranPasien, TanggalPendaftaran, Keluhan dan Status. Sedangkan metode atau operasi dari class ini antara lain GenerateIdPendaftaran(), SimpanPendaftaran(), UbahPendaftaran(), GetTanggalPendaftaran(), GetKeluhan() dan GetStatus(). Class ini memiliki asosiasi dengan class Rujukan, RumahSakit dan class Pemeriksaan dengan multiplicity many to many, yang mana memiliki arti banyak pendaftaran memiliki banyak rujukan, dan rumah sakit. Selain itu class PendaftaranPasien juga memiliki asosiasi dengan class pasien dengan multiplicity many to one yang artinya banyak pendaftaran yang dimiliki oleh satu pasien.

10. Class Rujukan

Class Pekerjaan memiliki atribut IDRujukan, TanggalRujukan, dan Status. Sedangkan metode atau operasi dari class ini antara lain GenerateIdRujukan(), SimpanRujukan(), UbahRujukan(), GetTanggalRujukan(), dan GetStatus(). Class ini memiliki asosiasi dengan class Pasien, Rumah Sakit dan class Poliklinik dengan multiplicity many to many, yang mana memiliki arti banyak rujukan memiliki banyak rujukan, poliklinik dan rumah sakit. Selain itu class Rujukan

jug juga memiliki asosiasi dengan class Pasien dengan multiplicity many to one yang artinya banyak Rujukan yang dimiliki oleh satu pasien.

11. Class Rumah Sakit

Class Pekerjaan memiliki atribut IDRumahSakit, NamaRS, Alamat dan Telepon. Sedangkan metode atau operasi dari class ini antara lain GenerateIdRumahSakit(), SimpanDataRS(), UbahDataRS(), GetNamaRS(), GetAlamat() dan GetTelepon(). Class ini memiliki asosiasi dengan class Rujukan, PendaftaranPasien, poliklinik dan class Pemeriksaan dengan multiplicity many to many, yang mana memiliki arti banyak Rumah Sakit memiliki banyak rujukan, poliklinik, pemeriksaan dan pendaftaran pasien. Selain itu class Rumah Sakit juga memiliki asosiasi dengan class dokter dengan multiplicity one to one yang artinya satu rumah sakit yang memiliki oleh satu dokter. Dengan kata lain satu dokter hanya bekerja pada satu rumah sakit.

12. Class Poliklinik

Class Poliklinik memiliki atribut IDPoli dan NamaPoli. Sedangkan metode atau operasi dari class ini antara lain GenerateIdPoli (), SimpanPoli(), UbahPoli(), dan GetNamaPoli(). Class ini memiliki asosiasi dengan class Rujukan, RumahSakit dan class Pemeriksaan dengan multiplicity many to many, yang mana memiliki arti banyak poliklinik memiliki banyak rujukan, pemeriksaan dan rumah sakit. Selain itu class poliklinik juga memiliki asosiasi dengan class dokter dengan multiplicity one to many yang artinya satu poliklinik memiliki banyak dokter

13. Class Pemeriksaan

Class pemeriksaan memiliki atribut IDPemeriksaan, Anamnesa, Diagnosa, TindakanMedis, AlergiObat dan Nasehat. Sedangkan metode atau operasi dari class ini antara lain GenerateIdPemeriksaan(), SimpanPemeriksaan(), UbahPemeriksaan(), GetAnamnesa(), GetDiagnosa(), GetTindakanMedis(), GetNasehat dan GetAlergiObat. Class ini memiliki asosiasi dengan class RumahSakit, Poliklinik, Dokter dan class PendaftaraPasien dengan multiplicity many to many, yang mana memiliki arti banyak pemeriksaan memiliki banyak pendaftaranPasien, poliklinik, dokter dan rumah sakit. Selain itu class pemeriksaan juga memiliki asosiasi dengan class resep obat dengan multiplicity one to one yang artinya satu pemeriksaan memiliki satu resep obat. Class pemeriksaan juga memiliki asosiasi dengan class obat dengan multiplicity one to many dengan alergi obat yang artinya satu pemeriksaan memiliki banyak alergi obat.

14. Class Dokter

Class Dokter memiliki atribut IDDokter, NamaDokter, AlamatDokter, Spesialisai, JenisKelamin dan Telepon. Sedangkan metode atau operasi dari class ini antara lain GenerateIdDokter(), SimpanDokter(), UbahDokter(), GetNamaDokter(), GetAlamat(), GetSpesialisasi(), GetJenisKelamin(), dan GetTelepon(). Class ini memiliki asosiasi dengan class RumahSakit dengan multiplicity one to one, yang mana memiliki arti satu rumah sakit memiliki satu dokter. Selain itu class dokter juga memiliki asosiasi dengan class pemeriksaan dengan multiplicity many to many yang artinya banyak dokter yang menangani oleh banyak pemeriksaan. Class dokter juga memiliki hubungan asosiasi dengan

class poliklinik dengan multiplicity many to one yang artinya satu atau banyak poliklinik memiliki satu dokter.

15. Class Resep Obat

Class Resep Obat memiliki atribut IDResepObat, TanggalResep, dan Dosis. Sedangkan metode atau operasi dari class ini antara lain GenerateIdResep(), SimpanResepObat(), UbahResepObat(), GetTanggalResep() dan GetDosis(). Class ini memiliki asosiasi dengan class pemeriksaan dengan multiplicity one to one, yang mana memiliki arti satu resep obat memiliki satu pemeriksaan. Selain itu class resep obat juga memiliki asosiasi dengan class obat dengan multiplicity one to many yang artinya satu resep obat memiliki banyak obat.

16. Class Obat

Class Obat memiliki atribut IDObat dan NamaObat. Sedangkan metode atau operasi dari class ini antara lain GenerateIdObat(), SimpanObat(), UbahObat(), dan GetNamaObat(). Class ini memiliki asosiasi dengan class pemeriksaan dan resep obat dengan multiplicity many to one, yang mana memiliki arti banyak alergi obat atau banyak obat dimiliki oleh satu resep obat atau satu pemeriksaan.

17. Class Rekam kesehatan

Class rekam kesehatan memiliki atribut IDRekamMedis. Sedangkan metode atau operasi dari class ini antara lain GenerateIdRM() dan GetIDRM(). Class ini memiliki asosiasi dengan class pasien dengan multiplicity one to one, yang mana memiliki arti satu pasien memiliki satu rekam medis. Selain itu class rekam kesehatan juga memiliki asosiasi dengan class pemeriksaan dengan multiplicity many to one yang artinya banyak pemeriksaan yang dimiliki oleh satu rekam kesehatan.

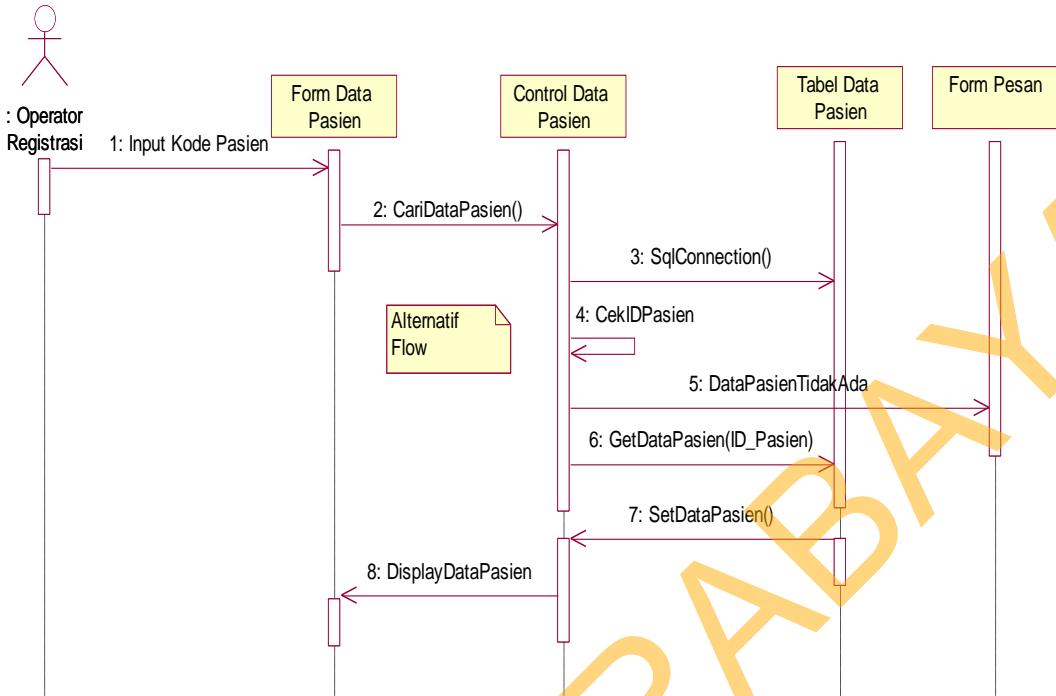
3.4.6 Sequence Diagram

Sequence diagram digunakan jika ingin melihat behavior beberapa objek di dalam use case tunggal. Sequence diagram juga sangat bagus untuk memperlihatkan interaksi antara objek yang ada pada sistem berdasarkan pada suatu urutan waktu, tetapi tidak terlalu bagus pada definisi yang rinci tentang behavior. Dalam sequence diagram juga terdapat stereotype antara boundary untuk mendefinisikan objek-objek GUI (Form-form), class control untuk mengontrol logika program dan entity mewakili penyimpanan data. Pada desain sistem informasi rekam kesehatan elektronik terpusat, terdapat beberapa sequence diagram yang memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam use case. Sequence diagram yang ada dalam sistem akan dijelaskan sebagai berikut :

A. Sequence Diagram Display Identitas Pasien

Pada sequence diagram yang ditunjuk pada gambar 3.44 terdapat beberapa objek yang berinteraksi antara lain Form DataPasien, ControlDataPasien, TabelDataPasien sebagai objek penyimpanan data dan FormPesan untuk menampilkan pesan. Bagian operator registrasi di rumah sakit merupakan aktor sebagai user yang berinteraksi dengan objek-objek tersebut.

Untuk lebih jelasnya dapat dilihat gambar 3.44 berikut.

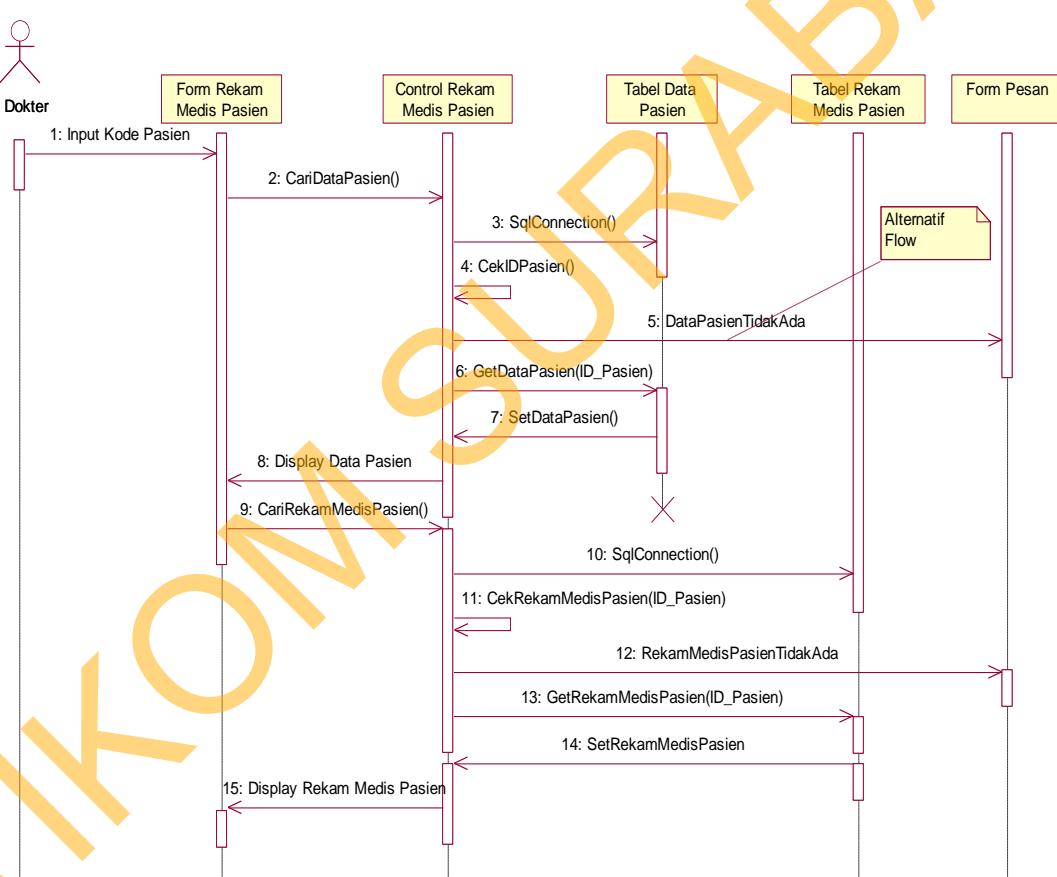


Gambar 3.44 *Sequence Diagram* Display Data Pasien

User clik data pasien maka tampil form DataPasien. User input kode pasien yang mana **disini** kode pasien adalah nomor RM (Rekam Medis) pasien. Setelah kode pasien diinputkan pada textbox yang disediakan, setelah user memilih tombol cari maka ControlDataPasien akan melakukan koneksi ke database dengan fungsi SqlConnection(). ControlDataPasien juga akan menjalankan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien sesuai dengan ID_Pasien yang diinputkan dari database. Apabila data pasien yang dicari ditemukan, ControlDataPasien akan melakukan fungsi SetDataPasien() dan kemudian menampilkan ke dalam Form DataPasien. Namun apabila data tidak ditemukan, akan ditampilkan pemberitahuan bahwa data tidak ditemukan pada form Pesan.

B. Sequence Diagram Display Rekam kesehatan

Pada sequence diagram yang akan ditunjukkan pada gambar 3.45, terdapat beberapa objek yang berinteraksi yaitu Form RekamMedisPasien, ControlRekamMedisPasien, TabelDataPasien sebagai objek penyimpanan data pasien, TabelRekamMedisPasien sebagai objek penyimpanan data rekam kesehatan dan Form Pesan untuk menampilkan pesan. Dokter merupakan aktor yang berinteraksi langsung dengan objek-objek tersebut.



Gambar 3.45 Sequence Diagram Display Data Rekam kesehatan

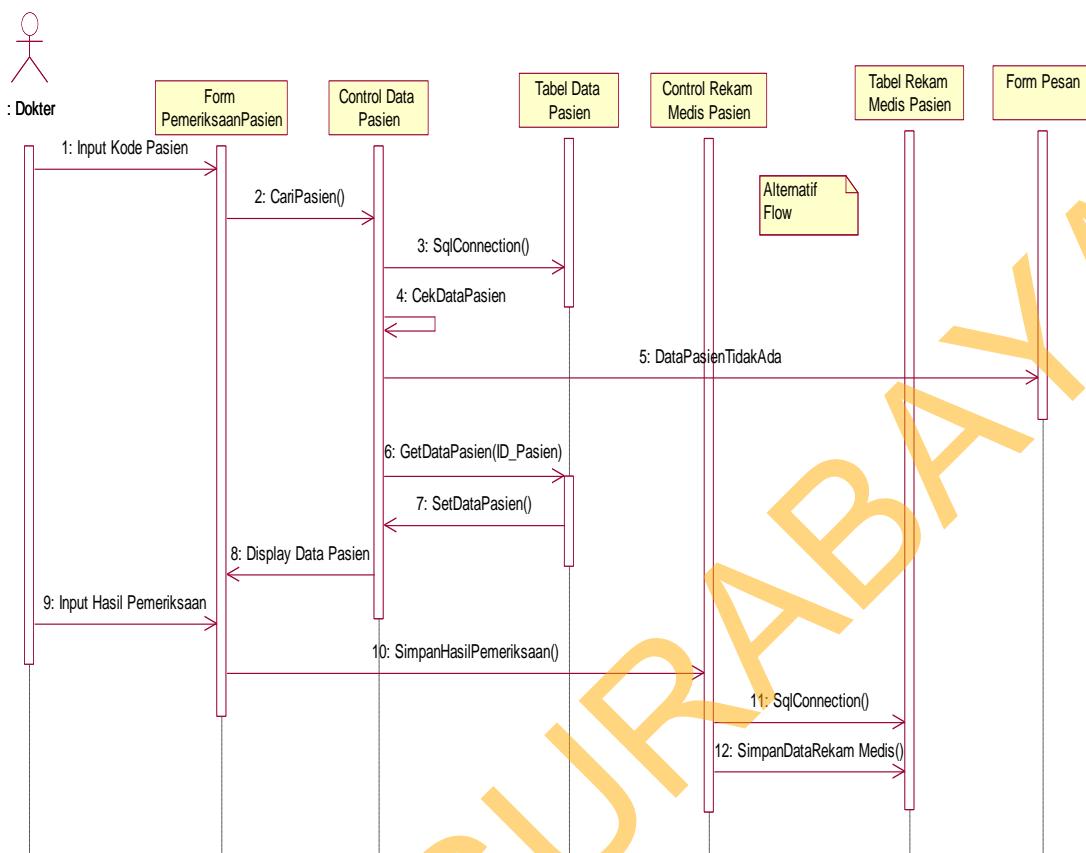
Interaksi diawali dengan user yaitu dokter inputkan kode pasien ke dalam textbox Form RekamMedisPasien, setelah user klik cari, maka ControlRekamMedisPasien menjalankan fungsi SqlConnection() untuk

menghubungkan ke dalam database dan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien berdasarkan ID_pasien yang diinputkan. Apabila ditemukan, ControlDataRekamMedisPasien akan menjalankan fungsi SetDataPasien() dan menampilkan data pasien yang dicari ke dalam form DataRekamMedisPasien. Namun apabila data pasien tidak ditemukan, maka akan ditampilkan pesan data tidak ditemukan pada form Pesan, sehingga proses tidak berlanjut ke proses berikutnya.

Apabila data pasien telah sesuai maka user akan click Tampilkan RekamMedis. ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() untuk menghubungkan ke dalam database dan fungsi GetDataRekamMedisPasien(ID_Pasien). Data rekam kesehatan akan dikembalikan ke form RekamMedisPasien oleh ControlRekamMedisPasien dengan menjalankan fungsi SetRekamMedisPasien().

C. Sequence Diagram Menyimpan Hasil Pemeriksaan

Pada **sequence diagram** yang akan ditunjukkan pada gambar 3.46, terdapat beberapa objek yang berinteraksi yaitu Form RekamMedisPasien, ControlRekamMedisPasien, ControlDataPasien, TabelDataPasien sebagai objek penyimpanan data pasien, TabelRekamMedisPasien sebagai objek penyimpanan data rekam kesehatan dan Form Pesan untuk menampilkan pesan. Dokter dan bagian rekam medis rumah sakit merupakan aktor yang berinteraksi langsung dengan objek-objek tersebut, namun disini dokter yang dicantumkan.



Gambar 3.46 *Sequence Diagram* Menyimpan Hasil Pemeriksaan

Interaksi ke objek diawali dengan user yaitu dokter menginputkan kode pasien ke dalam textbox Form RekamMedisPasien, setelah user klik cari, maka ControlRekamMedisPasien menjalankan fungsi SqlConnection() untuk menghubungkan ke dalam database dan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien berdasarkan ID_pasien yang diinputkan. Apabila ditemukan, ControlDataRekamMedisPasien akan menjalankan fungsi SetDataPasien() dan menampilkan data pasien yang dicari ke dalam form DataRekamMedisPasien. Namun apabila data pasien tidak ditemukan, maka akan ditampilkan pesan data tidak ditemukan pada form Pesan, sehingga proses tidak berlanjut ke proses berikutnya.

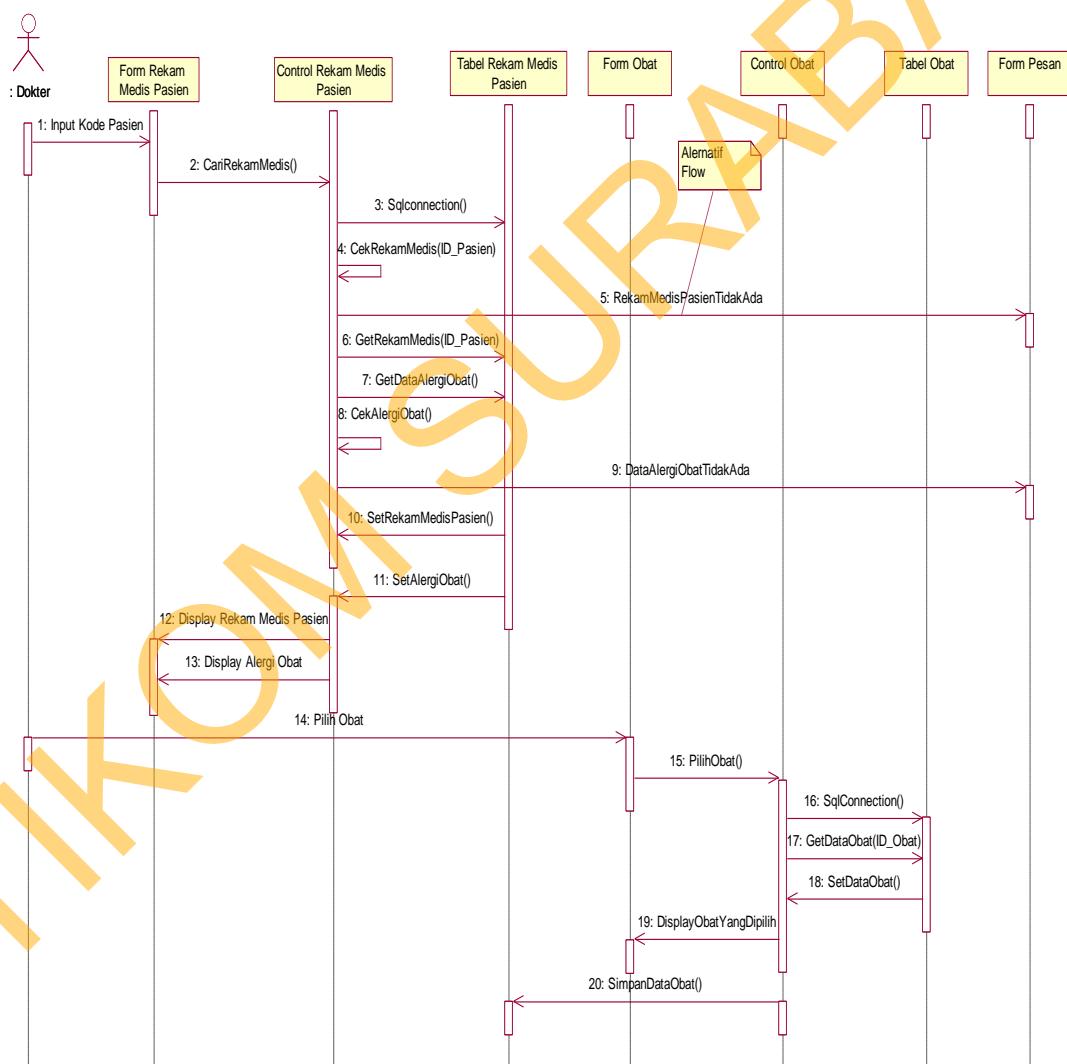
Apabila data rekam kesehatan telah ditampilkan, user akan klik tombol baru untuk menambahkan hasil pemeriksaan. Kemudian user menginputkan hasil pemeriksaan sesuai dengan petunjuk dan form rekam kesehatan. Setelah selesai menginputkan hasil pemeriksaan pasien yang terdiri dari hasil anamnesa, diagnosa, tindakan medis, pemeriksaan alergi obat, dan resep obat maka user memilih tombol simpan. ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() untuk menghubungkan ke dalam database serta menjalankan fungsi SimpanDataRekamMedis(). Data hasil pemeriksaan akan disimpan menjadi satu ke dalam tabel rekam medis.

D. Sequence Diagram Melakukan Tindakan Medis

Pada gambar 3.47 akan digambarkan sequence diagram melakukan tindakan medis. Adapun objek-objek yang berinteraksi antara lain form RekamMedisPasien, ControlRekamMedisPasien, tabel RekamMedisPasien, form Obat, ControlObat, tabel Obat, dan form Pesan. Bagian rekam medis rumah sakit atau disini yang dituliskan adalah dokter sebagai user merupakan aktor yang berinteraksi dengan objek-objek tersebut.

Proses melakukan tindakan medis dimulai dari user mencari data rekam kesehatan yang diberikan tindakan medis. Setelah user menginputkan kode pasien (Nomer RM), ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() untuk membuat koneksi ke database dan menjalankan fungsi GetRekamMedisPasien(ID_Pasien). Setelah data rekam medis ditemukan, user memeriksa riwayat alergi obat yang tercatat pada data rekam kesehatan. ControlRekamMedisPasien menjalankan fungsi GetAlergiObat(), apabila pasien

memiliki alergi obat, ControlRekamMedisPasien akan menampilkan data alergi obat ke dalam form DataRekamMedisPasien. Apabila data alergi obat telah diperiksa, dokter akan memilih obat yang akan diberikan kepada pasien kemudian menginputkan ke dalam form Obat. Setelah dokter memilih obat, ControlObat melakukan koneksi kembali ke database untuk mengakses tabel obat dengan menjalankan SqlConnection() dan fungsi GetDataObat().



Gambar 3.47 Sequence Diagram Melakukan Tindakan Medis

Database akan mengembalikan data obat dengan menjalankan fungsi SetDataObat(). ControlObat akan menampilkan obat yang telah dipilih dokter dan kemudian menyimpannya ke dalam tabel DataRekamMedisPasien sesuai dengan pemeriksaan yang dilakukan.

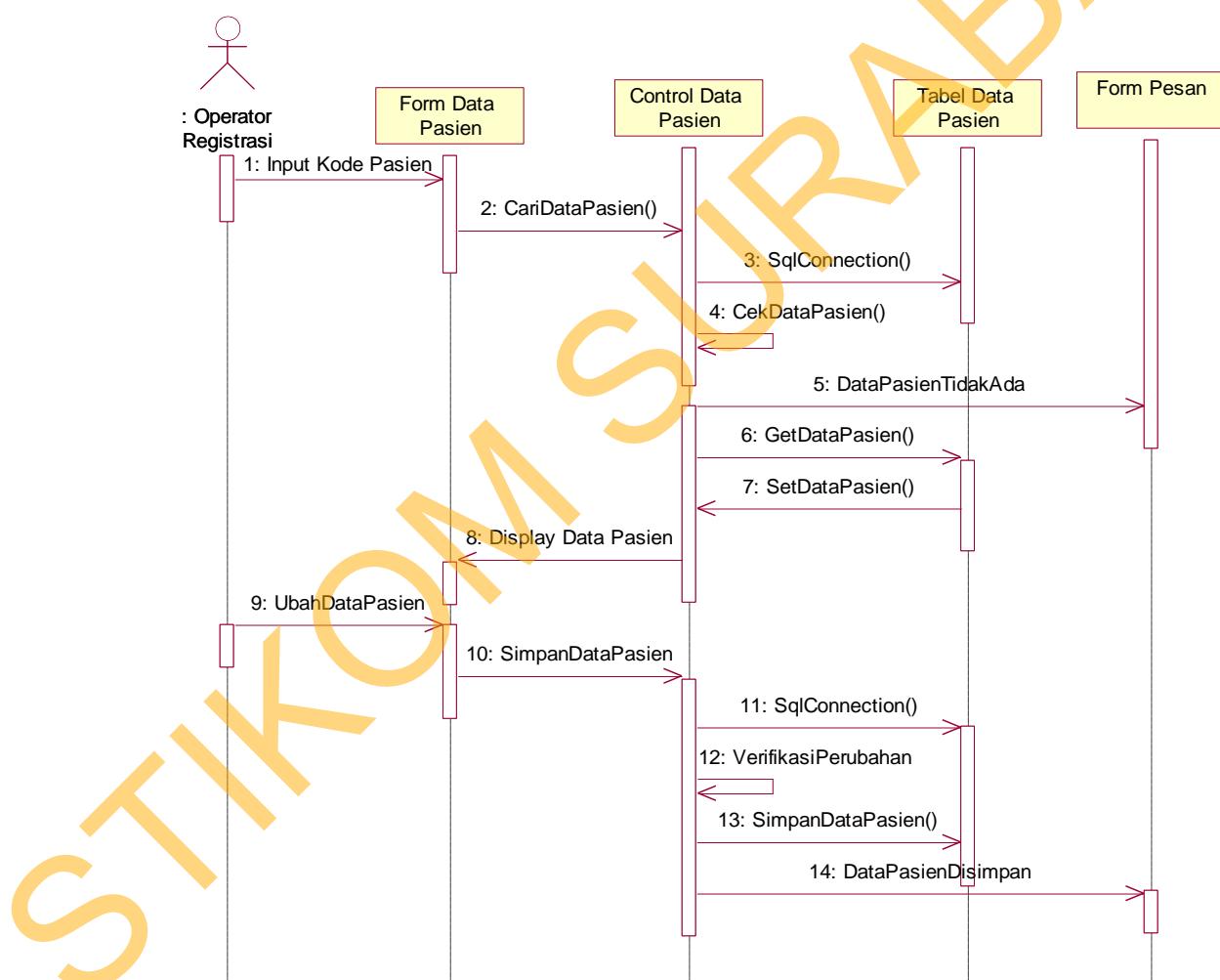
E. Sequence Diagram Melengkapi Identitas Pasien

Pada sequence diagram ini terdapat beberapa objek yang berinteraksi antara lain Form DataPasien, ControlDataPasien, TabelDataPasien sebagai objek penyimpanan data dan FormPesan untuk menampilkan pesan. Bagian operator registrasi di rumah sakit merupakan aktor sebagai user yang berinteraksi dengan objek-objek tersebut.

User memilih data pasien untuk menampilkan form DataPasien. User input kode pasien yang mana disini kode pasien adalah nomor RM (Rekam Medis) pasien. Setelah kode pasien diinputkan pada textbox yang disediakan, setelah user memilih tombol cari maka ControlDataPasien akan melakukan koneksi ke database dengan fungsi SqlConnection(). ControlDataPasien juga akan menjalankan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien sesuai dengan ID_Pasien yang diinputkan dari database. Apabila data pasien yang dicari ditemukan, ControlDataPasien akan melakukan fungsi SetDataPasien() dan kemudian menampilkan ke dalam Form DataPasien. Namun apabila data tidak ditemukan, akan ditampilkan pemberitahuan bahwa data tidak ditemukan pada form Pesan.

Untuk melengkapi data identitas pasien yang masih belum lengkap, user yaitu operator registrasi dapat memilih tombol ubah, maka akan ditampilkan data

pasien untuk diubah. Apabila perubahan telah selesai dilakukan, setelah user menekan tombol simpan, maka ControlDataPasien akan menjalankan fungsi SqlConnection() untuk membuka koneksi ke database yaitu tabel data pasien dan menjalankan fungsi SimpanDataPasien(). Setelah data disimpan, ControlDataPasien akan mengirimkan pemberitahuan melalui form Pesan bahwa perubahan data telah berhasil. Untuk lebih jelasnya dapat dilihat pada gambar berikut ini



Gambar 3.48 Sequence Diagram Melengkapi Identitas Pasien

F. Sequence Diagram Memeriksa Rujukan

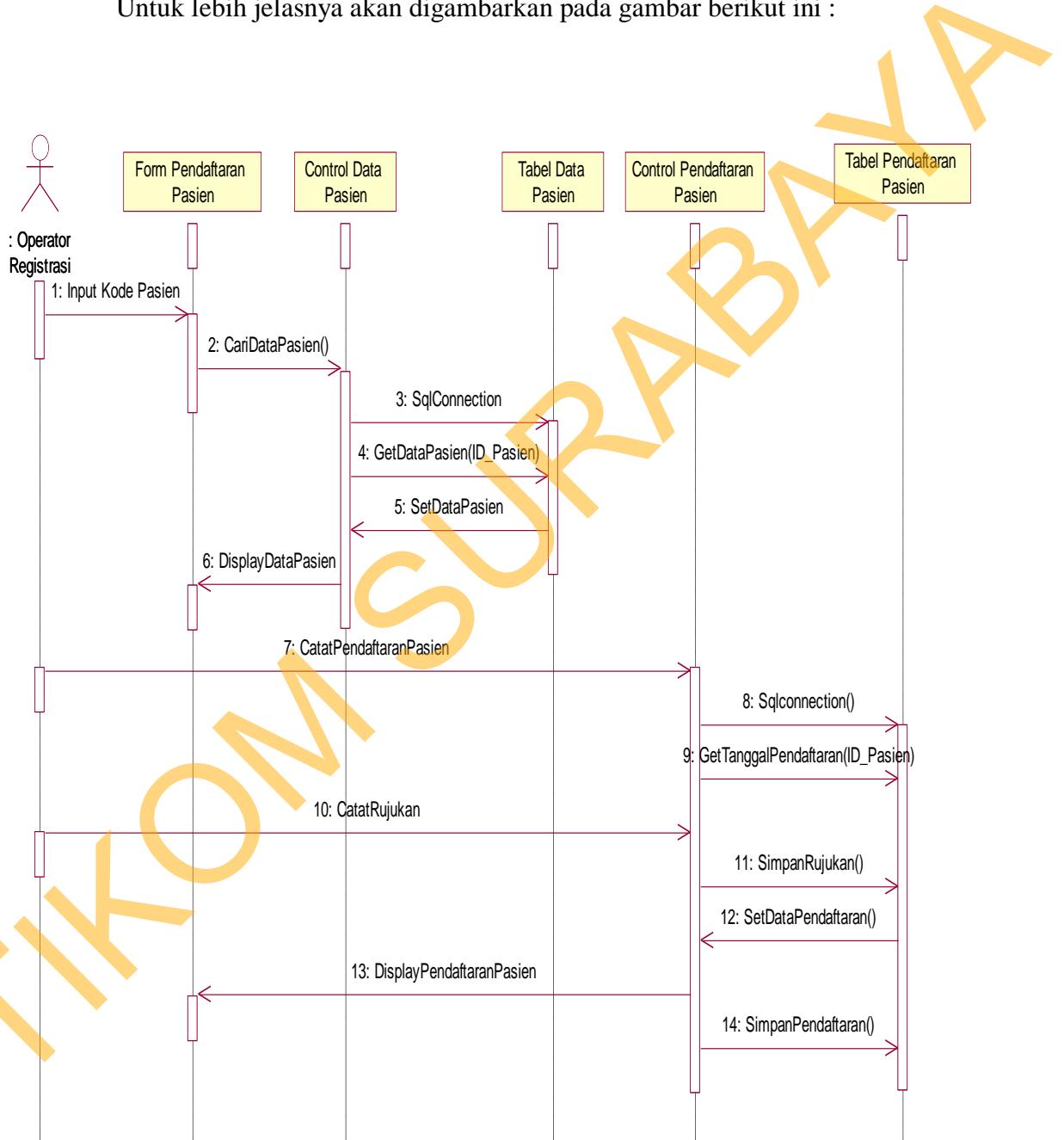
Pada sequence diagram ini terdapat beberapa objek yang berinteraksi antara lain Form DataPasien, ControlDataPasien, TabelDataPasien sebagai objek penyimpanan data dan FormPesan untuk menampilkan pesan. Bagian operator registrasi di rumah sakit merupakan aktor sebagai user yang berinteraksi dengan objek-objek tersebut.

User memilih data pasien untuk menampilkan form DataPasien. User input kode pasien yang mana disini kode pasien adalah nomor RM (Rekam Medis) pasien. Setelah kode pasien diinputkan pada textbox yang disediakan, setelah user memilih tombol cari maka ControlDataPasien akan melakukan koneksi ke database dengan fungsi SqlConnection(). ControlDataPasien juga akan menjalankan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien sesuai dengan ID_Pasien yang diinputkan dari database. Apabila data pasien yang dicari ditemukan, ControlDataPasien akan melakukan fungsi SetDataPasien() dan kemudian menampilkan ke dalam Form DataPasien. Namun apabila data tidak ditemukan, akan ditampilkan pemberitahuan bahwa data tidak ditemukan pada form Pesan.

User membuka menu pendaftaran pasien kemudian memilih daftar maka ControlPendaftaranPasien akan membuat koneksi ke database dengan menjalankan fungsi SqlConnection() dan fungsi GetTanggalPendaftaran(Id_Pasien). Apabila ada rujukan, rujukan akan ditambahkan pada saat pendaftaran dan user menyimpan ke dalam database pendaftaran melalui ControlPendaftaranPasien. Sebelum disimpan, data pendaftaran pasien akan ditampilkan oleh tabel PendaftaranPasien melalui

ControlPendaftaranPasien dengan menjalankan fungsi DisplayPendaftaranPasien(). Setelah proses selesai, pendaftaran akan disimpan dengan menjalankan fungsi SimpanPendaftaran().

Untuk lebih jelasnya akan digambarkan pada gambar berikut ini :



Gambar 3.49 Sequence Diagram Memeriksa rujukan

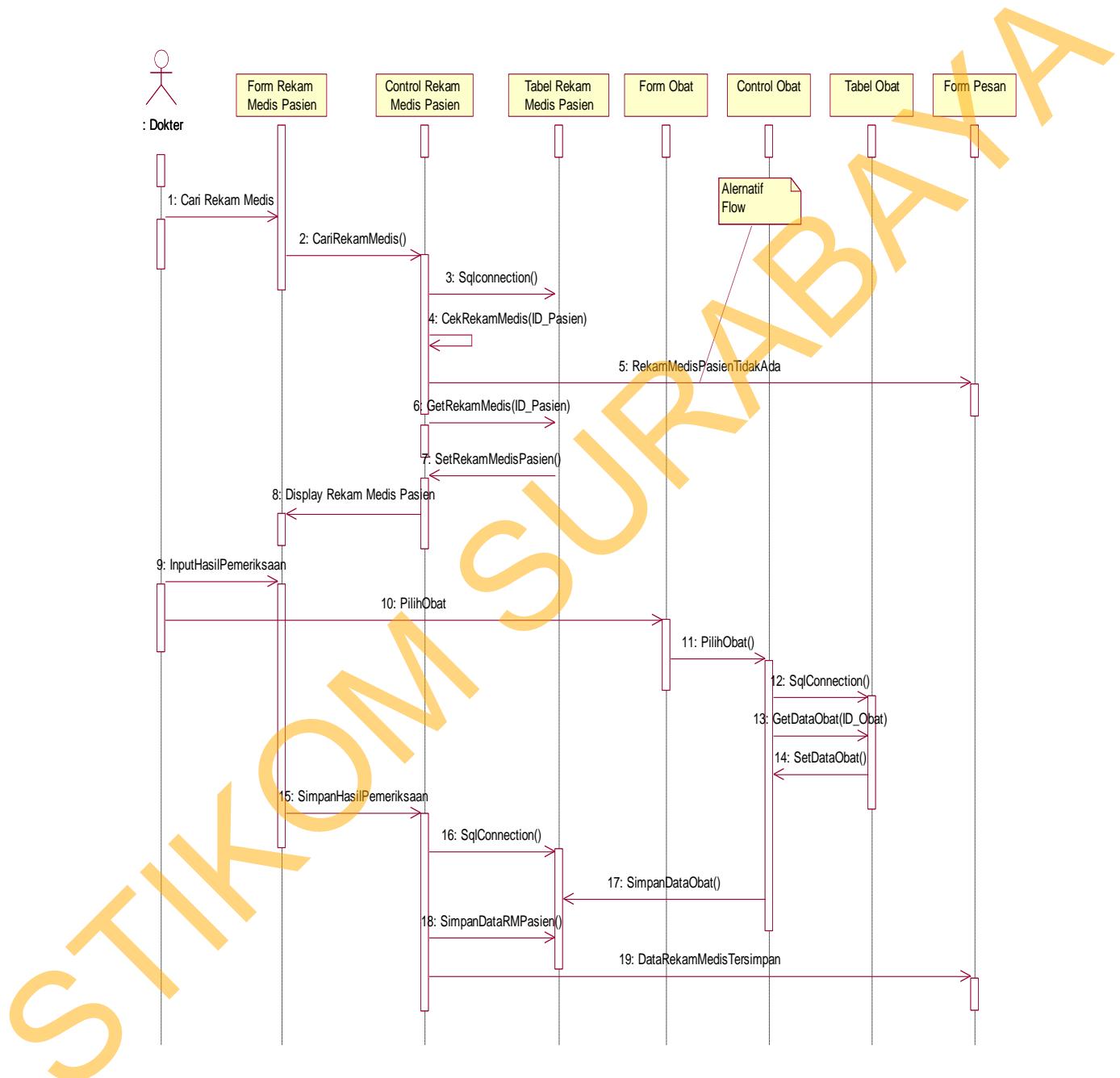
G. Sequence Diagram Menambahkan Data Rekam Medis

Pada sequence diagram yang akan ditunjukkan pada gambar 3.50, terdapat beberapa objek yang berinteraksi yaitu Form RekamMedisPasien, ControlRekamMedisPasien, TabelRekamMedisPasien sebagai objek penyimpanan data rekam kesehatan, Form Pesan untuk menampilkan pesan, form Obat, ControlObat, dan tabel Obat. Dokter dan bagian rekam medis rumah sakit merupakan aktor yang berinteraksi langsung dengan objek-objek tersebut, namun disini dokter yang dicantumkan.

Interaksi ke objek diawali dengan user yaitu dokter menginputkan kode pasien ke dalam textbox Form RekamMedisPasien, setelah user klik cari, maka ControlRekamMedisPasien menjalankan fungsi SqlConnection() untuk menghubungkan ke dalam database dan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien berdasarkan ID_pasien yang diinputkan. Apabila ditemukan, ControlDataRekamMedisPasien akan menjalankan fungsi SetDataPasien() dan menampilkan data pasien yang dicari ke dalam form DataRekamMedisPasien. Namun apabila data rekam kesehatan tidak ditemukan, maka akan ditampilkan pesan data tidak ditemukan pada form Pesan, sehingga proses tidak berlanjut ke proses berikutnya.

Apabila data rekam kesehatan telah ditampilkan, user akan klik tombol baru untuk menambahkan hasil pemeriksaan. Kemudian user menginputkan hasil pemeriksaan sesuai dengan petunjuk dan form rekam kesehatan. Setelah selesai menginputkan hasil pemeriksaan pasien yang terdiri dari hasil anamnesa, diagnosa, tindakan medis, pemeriksaan alergi obat, dan resep obat maka user memilih tombol simpan. ControlRekamMedisPasien akan menjalankan fungsi

SqlConnection() untuk menghubungkan ke dalam database serta menjalankan fungsi SimpanDataRekamMedis(). Data hasil pemeriksaan akan disimpan menjadi satu ke dalam tabel rekam medis.



Gambar 3.50 *Sequence Diagram* Menambah Data Rekam Medis

Pada gambar 3.50 digambarkan juga user menyimpan obat yang diberikan pada saat tindakan medis. Dokter akan memilih obat yang akan diberikan kepada pasien kemudian menginputkan ke dalam form Obat. Setelah dokter memilih obat, ControlObat melakukan koneksi kembali ke database untuk mengakses tabel obat dengan menjalankan `SqlConnection()` dan fungsi `GetDataObat()`.

Database akan mengembalikan data obat dengan menjalankan fungsi `SetDataObat()`. ControlObat akan menampilkan obat yang telah dipilih dokter dan kemudian menyimpannya ke dalam tabel DataRekamMedisPasien sesuai dengan pemeriksaan yang dilakukan. Seluruh hasil pemeriksaan sampai dengan tindakan medis akan disimpan melalui ControlRekamMedisPasien dengan menjalankan fungsi `SimpanHasilPemeriksaan()` dan juga menampilkan pemberitahuan data telah tersimpan pada form Pesan.

H. Sequence Diagram Pendaftaran Pasien

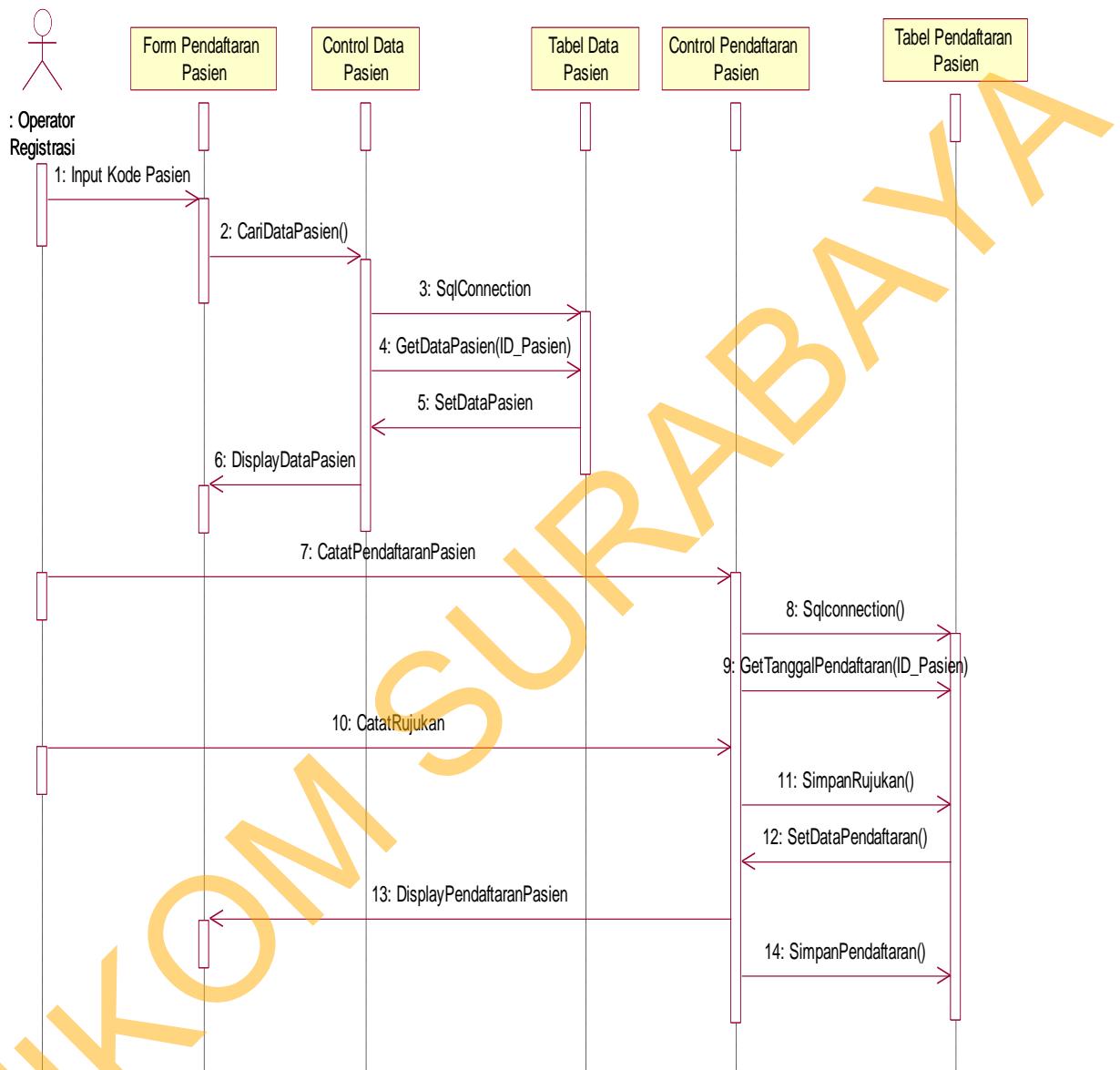
Pada sequence diagram ini terdapat beberapa objek yang berinteraksi antara lain **Form DataPasien**, **ControlDataPasien**, **TabelDataPasien** sebagai objek penyimpanan data dan **FormPesanan** untuk menampilkan pesan. Bagian operator registrasi di rumah sakit merupakan aktor sebagai user yang berinteraksi dengan objek-objek tersebut.

Pada gambar 3.51 digambarkan user memilih data pasien untuk menampilkan form DataPasien. User input kode pasien yang mana disini kode pasien adalah nomor RM (Rekam Medis) pasien. Setelah kode pasien diinputkan pada textbox yang disediakan, setelah user memilih tombol cari maka **ControlDataPasien** akan melakukan koneksi ke database dengan fungsi

SqlConnection(). ControlDataPasien juga akan menjalankan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien sesuai dengan ID_Pasien yang diinputkan dari database. Apabila data pasien yang dicari ditemukan, ControlDataPasien akan melakukan fungsi SetDataPasien() dan kemudian menampilkan ke dalam Form DataPasien. Namun apabila data tidak ditemukan, akan ditampilkan pemberitahuan bahwa data tidak ditemukan pada form Pesan.

User membuka menu pendaftaran pasien kemudian memilih daftar maka ControlPendaftaranPasien akan membuat koneksi ke database dengan menjalankan fungsi SqlConnection() dan fungsi GetTanggalPendaftaran(Id_Pasien). Apabila ada rujukan, rujukan akan ditambahkan pada saat pendaftaran dan user menyimpan ke dalam database pendaftaran melalui ControlPendaftaranPasien. Sebelum disimpan, data pendaftaran pasien akan ditampilkan oleh tabel PendaftaranPasien melalui ControlPendaftaranPasien dengan menjalankan fungsi DisplayPendaftaranPasien(). Setelah proses selesai, pendaftaran akan disimpan dengan menjalankan fungsi SimpanPendaftaran().

Untuk lebih jelasnya akan digambarkan pada gambar berikut ini :



Gambar 3.51 Sequence Diagram Menambahkan Pendaftaran Pasien

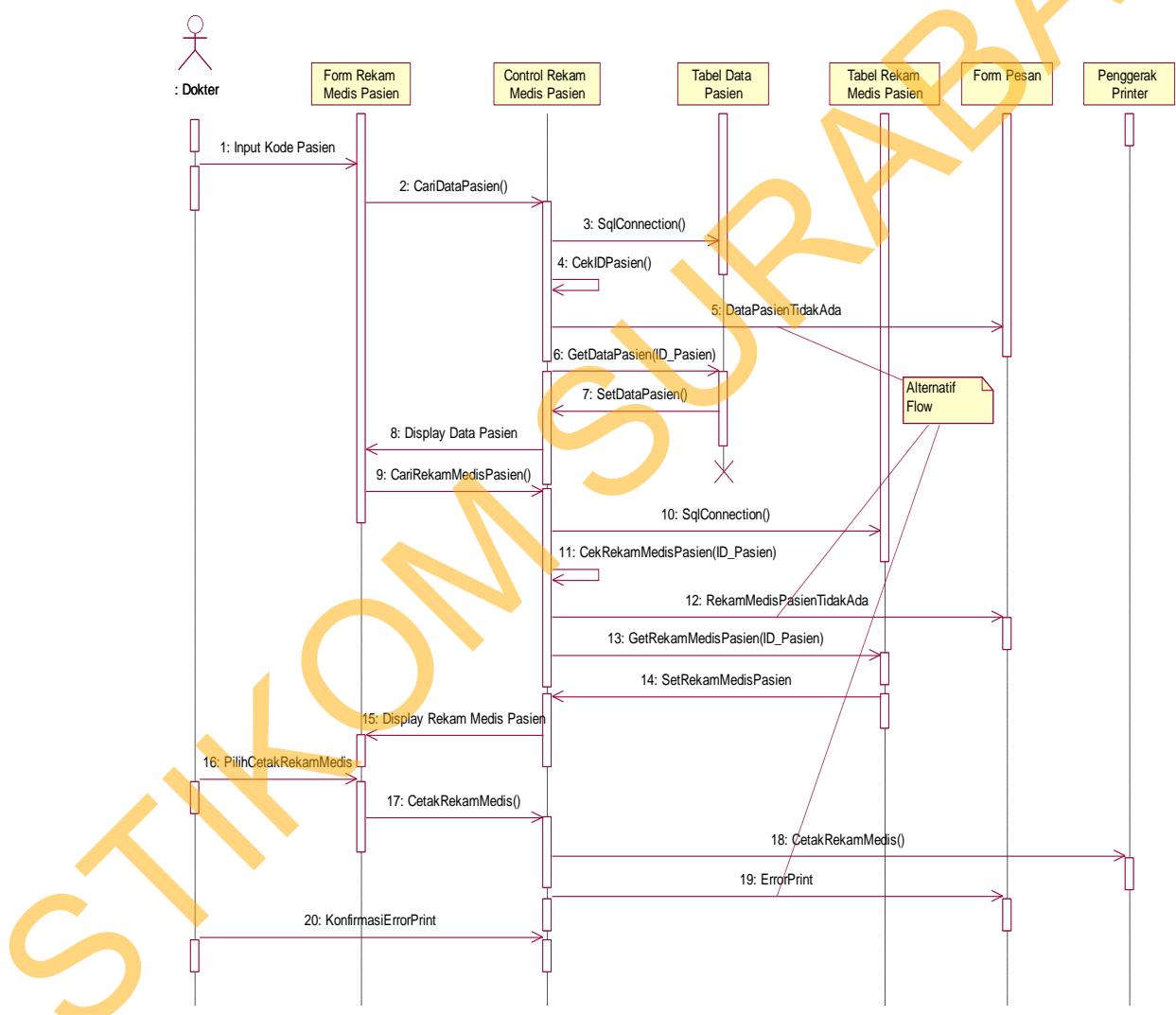
I. Sequence Diagram Mencetak Hasil Pemeriksaan

Pada sequence diagram yang akan ditunjukkan pada gambar 3.52, terdapat beberapa objek yang berinteraksi yaitu Form RekamMedisPasien, ControlRekamMedisPasien, TabelDataPasien sebagai objek penyimpanan data pasien, TabelRekamMedisPasien sebagai objek penyimpanan data rekam kesehatan, Form Pesan untuk menampilkan pesan dan objek printer sebagai media untuk mencetak. Dokter merupakan aktor yang berinteraksi langsung dengan objek-objek tersebut.

Interaksi diawali dengan user yaitu dokter inputkan kode pasien ke dalam textbox Form RekamMedisPasien, setelah user klik cari, maka ControlRekamMedisPasien menjalankan fungsi SqlConnection() untuk menghubungkan ke dalam database dan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien berdasarkan ID_pasien yang diinputkan. Apabila ditemukan, ControlDataRekamMedisPasien akan menjalankan fungsi SetDataPasien() dan menampilkan data pasien yang dicari ke dalam form DataRekamMedisPasien. Namun apabila data pasien tidak ditemukan, maka akan ditampilkan pesan data tidak ditemukan pada form Pesan, sehingga proses tidak berlanjut ke proses berikutnya.

Apabila data pasien telah sesuai maka user akan click Tampilkan RekamMedis. ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() untuk menghubungkan ke dalam database dan fungsi GetDataRekamMedisPasien(ID_Pasien). Data rekam kesehatan akan dikembalikan ke form RekamMedisPasien oleh ControlRekamMedisPasien dengan menjalankan fungsi SetRekamMedisPasien().

Dokter kemudian memilih data rekam medis yang akan dicetak kemudian memilih tombol cetak, ControlRekamMedisPasien akan menjalankan fungsi CetakRekamMedis() untuk memberikan perintah untuk memulai mencetak oleh printer. Apabila terjadi error pada saat mencetak, printer akan mengirimkan pesan melalui ControlRekamMedisPasien ke form Pesan. Setelah error dikonfirmasi oleh user, proses cetak rekam medis dilanjutkan.



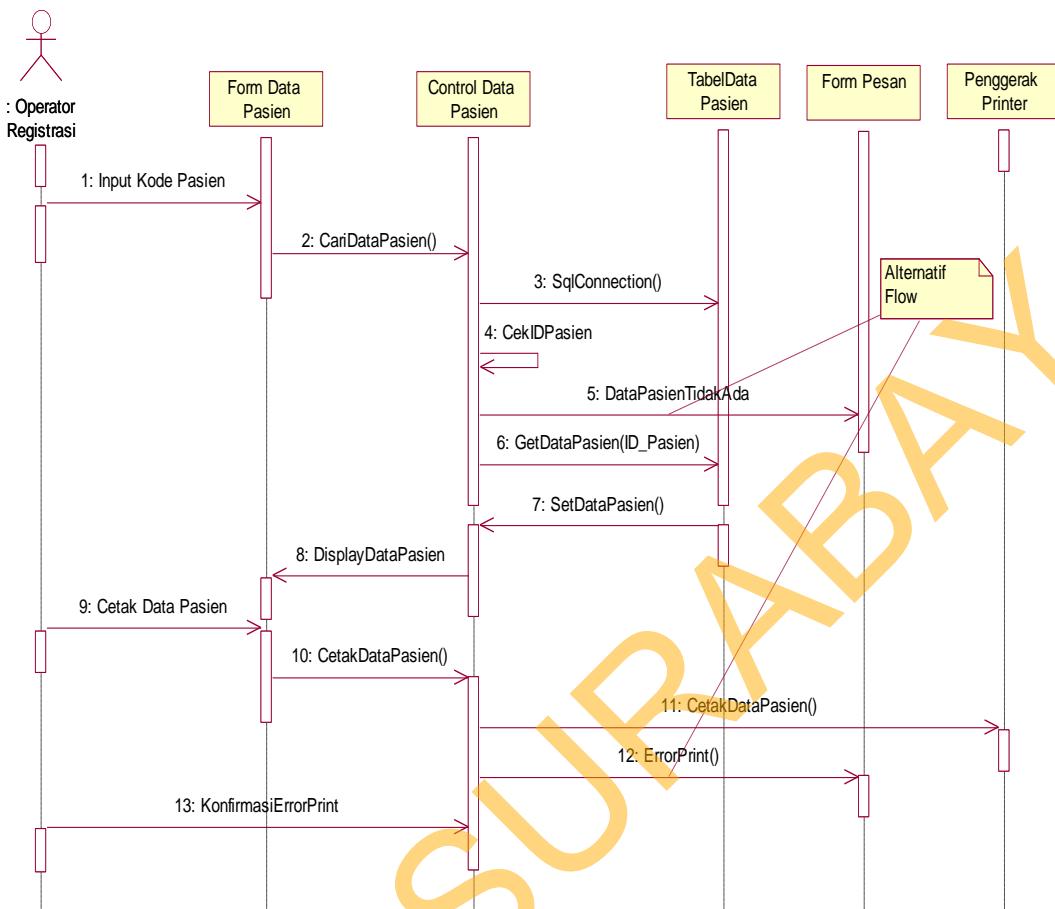
Gambar 3.52 Sequence Diagram Mencetak Hasil Pemeriksaan

J. Sequence Diagram Mencetak Identitas Pasien

Pada sequence diagram ini terdapat beberapa objek yang berinteraksi antara lain Form DataPasien, ControlDataPasien, TabelDataPasien sebagai objek penyimpanan data, FormPesan untuk menampilkan pesan dan Penggerak printer untuk mencetak. Bagian operator registrasi di rumah sakit merupakan aktor sebagai user yang berinteraksi dengan objek-objek tersebut.

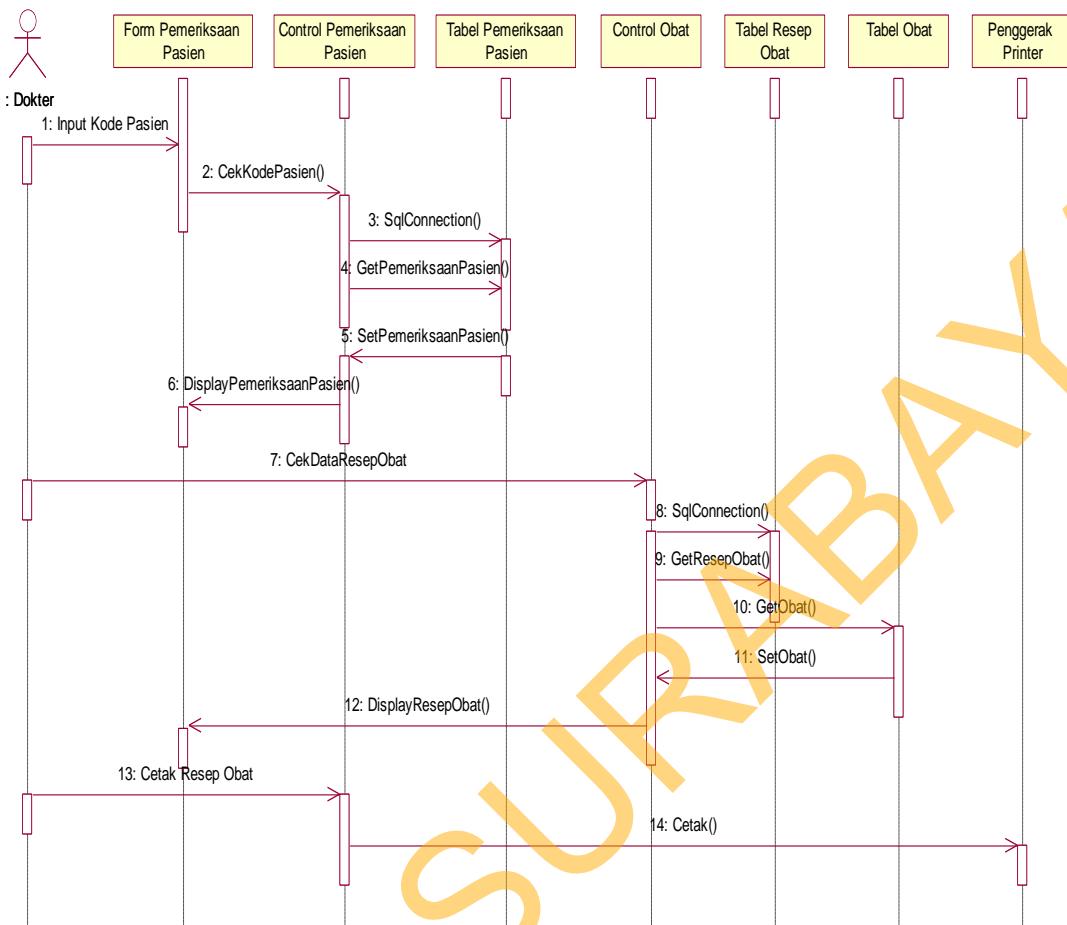
User clik data pasien maka tampil form DataPasien. User input kode pasien yang mana disini kode pasien adalah nomor RM (Rekam Medis) pasien. Setelah kode pasien diinputkan pada textbox yang disediakan, setelah user memilih tombol cari maka ControlDataPasien akan melakukan koneksi ke database dengan fungsi SqlConnection(). ControlDataPasien juga akan menjalankan fungsi GetDataPasien(ID_Pasien) untuk mendapatkan data pasien sesuai dengan ID_Pasien yang diinputkan dari database. Apabila data pasien yang dicari ditemukan, ControlDataPasien akan melakukan fungsi SetDataPasien() dan kemudian menampilkan ke dalam Form DataPasien. Namun apabila data tidak ditemukan, akan ditampilkan pemberitahuan bahwa data tidak ditemukan pada form Pesan.

Setelah data pasien ditampilkan, user memilih menu cetak data pasien kemudian memilih tombol cetak. ControlDataPasien akan menjalankan fungsi CetakDataPasien() untuk menggerakkan penggerak printer untuk memulai mencetak data pasien. Apabila terjadi error pada saat mencetak, maka ControlDataPasien akan menjalankan fungsi ErrorPrint() dan menampilkan pemberitahuan ke form Pesan. Apabila error telah di konfirmasi, proses mencetak akan dilanjutkan. Semua proses akan digambarkan pada gambar 3.53.

Gambar 3.53 *Sequence Diagram* Mencetak Identitas Pasien

K. Sequence Diagram Mencetak Resep Obat

Sequence diagram menampilkan interaksi antara beberapa objek dengan aktor. Objek-objek yang berinteraksi antara lain form PemeriksaanPasien, ControlPemeriksaanPasien, tabel PemeriksaanPasien, ControlObat, tabel ResepObat, tabelObat, dan penggerak printer untuk melakukan percetakan. Adapun aktor sebagai user yang berinteraksi dengan objek tersebut adalah dokter. Pada gambar 3.54 akan digambarkan sequence diagram untuk mencetak resep obat.



Gambar 3.54 *Sequence Diagram* Mencetak Resep Obat

Interaksi user dengan sistem dimulai dari user membuka pemeriksaan pasien dengan menginputkan kode pasien, setelah user memilih tombil cari, ControlPemeriksaanPasien akan menjalankan fungsi SqlConnection() dan GetPemeriksaanPasien() untuk menampilkan pemeriksaan pasien. Tabel pemeriksaan pasien akan mengembalikan data pemeriksaan pasien yang diinginkan dengan menjalankan fungsi SetPemeriksaanPasien(). Dari data pemeriksaan pasien, dokter akan memeriksa resep obat yang diberikan pada saat pemeriksaan tersebut. ControlObat akan menjalankan fungsi GetResepObat() untuk mendapatkan resep obat yang diberikan dari tabel ResepObat dan menjalankan

fungsi GetObat dari tabel Obat. Dari tabel resep obat dan obat, melalui ControlObat akan dijalankan fungsi SetResepObat() dan mengembalikan DisplayResepObat ke form Pemeriksaan. Untuk mencetak resep obat, user akan memilih tombol Cetak. ControlPemeriksaanPasien akan menjalankan fungsi Cetak() untuk memulai mencetak.

3.4.7 *Collaboration Diagram*

Collaboration diagram merupakan diagram yang digunakan sebagai alat untuk menggambarkan interaksi yang mengungkapkan keputusan mengenai perilaku sistem. Tipe diagram ini menekankan pada hubungan antar objek, sedangkan sequence diagram menekankan pada urutan kejadian. Dalam satu collaboration diagram terdapat beberapa *object*, *link*, dan *message*. Antara collaboration diagram dan sequence diagram bisa saling mengisi, dan atas dasar itulah dalam desain sistem ini collaboration diagram akan digunakan juga untuk memperlihatkan interaksi antar objek dalam satu use case.

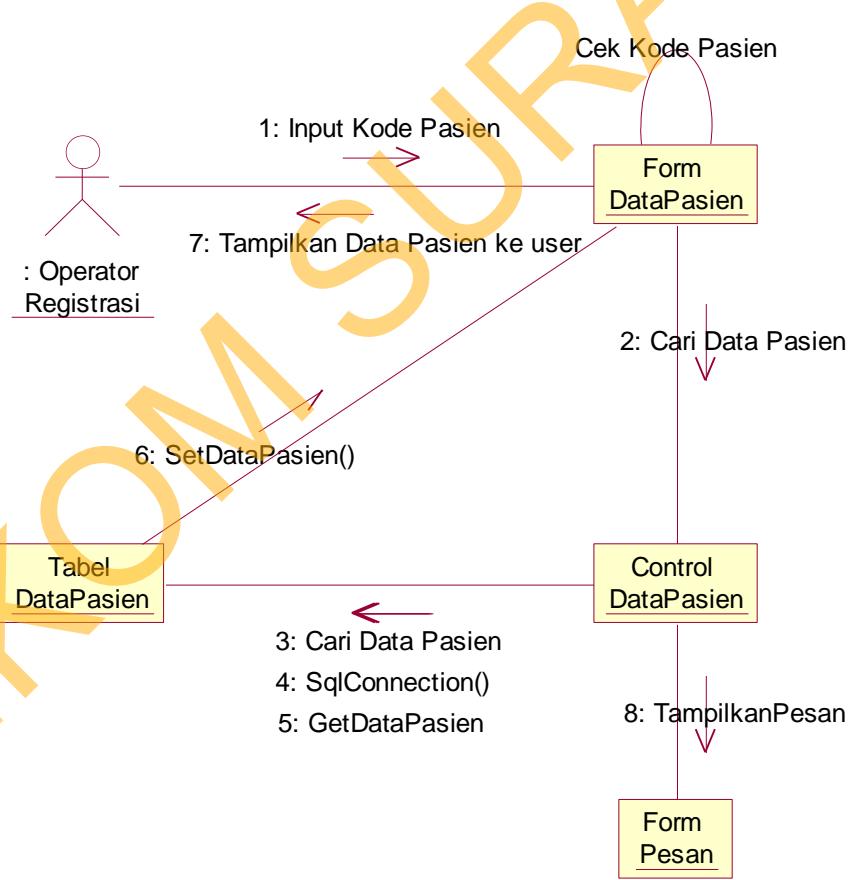
Collaboration diagram yang ada pada desain sistem informasi rekam kesehatan elektronik terpusat, antara lain sebagai berikut:

A. **Collaboration Diagram Display Data Pasien**

Pada collaboration diagram display data pasien menggambarkan interaksi atau hubungan antar objek yang terjadi pada use case display data pasien. Objek yang ada antara lain FormDataPasien, ControlDataPasien, TabelDataPasien dan FormPesan. Bagian operator registrasi adalah user yang berinteraksi dengan objek yang ada, diawali dengan user input kode pasien pada FormDataPasien yang akan

dicari datanya kemudian berdasarkan kode pasien akan diperiksa apakah data pasien ada. Melalui ControlDataPasien akan dijalankan fungsi CariDataPasien(), dan dijalankan fungsi SqlConnection() untuk membuka koneksi ke tabel DataPasien serta menjalankan GetDataPasien(). Apabila data ditemukan, tabel DataPasien akan mengembalikan data pasien dengan menjalankan fungsi SetDataPasien(). Apabila data tidak ditemukan, ControlDataPasien akan menampilkan pemberitahuan data pasien yang dicari tidak ditemukan. Dari FormDataPasien menampilkan data pasien kepada user.

Untuk lebih jelasnya akan digambarkan pada gambar 3.55 berikut.

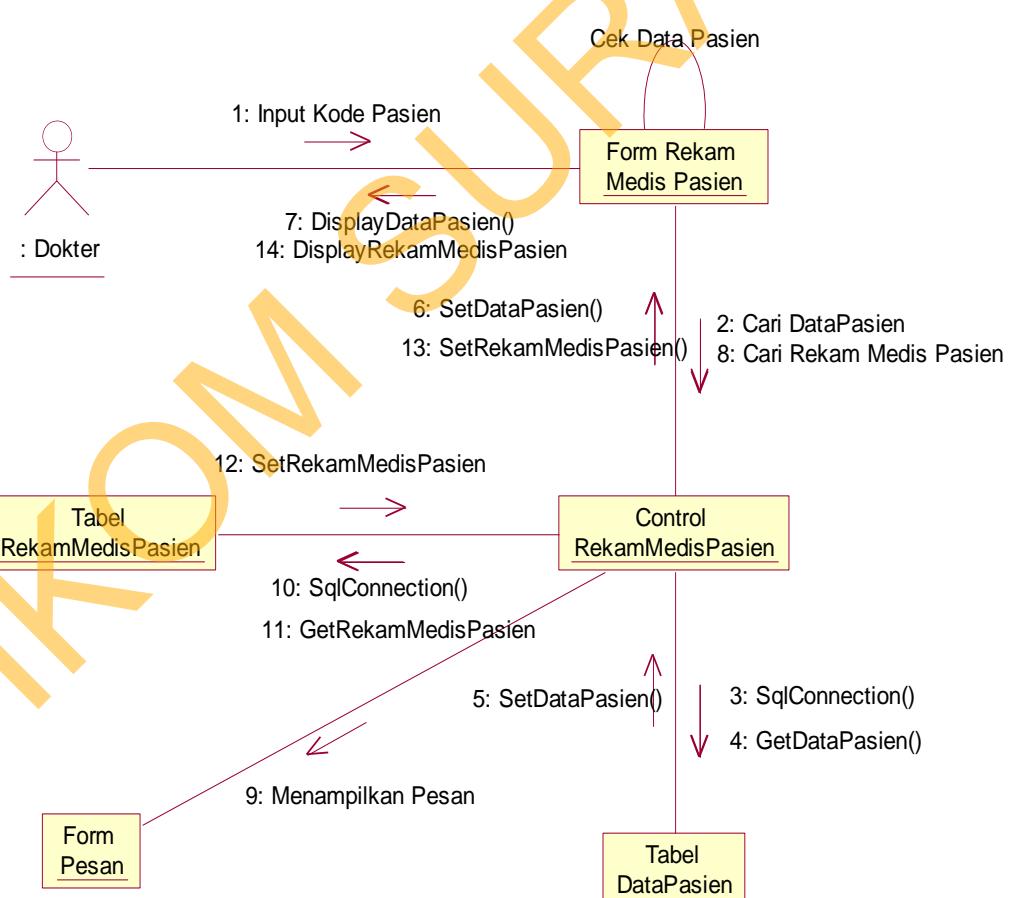


Gambar 3.55 *Collaboration Diagram* Display Data Pasien

B. Collaboration Diagram Display Rekam kesehatan

Untuk display rekam kesehatan, adapun objek-objek yang saling berinteraksi antara lain FormRekamMedisPasien, ControlRekamMedisPasien, TabelDataPasien, TabelRekamMedisPasien dan FormPesan. Untuk menampilkan rekam kesehatan aktor yang berinteraksi dengan objek-objek tersebut adalah dokter. Karena hanya dokter yang boleh mengetahui rekam medis seorang pasien. Seperti yang kita ketahui rekam kesehatan bersifat rahasia.

Pada gambar 3.56 akan digambarkan collaboration diagram untuk display data rekam kesehatan.



Gambar 3.56 *Collaboration Diagram* Display Rekam kesehatan

Interaksi yang digambarkan pada gambar 3.56 dimulai dari user ingin memeriksa dan menampilkan data pasien terlebih dahulu. User inputkan kode pasien pada FormRekamMedisPasien. Setelah user klik tombol tampil, ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() untuk membuat koneksi ke tabel data pasien dan menjalankan fungsi GetDataPasien() untuk meminta data pasiennya. Tabel DataPasien mengembalikan data dengan menjalankan fungsi SetDataPasien() dan melalui Form DataRekamMedisPasien akan ditampilkan DataPasien yang dicari. Setelah data pasien dibuka, user akan menampilkan data rekam medisnya dengan memilih tombol tampilkan rekam medis. ControlRekamMedis akan menjalankan fungsi SqlConnection() lagi untuk membuat koneksi ke database yaitu tabel RekamMedisPasien.

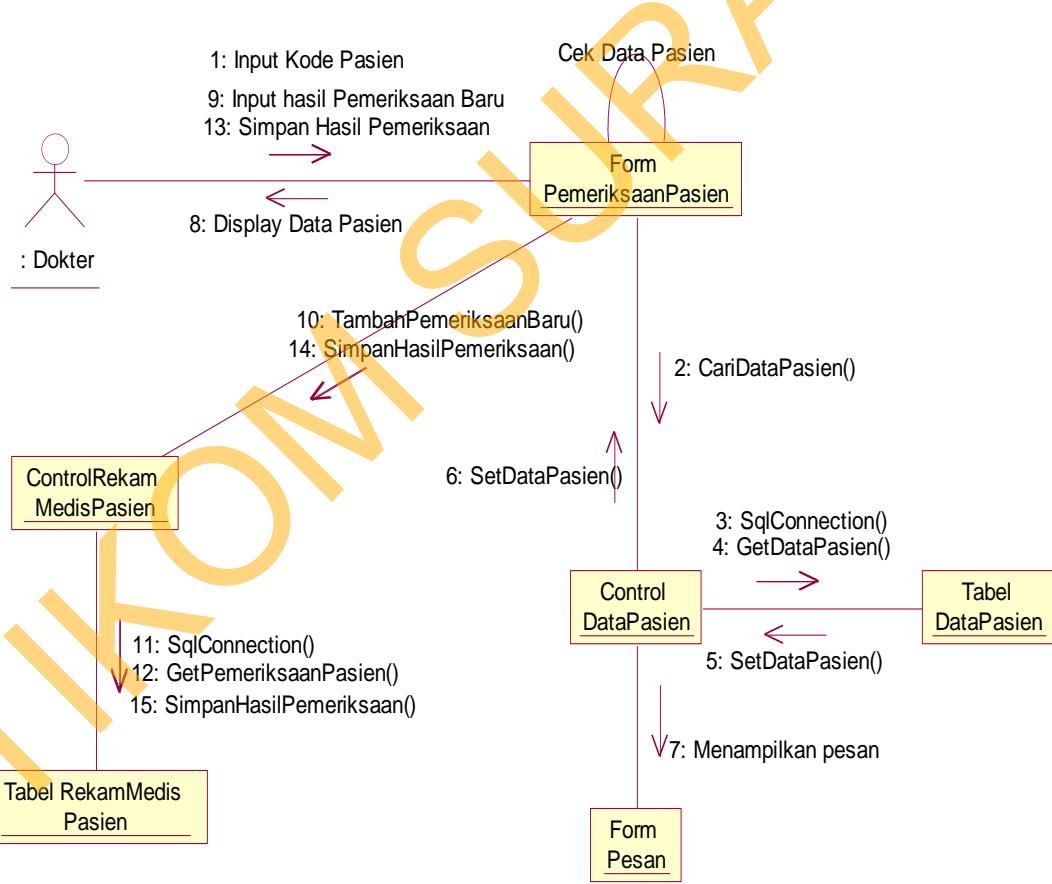
Selain itu untuk membaca data rekam kesehatan, akan dijalankan GetDataRekamMedisPasien() oleh ControlRekamMedisPasien(). Tabel rekam medis mengembalikan data rekam kesehatan melalui ControlRekamMedisPasien dan dijalankan SetRekamMedisPasien(). Data rekam kesehatan akan ditampilkan pada FormDataRekamMedisPasien.

Pada saat pemeriksaan data pasien dan data rekam kesehatan, apabila data yang dicari tidak ditemukan, maka ControlRekamMedisPasien akan menjalankan perintah untuk menampilkan pemberitahuan bahwa data yang dicari tidak ditemukan dan pemberitahuan tersebut ditampilkan pada FormPesan.

C. Collaboratin Diagram Menyimpan Hasil Pemeriksaan

Pada gambar 3.57 akan dijelaskan untuk collaboration diagram menyimpan hasil pemeriksaan beberapa objek yang saling berinteraksi antara lain FormPemeriksaanPasien, ControlDataPasien, TabelDataPasien, ControlRekamMedisPasien, TabelRekamMedisPasien, dan TabelPesan. Bagian rekam medis atau yang disebutkan disini adalah dokter merupakan aktor atau user yang berinteraksi dengan objek-objek yang telah disebutkan.

Berikut ini merupakan gambar 3.57 collaboration diagram menyimpan hasil pemeriksaan pasien.



Gambar 3.57 Collaboration Diagram Menyimpan Hasil Pemeriksaan

Interaksi user dimulai dengan membuka membuka data pasien yang diperiksa. User input kode pasien dan memilih tombol cari. ControlDataPasien akan menjalankan pencarian data pasien dengan menjalankan fungsi SqlConnection() untuk membuat koneksi ke database dan GetDataPasien() untuk menampilkan data pasien yang dicari berdasarkan kode pasien yang di inputkan. Dari tabelPasien, akan dikembalikan data pasien yang dicari dengan menjalankan fungsi SetDataPasien(). Melalui FormPemeriksaanPasien akan ditampilkan data pasiennya. Apabila data pasien yang dicari telah sesuai, user akan membuka data rekam medisnya dengan memilih tampil rekam kesehatan. ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() dan GetRekamMedisPasien() untuk menampilkan data rekam kesehatan dari dalam database yaitu tabelRekamMedisPasien. Rekam kesehatan dikembalikan ke formPemeriksaanPasien dengan menjalankan fungsi SetRekamMedisPasien dan akan ditampilkan pada FormPemeriksaanPasien.

User akan memilih menu baru pada formPemeriksaanPasien untuk menambahkan hasil pemeriksaan pasien. User inputkan hasil pemeriksaan kemudian memilih simpan. ControlRekamMedisPasien akan menjalankan fungsi SimpanRekamMedis() untuk menyimpan data hasil pemeriksaan ke dalam tabelDataRekamMedisPasien yang bersangkutan.

Seluruh pemberitahuan baik itu pemberitahuan data tidak ditemukan dan data telah disimpan, akan ditampilkan pada satu form yaitu formPesan.

D. Collaboration Diagram Melakukan Tindakan Medis

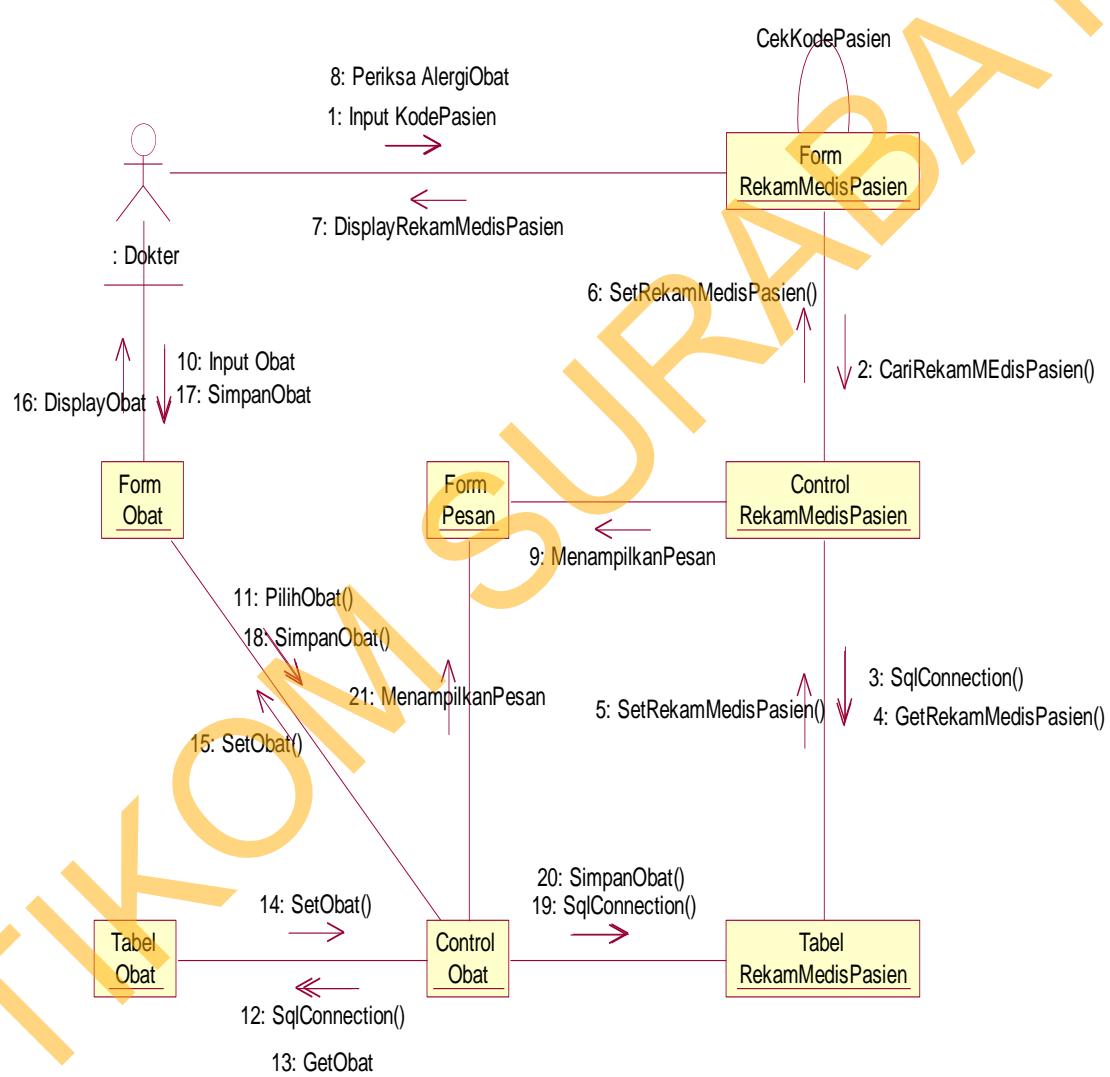
Collaboration diagram melakukan tindakan medis menggambarkan interaksi-interaksi antar objek yang terlibat di dalamnya antara lain: FormRekamMedisPasien, ControlRekamMedisPasien, tabelRekamMedisPasien, FormObat, ControlObat, tabelObat dan FomPesan. Aktor yang berinteraksi dengan objek-objek tersebut adalah dokter. Dokter merupakan aktor yang melakukan tindakan medis dan mencatat seluruh kegiatan yang dilakukan.

Pada gambar 3.58 akan digambarkan proses yang terjadi pada saat dokter melakukan tindakan medis. Interaksi diawali dengan user input kode pasien pada formRekamMedisPasien. Saat proses pencarian, ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() untuk membuat koneksi ke tabelRekamMedisPasien dan fungsi GetRekamMedisPasien() untuk mendapatkan data rekam kesehatan yang bersangkutan. Apabila data rekam kesehatan ditemukan, tabelRekamMedisPasien akan mengembalikan data dengan menjalankan fungsi SetRekamMedisPasien(). Data rekam kesehatan akan ditampilkan melalui formRekamMedisPasien. Pada saat tindakan medis, dokter juga akan memeriksa apakah pasien sebelumnya memiliki riwayat alergi obat atau tidak. Pemeriksaan tersebut akan dilihat dari data rekam medis yang dimiliki oleh pasien. Dokter akan memilih obat yang akan diberikan kepada pasien untuk menangani penyakit pasien.

Apabila user telah memilih obat, ControlObat akan menjalankan fungsi SqlConnection() dan GetObat() untuk menampilkan data obat. Tabel obat mengembalikan data dengan menjalankan SetObat dan ditampilkan pada user melalui ForObat. Obat-obat yang dipilih akan disimpan ke dalam

tabelRekamMedisPasien. ControlObat akan menjalankan fungsi SqlConnection() lagi untuk koneksi ke tabelRekamMedisPasien dan fungsi SimpanObat() agar obat yang diberikan pada saat tindakan medis tercatat pada rekam kesehatan.

Berikut merupakan gambar 3.58 untuk collaboration diagram melakukan tindakan medis.

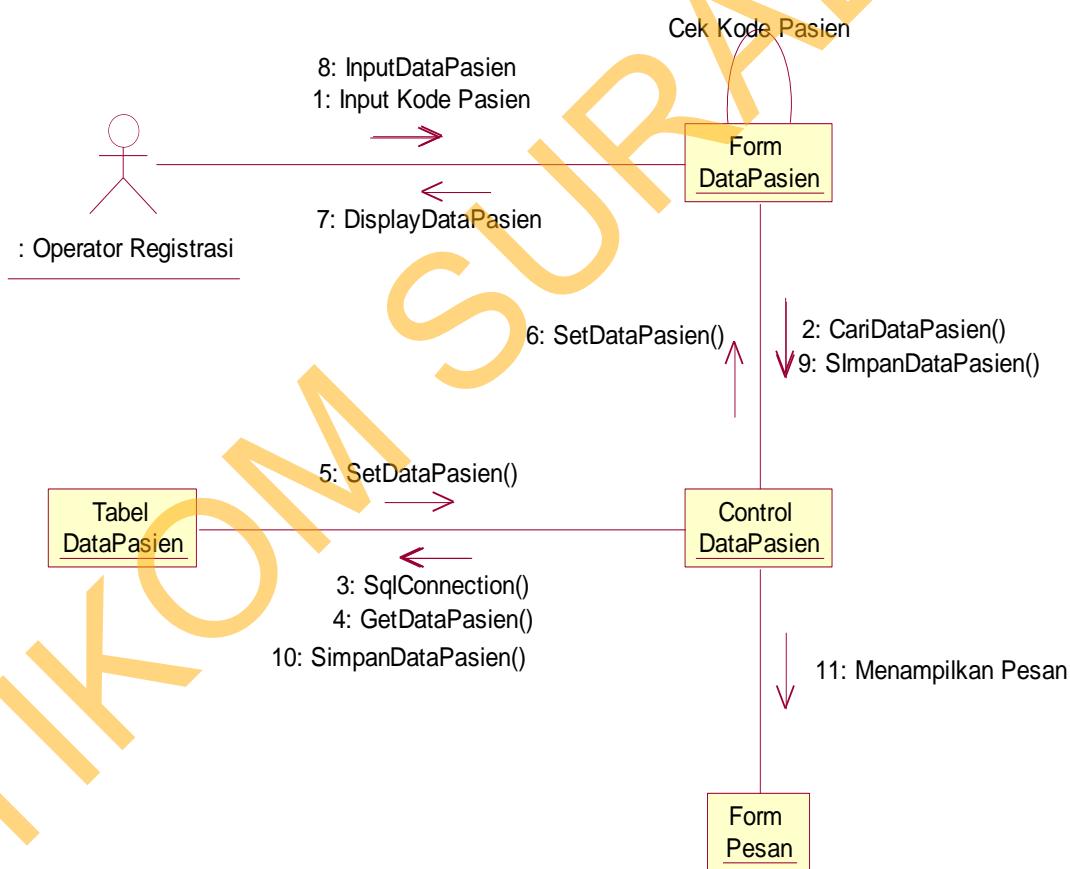


Gambar 3.58 Collaboration Diagram Melakukan Tindakan Medis

E. Collaboration Diagram Melengkapi Identitas Pasien

Pada collaboration diagram melengkapi identitas pasien, objek-objek yang saling berinteraksi antara lain sebagai berikut : FormDataPasien, ControlDataPasien, TabelDataPasien, dan FormPesan. Operator registrasi merupakan aktor yang bersifat sebagai user yang akan berinteraksi dengan objek-objek tersebut.

Pada gambar 3.59 dapat dilihat bagaimana interaksi antara user dengan objek-objek pada collaboration diagram ini.



Gambar 3.59 *Collaboration Diagram Melengkapi Identitas Pasien*

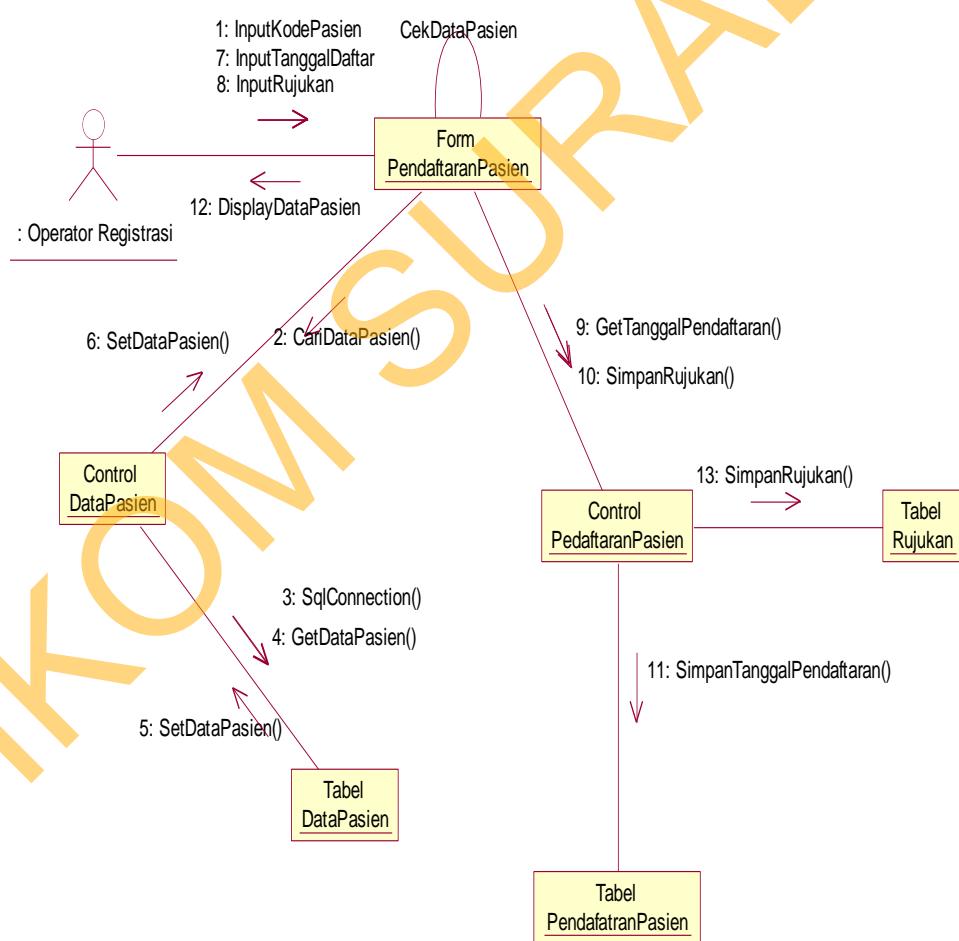
Interaksi diawali dengan user inputkan kode pasien yang akan dilengkapi identitasnya. Kode pasien diinputkan pada FormDataPasien. Setelah user memilih cari, ControlDataPasien akan menjalankan fungsi SqlConnection() untuk membuat koneksi ke dalam tabelDataPasien dan GetDataPasien() untuk mendapatkan data pasien yang dicari berdasarkan kode pasien yang telah diinputkan. Apabila data pasien ditemukan, tabelDataPasien akan menjalankan fungsi SetDataPasien() untuk mengembalikan data yang diminta oleh user. Data pasien akan ditampilkan pada formDataPasien. Untuk melengkapi data yang masih belum lengkap, user memilih menu editDataPasien dan kemudian melengkapi data pasien yang masih kurang lengkap. Apabila data telah seselai dilengkapi, user memilih simpan untuk menyimpan dan memperbarahui data pasien. Control akan menjalankan fungsi SimpanDataPasien() untuk memperbarahui data pasien pada tabelDataPasien dalam database.

F. Collaboration Diagram Memeriksa Rujukan

Pada gambar 3.60 merupakan collaboration diagram untuk proses pemeriksaan rujukan pada saat pendaftaran pasien. Objek-objek yang berinteraksi didalamnya antara lain FormPendaftaranPasien, ControlPendaftaranPasien, ControlDataPasien, TabelPasien, TabelRujukan dan TabelPendaftaranPasien. Operator registrasi merupakan aktor yang berinteraksi dengan objek-objek tersebut sebagai user sistem.

Interaksi diawali dengan user input kode pasien pada FormPendaftaranPasien untuk memeriksa data pasien, kemudian ControlDataPasien akan menjalankan fungsi SqlConnection() untuk membuat

koneksi ke TabelDataPasien dan juga menjalankan fungsi GetDataPasien() untuk memeriksa dan menampilkan data pasien yang dicari. TabelDataPasien akan mengembalikan data pasien yang dicari dengan menjalankan fungsi SetDataPasien(), ControlDataPasien akan melanjutkan menjalankan fungsi SetDataPasien() dan menampilkan data pasien pada FormPendaftaranPasien untuk kemudian menampilkan ke user. Selanjutnya user menambahkan rujukan pasien pada pendaftaran pasien. ControlPendaftaranPasien akan menyimpan pendaftaran ke TabelPendaftaranPasien dan menyimpan rujukan ke TabelRujukan.



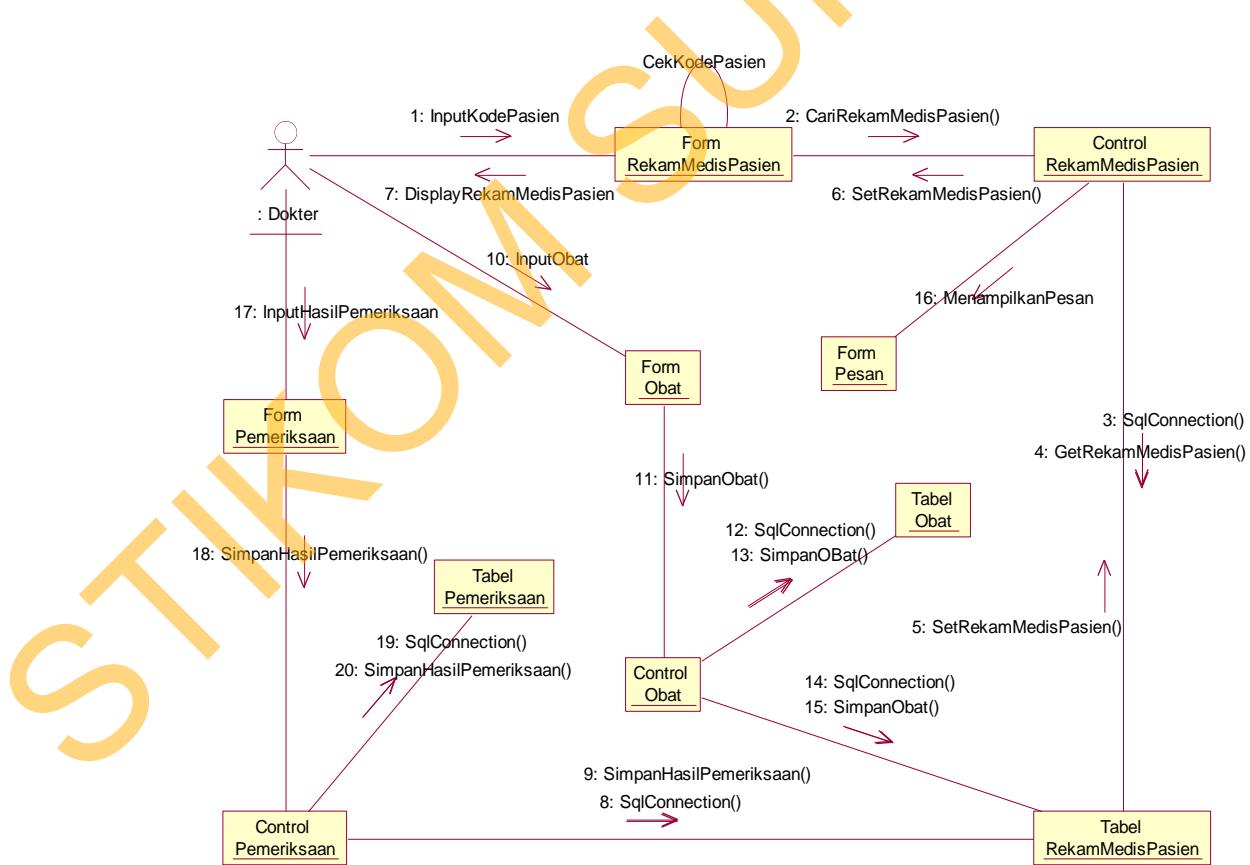
Gambar 3.60 *Collaboration Diagram Memeriksa Rujukan*

G. Collaboration Diagram Menambahkan Data Rekam kesehatan

Dalam collaboration diagram menambahkan data rekam kesehatan, adapun beberapa objek yang saling berinteraksi di dalamnya. Objek-objek tersebut antara lain FormRekamMedisPasien, ControlRekamMedisPasien, FormPesan, FormObat, ControlObat, FormPemeriksaan, ControlPemeriksaan, TabelPemeriksaan, TabelObat, TabelRekamMedisPasien. User yang berinteraksi langsung dengan objek-objek tersebut adalah dokter dan petugas rekam medis, namun disini yang dicantumkan adalah dokter.

Interaksi dimulai saat user input kode pasien pada FormRekamMedisPasien untuk mencari dan menampilkan riwayat rekam medis yang dimiliki oleh pasien. Setelah user memilih cari, ControlRekamMedisPasien akan menjalankan fungsi SqlConnection() untuk koneksi ke TabelRekamMedisPasien dan fungsi GetRekamMedisPasien() untuk menampilkan rekam kesehatan. Data rekam kesehatan yang dicari akan dikembalikan dengan fungsi SetRekamMedisPasien() dan ditampilkan pada FormRekamMedisPasien. Untuk menambahkan pemeriksaan ke dalam data rekam kesehatan, user dapat memilih FormPemeriksaan dan memilih baru. User inputkan hasil pemeriksaan pasien melalui FormPemeriksaan dan ControlPemeriksaan akan membuat koneksi ke TabelPemeriksaan dengan menjalankan fungsi SqlConnection() dan menyimpan hasil pemeriksaan dengan fungsi SimpanPemeriksaan().

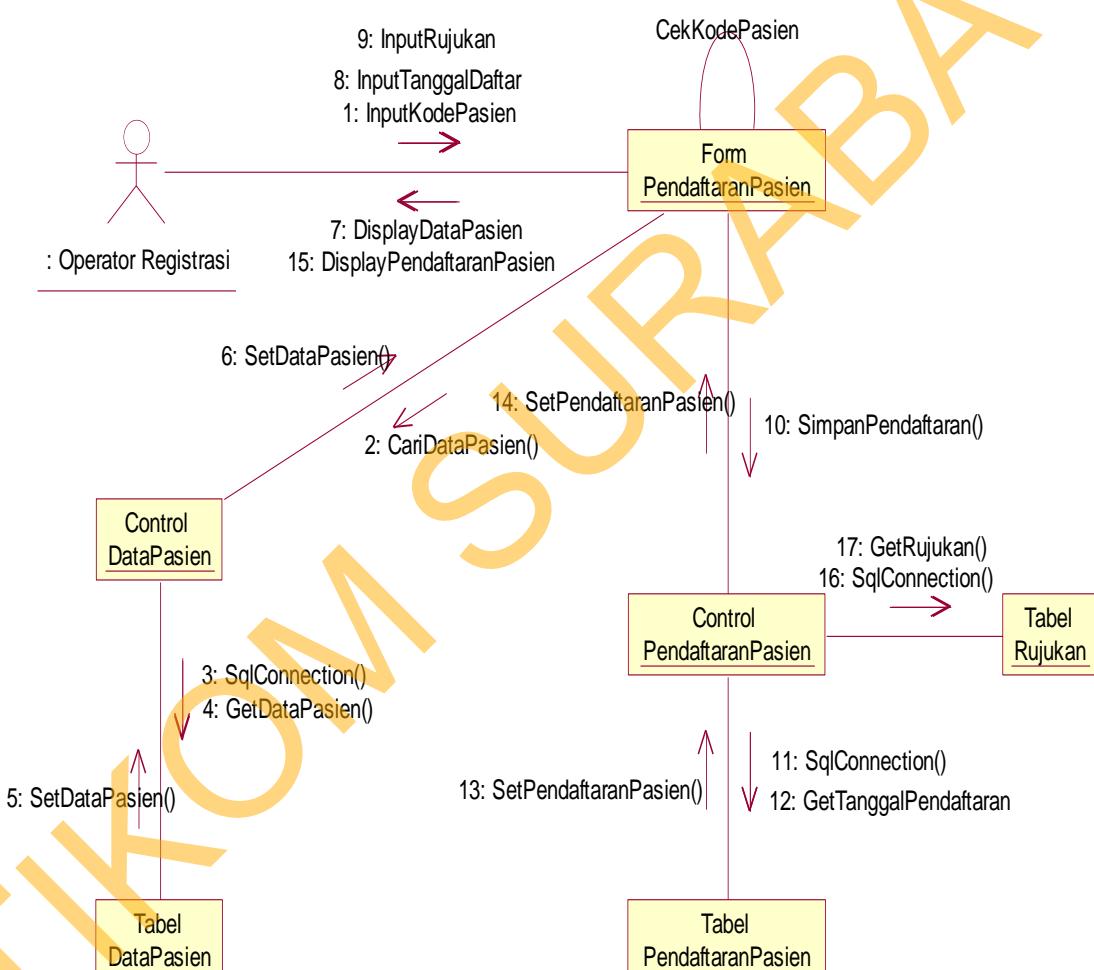
Untuk obat-obat yang diberikan, user dokter akan memilih obat apa saja yang diberikan untuk kemudian disimpan. ControlObat menjalankan fungsi SqlConnection() dan fungsi SimpanObat() untuk menyimpan data obat ke dalam TabelObat. Selain menyimpan pada TabelPemeriksaan dan TabelObat, seluruh pemeriksaan dan obat yang diberikan kepada pasien akan disimpan ke dalam TabelRekamMedisPasien. ControlPemeriksaan dan ControlObat melakukan koneksi dengan SqlConnection() dan menyimpan hasil pemeriksaan beserta obat yang diberikan ke dalam data rekam kesehatan di database terpusat. Segala macam bentuk pesan akan ditampilkan di dalam FormPesan. Untuk lebih jelasnya proses interaksi user dengan objek yang terjadi digambarkan dalam collaboration diagram gambar 3.61.



Gambar 3.61 *Collaboration Diagram* Menambahkan Data Rekam kesehatan

H. Collaboration Diagram Pendaftaraan Pasien

Pada gambar 3.62 akan digambarkan collaboration diagram untuk proses pendaftaran pasien. Pendaftaran pasien mengandung beberapa objek di dalamnya antara lain FormPendaftaranPasien, ControlDataPasien, ControlPendaftaranPasien, TabelDataPasien, TabelPendaftaranPasien, dan TabelRujukan. User yang berinteraksi dengan objek-objek tersebut adalah operator registrasi yaitu petugas yang bertugas menerima pendaftaran pasien.



Gambar 3.62 Collaboration Diagram Pendaftaran Pasien

Interaksi user diawali dengan user yang akan menampilkan data pasien yang melakukan pendaftaran apabila pasien tersebut telah terdaftar pada sistem. User inputkan kode pasien pada FormPendaftaranPasien, selanjutnya ControlDataPasien akan menjalankan fungsi SqlConnection() untuk koneksi ke

TabelDataPasien dan fungsi GetDataPasien untuk mendapatkan data pasien yang dicari. Apabila ditemukan, data pasien akan dikembalikan oleh TabelDataPasien dengan menjalankan fungsi SetDataPasien() dan ControlDataPasien akan menampilkan ke FormPendaftaranPasien. Untuk pasien baru akan langsung didaftarkan dahulu pada FormPendaftaranPasien. Selanjutnya setelah data pasien ditampilkan user akan mencatat tanggal pendaftaran dan mencatat rujukan apabila pasien datang dengan rujukan. ControlPendaftaranPasien akan menyimpan data pendaftaran ke dalam TabelPendaftaranPasien dan rujukan ke TabelRujukan. Tentunya ControlPendaftaranPasien akan menjalankan SqlConnection() ke masing-masing tabel dan kemudian menyimpan data pendaftaran dan rujukan.

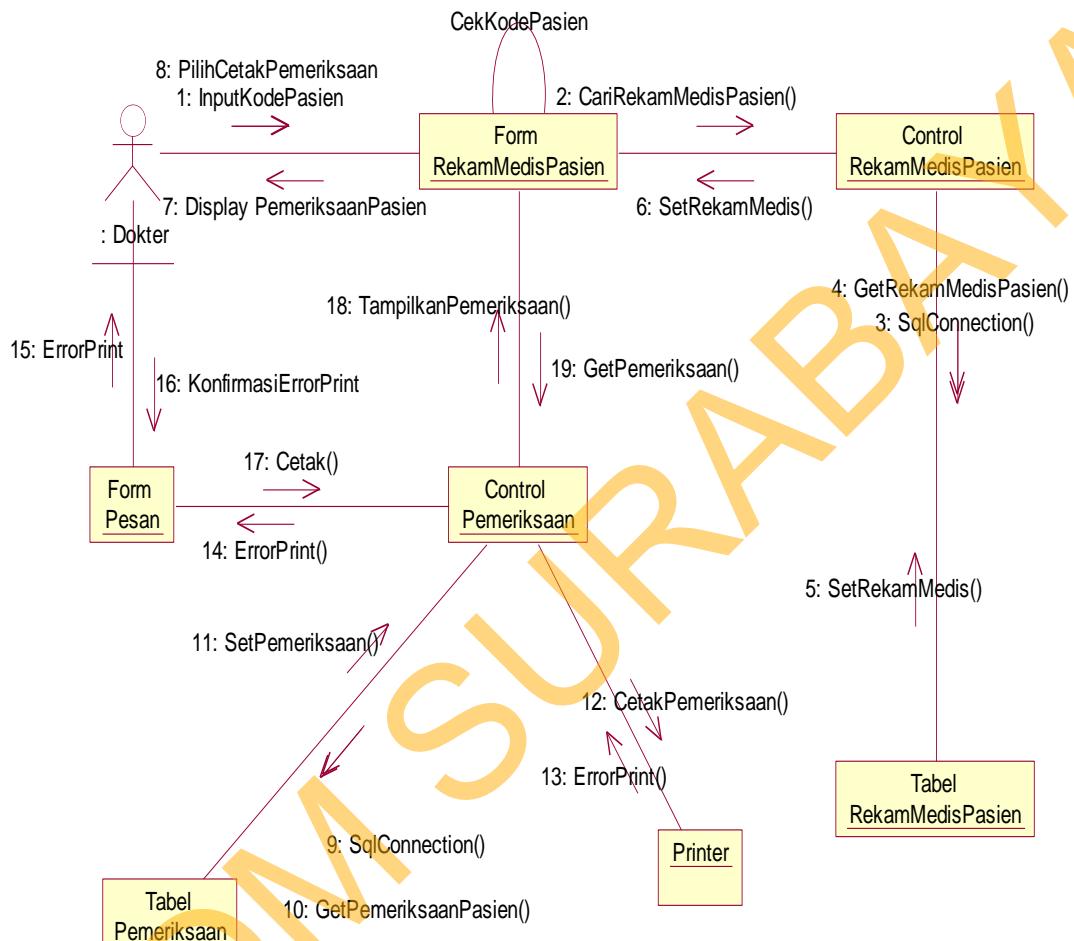
I. Collaboration Diagram Mencetak Hasil Pemeriksaan Pasien

Untuk mencetak pemeriksaan pasien ada beberapa objek yang saling berinteraksi di dalamnya antara lain FormRekamMedisPasien, ControlRekamMedisPasien, ControlPemeriksaan, TabelRekamMedisPasien, TabelPemeriksaan, FormPesanan, dan Printer. User yang berinteraksi dengan objek-objek tersebut adalah dokter atau bagian rekam medis di masing-masing rumah sakit.

Interaksi user diawali dengan user input kode pasien pada FormRekamMedisPasien untuk menampilkan data rekam kesehatan. ControlRekamMedisPasien menjalankan fungsi SqlConnection() dan GetDataRekamMedisPasien. Dari TabelRekamMedisPasien akan mengembalikan data dengan SetRekamMedisPasien(). Data rekam kesehatan yg dicari akan ditampilkan pada FormRekamMedisPasien. User akan memilih data pemeriksaan yang ingin dicetak dan menampilkan detail pemeriksaan dengan memilih tampilan pemeriksaan. ControlPemeriksaan akan menjalankan SqlConnection() dan GetPemeriksaan() untuk mendapatkan data dari TabelPemeriksaan. Hasil pemeriksaan akan dikembalikan dengan fungsi SetPemeriksaan() dan kemudian data pemeriksaan ditampilkan pada FormRekamMedisPasien.

Untuk mencetak, user dapat memilih tombol cetak yang disediakan pada FormRekamMedisPasien. ControlPemeriksaan menjalankan fungsi Cetak() dan mengirimkan pada penggerak printer untuk memulai mencetak. Apabila terjadi

error atau gagal dalam mencetak akan ditampilkan pada FormPesan. Error yang telah dikonfirmasi oleh user, proses mencetak akan dilanjutkan. Untuk lebih jelasnya dapat dilihat pada gambar 3.63.

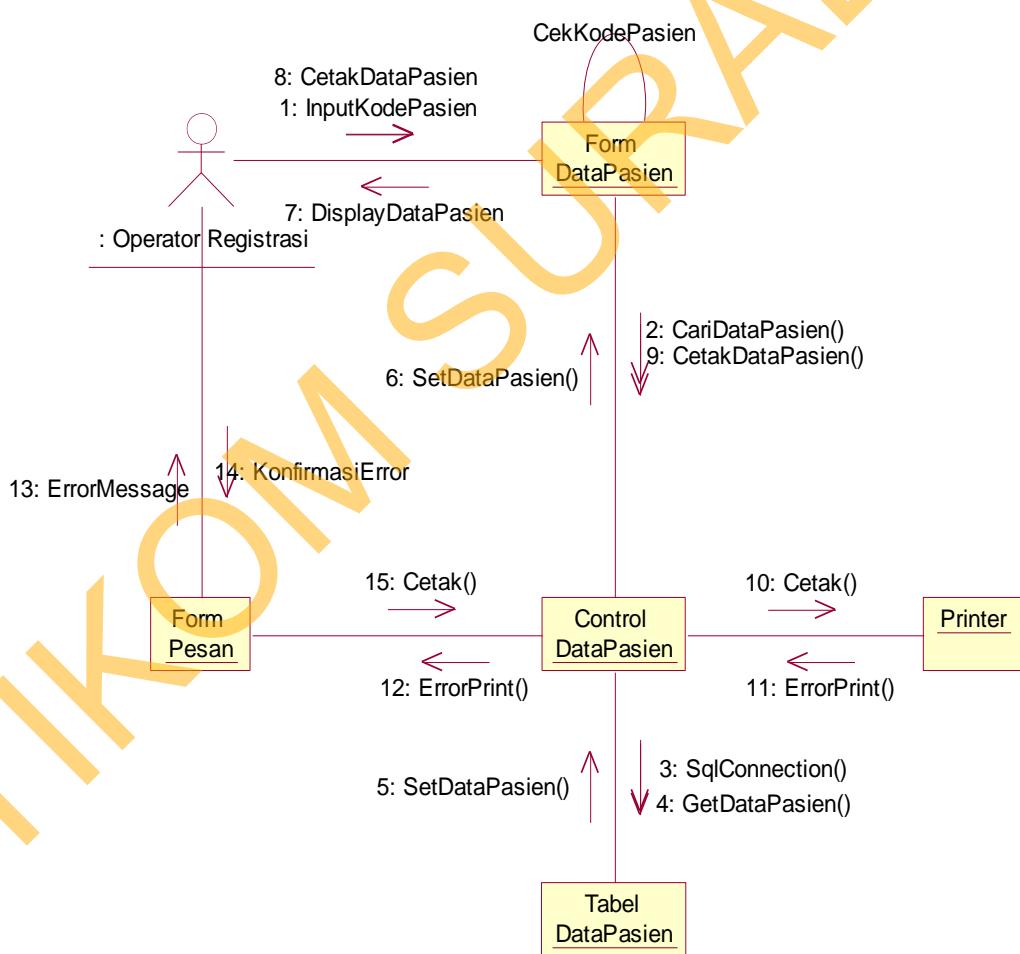


Gambar 3.63 Collaboration Diagram Mencetak Hasil Pemeriksaan Pasien

J. Collaboration Diagram Mencetak Identitas Pasien

Pada collaboration diagram display data pasien menggambarkan interaksi atau hubungan antar objek yang terjadi pada use case display data pasien. Objek yang ada antara lain **FormDataPasien**, **ControlDataPasien**, **TabelDataPasien**, **FormPesan** dan **Printer**. Bagian operator registrasi adalah user yang berinteraksi dengan objek yang ada, diawali dengan user input kode pasien pada

FormDataPasien yang akan dicari datanya kemudian berdasarkan kode pasien akan diperiksa apakah data pasien ada. Melalui ControlDataPasien akan dijalankan fungsi CariDataPasien(), dan dijalankan fungsi SqlConnection() untuk membuka koneksi ke tabel DataPasien serta menjalankan GetDataPasien(). Apabila data ditemukan, tabel DataPasien akan mengembalikan data pasien dengan menjalankan fungsi SetDataPasien(). Apabila data tidak ditemukan, ControlDataPasien akan menampilkan pemberitahuan data pasien yang dicari tidak ditemukan. Dari FormDataPasien menampilkan data pasien kepada user.

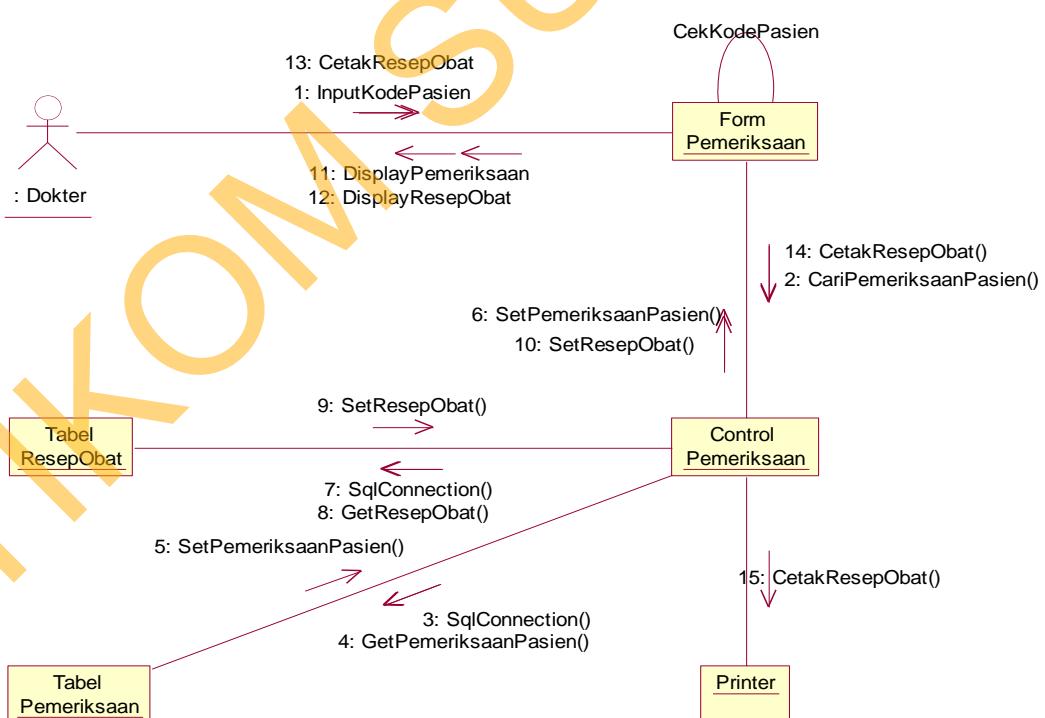


Gambar 3.64 *Collaboration Diagram* Mencetak Identitas Pasien

Seperti yang telah digambarkan pada gambar 3.64, selanjutnya user yang ingin mencetak data pasien tersebut dapat memilih tombol cetak data pasien pada FormDataPasien. ControlDataPasien menjalankan fungsi Cetak() dan mengirimkan pada penggerak printer untuk memulai mencetak. Apabila terjadi error atau gagal dalam mencetak akan ditampilkan pada FormPesan. Error yang telah dikonfirmasi oleh user, proses mencetak akan dilanjutkan.

K. Collaboration Diagram Mencetak Resep Obat

Pada gambar 3.65 akan digambarkan untuk proses mencetak resep obat objek-objek yang terlibat antara lain FormPemeriksaan, ControlPemeriksaan, TabelPemeriksaan, TabelResepObat, dan Printer. Dokter merupakan satu-satunya user yang bisa atau berwenang mencetak resep obat dan berinteraksi dengan



objek-objek tersebut.

Gambar 3.65 *Collaboration Diagram* Mencetak Resep Obat

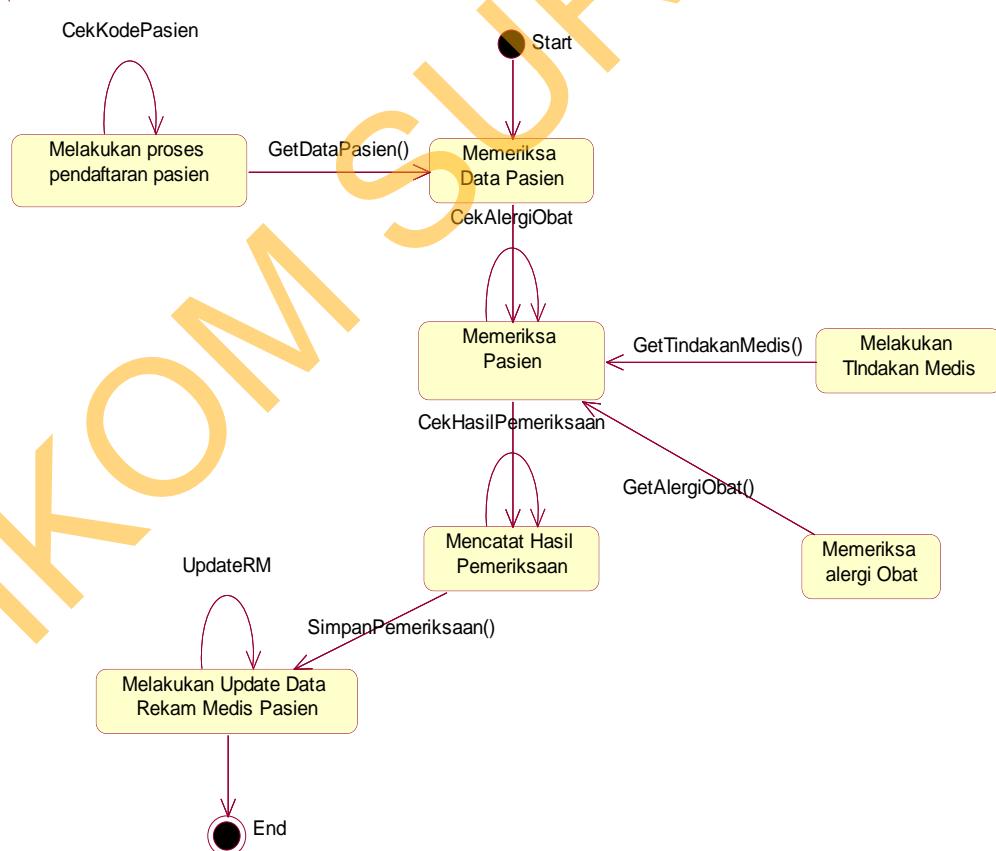
Interaksi diawali user memeriksa pemeriksaan yang dilakukan pasien sehingga munculnya resep obat. User input kode pasien pada FormPemeriksaan dan pilih cari pemeriksaan pasien. ControlPemeriksaan akan menjalankan fungsi SqlConnection() dan fungsi GetPemeriksaan() untuk menampilkan pemeriksaan pasien. Selain itu ControlPemeriksaan akan melakukan koneksi ke TabelResepObat untuk memperoleh data resep obat yang diberikan saat pemeriksaan. Setelah pemeriksaan pasien dan resep obat ditampilkan pada FormPemeriksaanPasien, user memilih cetak resep obat pada FormPemeriksaanPasien. Control Pemeriksaan akan menjalankan fungsi Cetak() ke printer untuk mencetak resep obat.

3.4.8 *Statechart Diagram*

Statechart diagram memperlihatkan berbagai *state* (keadaan sesaat) yang dilalui sebuah objek, dan kejadian-kejadian yang menyebabkan sebuah transisi dari satu state ke state lainnya, dan aksi yang mengakibatkan suatu perubahan state. Statechart diagram juga merupakan teknik yang umum digunakan untuk menggambarkan tingkah laku dinamik dari sebuah class yang ada pada sistem. Dalam pendekatan berorientasi objek, statechart diagram melewati beberapa use case. Dalam desain sistem informasi rekam kesehatan elektronik terpusat ini, statechart diagram akan digunakan untuk menggambarkan beberapa behavior class dan objek yang merupakan inti dari sistem. Statechart diagram yang ada adalah sebagai berikut.

A. Statechart Diagram Class Entity Pasien

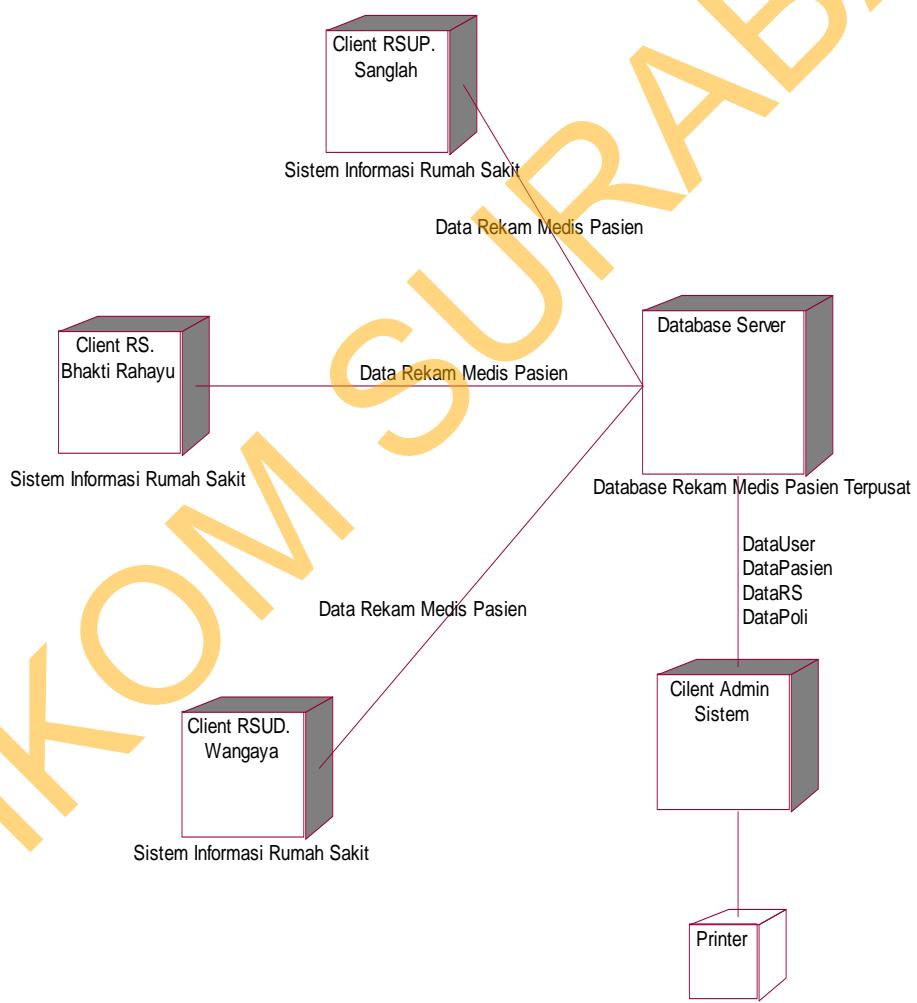
Pada gambar 3.66 statechart diagram pasien akan memperlihatkan tingkah laku dari objek pasien. Objek pasien yaitu data pasien yang nantinya menjadi salah satu objek yang ada dalam sistem yaitu objek pasien memiliki behavior yang sangat dinamis. Beberapa state yang akan menggunakan objek pasien antara lain melakukan proses pendaftaran pasien, memeriksa pasien, melakukan tindakan medis, memeriksa alergi obat, mencatat hasil pemeriksaan, dan melakukan update data rekam kesehatan. Untuk proses memeriksa pasien, memeriksa alergi obat dan melakukan update data rekam kesehatan, data pasien secara nyata yang akan digunakan.



Gambar 3.66 Statechart Diagram Class entity Pasien

3.4.9 Deployment Diagram

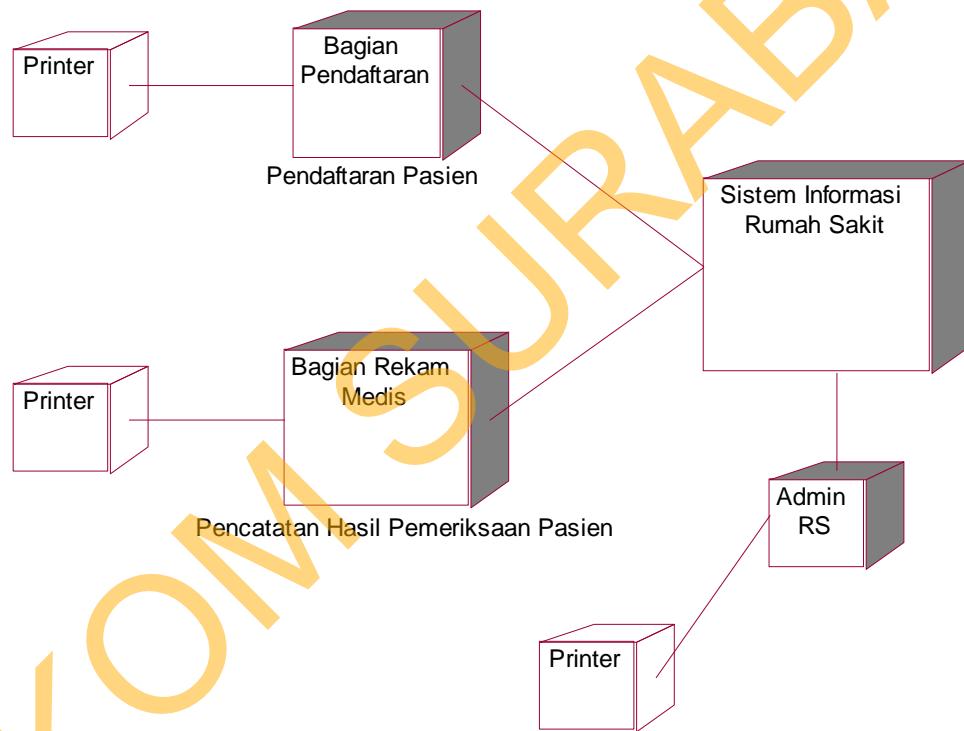
Deploymet diagram menunjukkan tata letak sebuah sistem secara fisik, menampakkan bagian-bagian software yang berjalan pada bagian-bagian hardware. Bagian utama hardware/perangkat keras adalah node yaitu nama umum untuk semua jenis kumputasi. Ada 2 tipe node yaitu processor yang bisa mengeksekusi sebuah component dan device merupakan hardware seperti printer dan monitor. Setiap sistem hanya memiliki satu deployment diagram.



Gambar 3.67 Deployment Diagram Sistem Informasi Rekam kesehatan Elektronik Terpusat

Pada gambar 3.67 digambarkan terdapat beberapa processor dan device serta koneksi yang menghubungkan antar processor yang satu dengan yang lain. Untuk pengelolaan database server, dikelola oleh administrator sistem. Untuk aplikasi database menggunakan SQL Server 2005. Untuk user lain yaitu di rumah sakit menggunakan aplikasi client yaitu Sistem Informasi Rumah Sakit.

Untuk deployment diagram Sistem Informasi Rumah Sakit akan digambarkan pada gambar 3.68.



Gambar 3.68 *Deployment Diagram* Sistem Informasi Rumah Sakit

Pada gambar 3.68 digambarkan terdapat beberapa processor dan device serta koneksi yang menghubungkan antar processor yang satu dengan yang lain. Untuk pengelolaan sistem informasi rumah sakit dikelola oleh administrator rumah sakit. Untuk aplikasi menggunakan aplikasi client dan database menggunakan SQL Server 2005. Terdapat beberapa device yang akan digunakan

yaitu printer yang akan ditempatkan pada masing-masing bagian yang akan digunakan sesuai dengan kebutuhannya.

3.5 Pemetaan Objek ke Tabel Database

Pemetaan class maupun objek-objek yang ada dalam tahap analisis dan desain sistem ke dalam class entity merupakan tahap untuk mengenal secara dini class atau objek mana yang nantinya akan menjadi tabel penyimpanan data dalam database. Selain untuk desain sistem, class biasanya digunakan untuk mendefinisikan objek-objek bisnis. Class seperti ini biasanya mendefinisikan model database dari suatu aplikasi dan class entity juga akan menjelaskan esensi dari sebuah sistem informasi itu sendiri.

Dalam desain sistem informasi rekam kesehatan elektronik terpusat ini, tahap analisis dari desain berhasil menemukan beberapa class maupun objek yang akan dijadikan tabel-tabel penyimpanan data dalam database. Untuk menjelaskan hal tersebut akan digunakan penggambaran struktur tabel. Struktur tabel-tabel yang ada dalam database adalah sebagai berikut:

1. Nama Tabel : Pasien
Fungsi : Menyimpan data pasien
Class : Pasien

Tabel 3.3 Struktur Tabel Pasien

Field	Tipe	Ukuran	Keterangan
ID_Pasien	Integer	-	Primary Key
NO_Identitas	Integer	-	-
ID_Desa	Integer	-	Foreign Key
ID_Kecamatan	Integer	-	Foreign Key
ID_Kabupaten	Integer	-	Foreign Key
ID_Provinsi	Integer	-	Foreign Key
ID_Pekerjaan	Integer	-	Foreign Key
ID_Agama	Integer	-	Foreign Key
ID_Penanggung_Jawab	Integer	-	Foreign Key
Nama_Pasien	Varchar	50	-
Alamat	Varchar	50	-
Telepon	Varchar	13	-
Jenis_kelamin	Char	1	-
Tempat_lahir	Varchar	50	-
Tanggal_Lahir	Date	-	-
Golongan_Darah	Char	2	-
Kewarganegaraan	Varchar	50	-
Pendidikan_Terakhir	Varchar	50	-
Status_Pernikahan	Varchar	50	-

2. Nama Tabel : Penanggung_Jawab

Fungsi : Menyimpan data penanggung jawab pasien

Class : PenanggungJawab

Tabel 3.4 Struktur Tabel Penanggung Jawab

Field	Tipe	Ukuran	Keterangan
ID_Penanggung_Jawab	Integer	-	Primary Key
ID_Desa	Integer	-	Foreign Key
ID_Kecamatan	Integer	-	Foreign Key
ID_Kabupaten	Integer	-	Foreign Key
ID_Provinsi	Integer	-	Foreign Key
ID_Pekerjaan	Integer	-	Foreign Key
Nama_Penanggung_Jawab	Varchar	50	-
Alamat	Varchar	50	-
Telepon	Varchar	13	-
Hubungan_Keluarga	Varchar	50	-

3. Nama Tabel : Desa
 Fungsi : Menyimpan Data Identitas Desa
 Class : Desa

Tabel 3.5 Struktur Tabel Desa

Field	Tipe	Ukuran	Keterangan
ID_Des a	Integer	-	Primary Key
Nama_Des a	Varchar	50	-
ID_Kecamatan	Integer	-	Foreign Key

4. Nama Tabel : Kecamatan
 Fungsi : Menyimpan Data Identitas Kecamatan
 Class : Kecamatan

Tabel 3.6 Struktur Tabel Kecamatan

Field	Tipe	Ukuran	Keterangan
ID_Kecamatan	Integer	-	Primary Key
Nama_Kecamatan	Varchar	50	-
ID_Kabupaten	Integer	-	Foreign Key

5. Nama Tabel : Kabupaten
 Fungsi : Menyimpan Data Identitas Kabupaten
 Class : Kabupaten

Tabel 3.7 Struktur Tabel Kabupaten

Field	Tipe	Ukuran	Keterangan
ID_Kabupaten	Integer	-	Primary Key
Nama_Kabupaten	Varchar	50	-
ID_Provinsi	Integer	-	Foreign Key

6. Nama Tabel : Provinsi
 Fungsi : Menyimpan Data Identitas Provinsi
 Class : Provinsi

Tabel 3.8 Struktur Tabel Provinsi

Field	Tipe	Ukuran	Keterangan
ID_Provinsi	Integer	-	Primary Key
Nama_Provinsi	Varchar	50	-

7. Nama Tabel : Pekerjaan
 Fungsi : Menyimpan jenis-jenis pekerjaan
 Class : Pekerjaan

Tabel 3.9 Struktur Tabel Pekerjaan

Field	Tipe	Ukuran	Keterangan
ID_Pekerjaan	Integer	-	Primary Key
Nama_Pekerjaan	Varchar	50	-

8. Nama Tabel : Agama
 Fungsi : Menyimpan jenis-jenis agama
 Class : Agama

Tabel 3.10 Struktur Tabel Agama

Field	Tipe	Ukuran	Keterangan
ID_Desa	Integer	-	Primary Key
NamaDesa	Varchar	50	-

9. Nama Tabel : Registrasi_Pasien
 Fungsi : Menyimpan registrasi pasien
 Class : PendaftaranPasien

Tabel 3.11 Struktur Tabel Registrasi Pasien

Field	Tipe	Ukuran	Keterangan
ID_Registrasi_Pasien	Integer	-	Primary Key
ID_Pasien	Integer	-	Foreign Key
ID_Rujukan	Integer	-	Foreign Key
Tanggal_Pendaftaran	Date	-	-
Keluhan	Varchar	50	-
Status	Varchar	50	-

10. Nama Tabel : Registrasi Poliklinik

Fungsi : Meyimpan Transaksi Registrasi Pasien

Class : -

Field	Tipe	Ukuran	Keterangan
ID_Registrasi_Poliklinik	Integer	-	Primary Key
ID_Poliklinik	Integer	-	Foreign Key
No_Antrean	Integer	-	-
Jam_Masuk_Poliklinik	time	-	-

11. Nama Tabel : Rujukan

Fungsi : Menyimpan rujukan pasien

Class : Rujukan

Tabel 3.12 Struktur Tabel Rujukan

Field	Tipe	Ukuran	Keterangan
ID_Rujukan	Integer	-	Primary Key
ID_Pasien	Integer	-	Foreign Key
ID_Poliklinik	Integer	-	Foreign Key
ID_RS	Integer	-	Foreign Key
Tanggal_Rujukan	Date	-	-
Status	Varchar	50	-

12. Nama Tabel : Rumah_Sakit

Fungsi : Menyimpan data rumah sakit

Class : RumahSakit

Tabel 3.13 Struktur Tabel Rumah Sakit

Field	Tipe	Ukuran	Keterangan
ID_RS	Integer	-	Primary Key
Nama_RS	Varchar	50	-
Alamat	Varchar	50	-
Telepon	Varchar	13	-
ID_Desa	Varchar	50	Foreign Key
ID_Kecamatan	Varchar	50	Foreign Key
ID_Provinsi	Integer	-	Foreign Key
ID_Kabupaten	Integer	-	Foreign Key

13. Nama Tabel : Poliklinik

Fungsi : Menyimpan data poliklinik

Class : Poliklinik

Tabel 3.14 Struktur Tabel Poliklinik

Field	Tipe	Ukuran	Keterangan
ID_Poliklinik	Integer	-	Primary Key
ID_RS	Integer	-	Foreign Key
Nama_Poliklinik	Varchar	50	-

14. Nama Tabel : Dokter

Fungsi : Menyimpan Data Dokter

Class : Dokter

Tabel 3.15 Struktur Tabel Dokter

Field	Tipe	Ukuran	Keterangan
ID_Dokter	Integer	-	Primary Key
ID_Poliklinik	Integer	-	Foreign Key
ID_RS	Integer	-	Foreign Key
Nama_Dokter	Varchar	50	-
Alamat	Varchar	50	-
Spesialisasi	Varchar	50	-
Jenis_Kelamin	Char	1	-
Telepon	Varchar	13	-

15. Nama Tabel : Resep_Obat

Fungsi : Menyimpan Data Resep Obat

Class : ResepObat

Tabel 3.16 Struktur Tabel Resep Obat

Field	Tipe	Ukuran	Keterangan
ID_Resep_Obat	Integer	-	Primary Key
ID_Pemeriksaan	Integer	-	Foreign Key
Tanggal_Resep	Date	-	-
ID_Obat	Integer	-	Foreign Key
Dosis	Varchar	50	-

16. Nama Tabel : Obat

Fungsi : Menyimpan Data Obat

Class : Obat

Tabel 3.17 Struktur Tabel Obat

Field	Tipe	Ukuran	Keterangan
ID_Obat	Integer	-	Primary Key
Nama_Obat	Varchar	50	-

17. Nama Tabel : Pemeriksaan

Fungsi : Menyimpan Pemeriksaan Pasien

Class : Pemeriksaan

Tabel 3.18 Struktur Tabel Pemeriksaan

Field	Tipe	Ukuran	Keterangan
ID_Pemeriksaan	Integer	-	Primary Key
ID_Registrasi_Pasien	Integer	-	Foreign Key
ID_Resep_Obat	Integer	-	Foreign Key
ID_Dokter	Integer	-	Foreign Key
Anamnesa	Varchar	50	-
Diagnosa	Varchar	50	-
Tindakan_Medis	Varchar	50	-
Alergi_Obat	Varchar	50	-
Nasehat	Varchar	50	-
ID_RS	Integer	-	Foreign Key
ID_Poliklinik	Integer	-	Foreign Key
ID_ICD10	Integer	-	Foreign Key
ID_ICDCM	Integer	-	Foreign Key
ID_Lab	Integer	-	Foreign Key
ID_Rad	Integer	-	Foreign Key

18. Nama Tabel : Laboratorium
 Fungsi : Menyimpan Data Laboratorium
 Class : -

Tabel 3.19 Struktur Tabel Laboratorium

Field	Tipe	Ukuran	Keterangan
ID_Lab	Integer	-	Primary Key
Nama_Lab	Varchar	50	-

19. Nama Tabel : Radiologi
 Fungsi : Menyimpan Data Radiologi
 Class : -

Tabel 3.20 Struktur Tabel Radiologi

Field	Tipe	Ukuran	Keterangan
ID_Rad	Integer	-	Primary Key
Nama_Rad	Varchar	50	-

20. Nama Tabel : Penyakit_ICD_10
 Fungsi : Menyimpan Data Penyakit
 Class : -

Tabel 3.21 Struktur Tabel Penyakit_ICD_10

Field	Tipe	Ukuran	Keterangan
ID_ICD_10	Integer	-	Primary Key
Nama_Penyakit	Varchar	50	-

21. Nama Tabel : Penyakit_ICD_9
 Fungsi : Menyimpan Data Tindakan dan Prosedur
 Class : -

Tabel 3.22 Struktur Tabel Tindakan_ICD_9

Field	Tipe	Ukuran	Keterangan
ID_ICD_9	Integer	-	Primary Key
Prosedur_Tindakan	Varchar	50	-

22. Nama Tabel : Bekerja

Fungsi : Menyimpan data dokter yang bekerja pada rumah sakit

Class : -

Tabel 3.23 Struktur Tabel Bekerja

Field	Tipe	Ukuran	Keterangan
ID_Bekerja	Integer	-	Primary Key
ID_RS	Integer	-	Foreign Key
ID_Dokter	Integer	-	Foreign Key
Tanggal_Mulai_Bekerja	date	-	-
Jadwal_Kerja	Time	-	-

23. Nama Tabel : Rekam_Medis_Pasien

Fungsi : Menyimpan Data Rekam kesehatan

Class : RekamMedisPasien

Tabel 3.24 Struktur Tabel Rekam Medis

Field	Tipe	Ukuran	Keterangan
ID_Rekam_Medis	Integer	-	Primary Key
ID_Pasien	Integer	-	Foreign Key
ID_Pemeriksaan	Integer	-	Foreign Key

Setelah pembuatan struktur tabel-tabel yang akan digunakan dalam aplikasi, selanjutnya tabel akan diimplementasikan ke dalam suatu database.

Dalam desain sistem informasi ini yang didapat akan diimplementasikan ke database architect menggunakan power designer. Database yang sudah dibuat akan tampak seperti pada gambar 3.69 dan gambar 3.70 yang menggambarkan diagram database.

A. Pengkodean Yang Digunakan

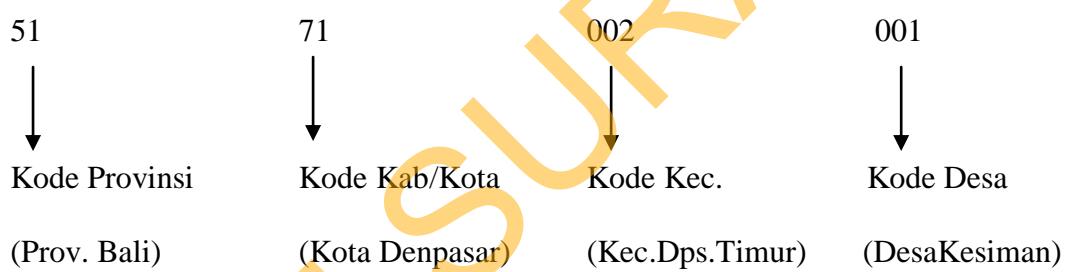
Pengkodean yang digunakan dalam rancangan sistem informasi ini sesuai dengan Keputusan Menteri Kesehatan RI nomor : 844/Menkes/SK/X/2006 tentang penetapan standar kode data bidang kesehatan. Adapun pengkodean sesuai standar nasional yang digunakan antara lain :

1. Kode Wilayah

Kode wilayah terdiri dari 10 digit yaitu Kode Provinsi 2 digit, Kode Kabupaten/Kota 2 digit, Kode Kecamatan 3 digit, dan Kode Desa 3 digit.

Struktur Kode :

Contoh untuk Desa Kesiman :



2. Kode Rumah Sakit

Kode rumah sakit yang telah dikeluarkan Ditjen Pelayanan Medik terdiri dari 7 (tujuh) digit dengan penjelasan sebagai berikut :

- 2 (dua) digit pertama menunjukkan angka kode Provinsi
- 2 (dua) digit kedua menunjukkan angka kode Kabupaten/Kota
- 3 (tiga) digit terakhir menunjukkan nomor urut rumah sakit sesuai dengan urutan pengajuan penetapan oleh Direktorat Jendral Pelayanan Medik c.q Sekretariat Direktorat Jendral Pelayanan Medik.

Contoh Pengkodean Untuk RSUP. Sanglah (Provinsi Bali)

51	71	001
↓	↓	↓
Kode Provinsi Bali	Kode Kab/Kota Denpasar	No. Urut Rumah Sakit

3. Kode Obat

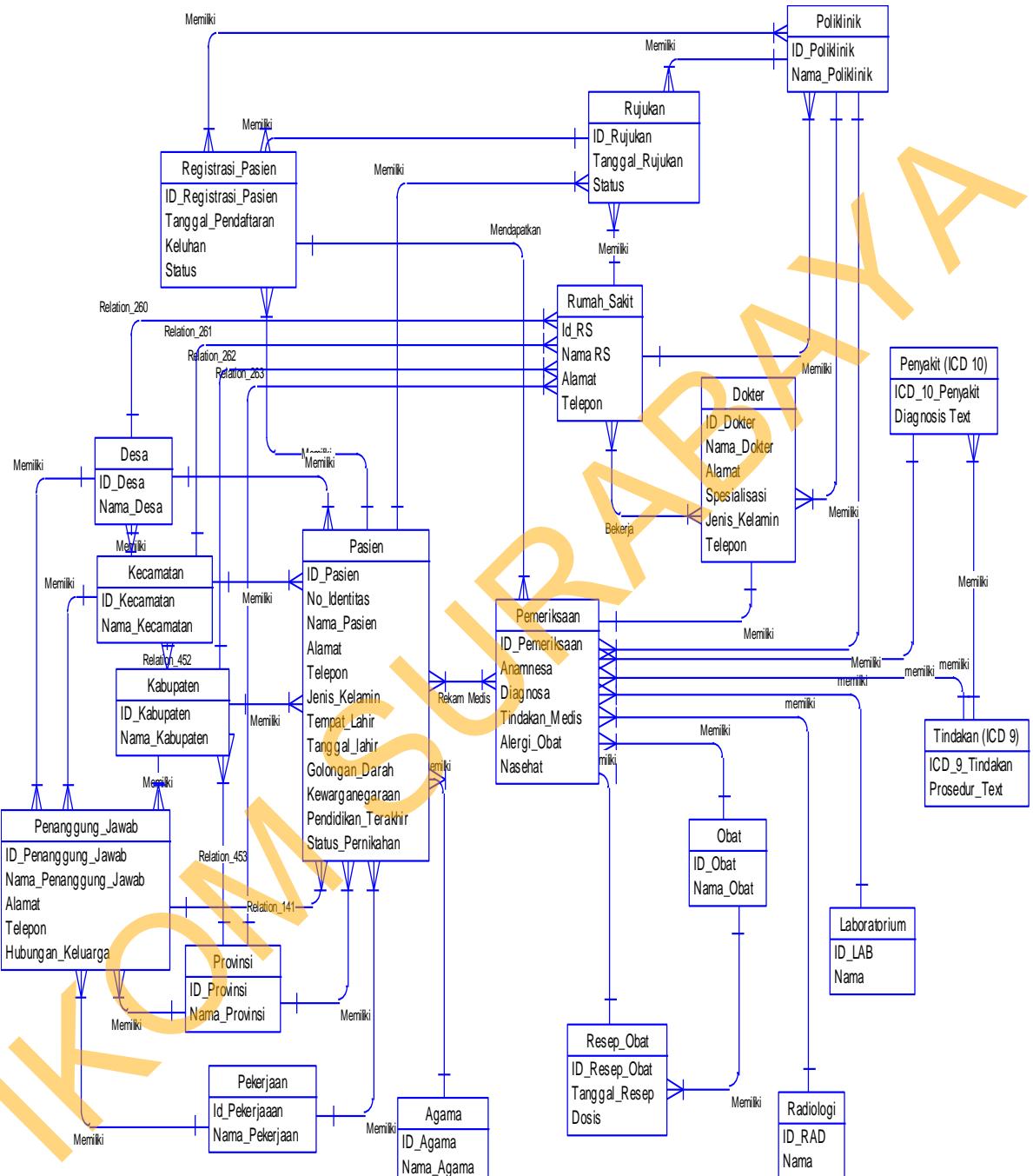
Kode obat yang digunakan terdiri dari 12 digit yaitu sebagai berikut :

- a) 6 (enam) digit pertama merupakan Kode Farmakoterapi
- b) 4 (empat) digit kedua merupakan Kode Nama Obat
- c) 2 (dua) digit terakhir merupakan Kekuatan Sediaan

Contoh Penerapannya pada obat Abamox Kapsul 250 mg yaitu :

01.01.04	0158	02
↓	↓	↓
Kode Farmakoterapi (Antiinfeksi/antibiotic)	Kode Nama Obat (Abamox)	Kode Kekuatan Sediaan (Kapsul 250mg)

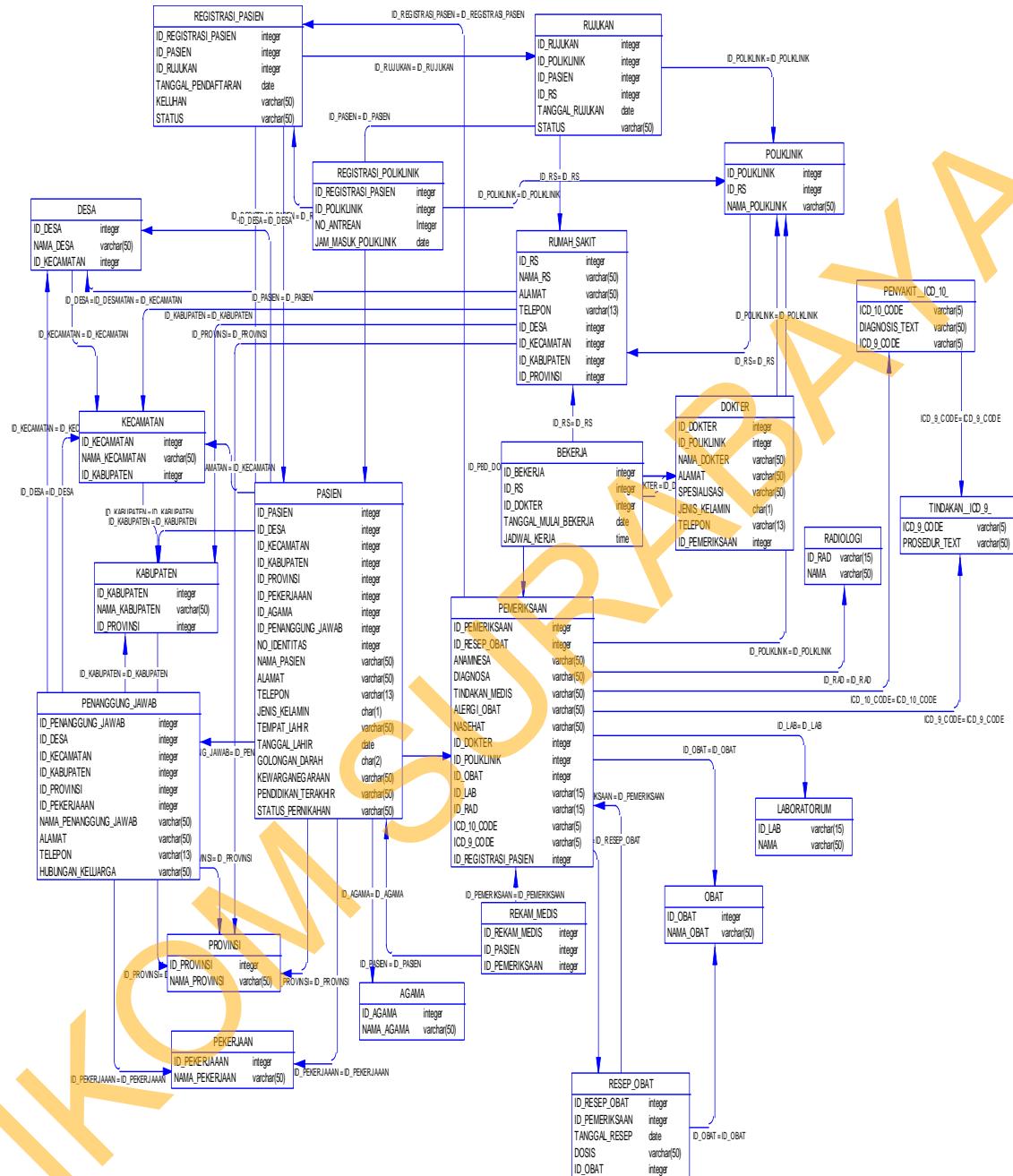
B. Conceptual Data Model



Gambar 3.69 *Conceptual Data Model* Database Rekam kesehatan Elektronik

Terpusat

C. Physical Data Model



Gambar 3.70 *Physical Data Model* Database Rekam kesehatan Elektronik

Terpusat

Dari gambar 3.69 dan gambar 3.70 terlihat tabel-tabel dan relasi antar tabel yang ada dalam database. Tabel yang ada antara lain : Pasien, Penanggung jawab, Desa, Kecamatan, Kabupaten, Provinsi, Pekerjaan, Agama,

Pendaftaran_Pasien, Rujukan, Rumah_Sakit, Poliklinik, Dokter, Obat, Resep_Obat, Pemeriksaan dan Rekam_Medis_Pasien. Masing-masing tabel juga memiliki field sebagai primary key. Dari struktur tabel tersebut akan diimplementasikan ke dalam tabel fisik dalam database. Tabel-tabel tersebut dapat dilihat pada gambar berikut :

1. Tabel Desa

ID Desa	Nama Desa	ID Kecamatan
5171032006	Dauh Puri Kangin	517103
5171032003	Dauh Puri Kauh	517103
5171032004	Dauh Puri Klod	517103
5171032011	Padangsambian Kaja	517103
5171032001	Padangsambian klod	517103
5171032002	Pemecutan Klod	517103
5171032008	Tegal Harum	517103
5171032009	Tegal Kerta	517103

Gambar 3.71 Contoh Tabel Desa

Gambar 3.70 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel desa yaitu ID_Des'a, ID_Kecamatan dan Nama_Des'a. Dalam tabel ini ID_Des'a merupakan primary key dan ID_Kecamatan merupakan foreign key.

2. Tabel Kecamatan

ID Kecamatan	Nama Kecamatan	ID Kabupaten/Kota
517103	Denpasar Barat	5171
517104	Denpasar Utara	5171
517102	Denpasar Timur	5171
517101	Denpasar Selatan	5171

Gambar 3.72 Contoh Tabel Kecamatan

Gambar 3.71 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel kecamatan yaitu ID_Kecamatan, ID_Kabupaten/Kota dan Nama_Kecamatan. Dalam tabel ini ID_Kecamatan merupakan primary key dan ID_Kabupaten/Kota merupakan foreign key.

3. Tabel Kabupaten

ID Kabupaten/Kota	Nama Kabupaten/Kota	ID Provinsi
5171	Denpasar	51
5103	Badung	51
5102	Tabanan	51
5101	Jembrana	51
5105	Kelungkung	51
5107	Karangasem	51
5106	Bangli	51
5104	Gianyar	51
5108	Singaraja	51

Gambar 3.73 Contoh Tabel Kabupaten/Kota

Gambar 3.72 memperlihatkan data-data yang tersimpan dalam setiap field yang ada pada tabel kabupaten/kota yaitu ID_Kabupaten/Kota, ID_Provinsi dan Nama_Kabupaten/Kota. Dalam tabel ini ID_Kabupaten/Kota merupakan primary key dan ID_Provinsi merupakan foreign key.

4. Tabel Provinsi

ID Provinsi	Nama Provinsi
31	DKI. Jakarta
32	Jawa Barat
33	Jawa Tengah
36	Banten
34	Yogyakarta
35	Jawa Timur
51	Bali
52	Nusa Tenggara Barat
53	Nusa Tenggara Timur

Gambar 3.74 Contoh Tabel Provinsi

Gambar 3.74 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel provinsi yaitu ID_Provinsi dan Nama_Provinsi. Dalam tabel ini ID_Provinsi merupakan primary key.

5. Tabel Pasien

ID_Pasien	NIK	Mc.Nama	Provinsi	Desa	Dicantik	Dikukuh	D_Pasien	D_Parjain	D_Sama	Nama_Pasien	Alamat	Telepon	Isi_Kamar	Tgl	Gol.Darah	Kewarganegaraan	Pendidikan Terahir	Status_Pernikahan			
00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000	00000000000000000000000000000000

Gambar 3.75 Contoh Tabel Pasien

Gambar 3.75 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel pasien yaitu ID_Pasien, ID_RM, No_Identitas, ID_Penanggung_Jawab, ID_Desa, ID_Kecamatan, ID_Kabupaten, ID_Provinsi, ID_Pekerjaan, ID_Agama, Nama_Pasien, Alamat, Telepon, Jenis_Kelamin, Tempat/TanggalLahir, Golongan_Darah, Kewarganegaraan, Pendidikan_Terakhir dan Status_Pernikahan. Dalam tabel ini ID_Pasien merupakan primary key dan ID_RM sebagai foreign key.

6. Tabel Penanggung Jawab

Gambar 3.76 Contoh Tabel Penanggung Jawab

Gambar 3.76 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel penanggung jawab yaitu ID_Penanggung_Jawab, ID_Desa, ID_Kecamatan, ID_Kabupaten, ID_Provinsi, ID_Pekerjaan, Nama_Penanggung_Jawab, Alamat, Telepon, dan Hubungan_Keluarga. Dalam tabel ini ID_Pasien merupakan primary key.

7. Tabel Pekerjaan

Gambar 3.77 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel pekerjaan yaitu ID_Provinsi dan Nama_Provinsi. Dalam tabel ini ID_Provinsi merupakan primary key.

ID Pekerjaan	Nama Pekerjaan
01	PNS
02	Wiraswasta
03	Dokter
04	Petani
05	Pelajar
06	Mahasiswa
07	Pedagang
08	Guru
09	Sopir

Gambar 3.77 Contoh Tabel Pekerjaan

8. Tabel Agama

ID Agama	Nama Agama
01	Islam
02	Hindu
03	Kristen Katolik
04	Kristen Protestan
05	Buddha
06	Kong Hu Cu

Gambar 3.78 Contoh Tabel Pekerjaan

Gambar 3.77 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel agama yaitu ID_Agama dan Nama_Agama. Dalam tabel ini ID_Agama merupakan primary key.

9. Tabel Pendaftaran_Pasien

ID Pendaftaran Pasien	ID Pasien	ID Pemeriksaan	ID Rujukan	Tanggal_Pendaftaran	Keluhan	Status
0102091200001	520209120680001	01030001	01030001	20/08/2012	Demam	Gawat

Gambar 3.79 Contoh Tabel Pendaftaran Pasien

Gambar 3.79 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel pendaftaran pasien yaitu ID_Pendaftaran_Pasien, ID_Pasien, ID_Pemeriksaan, ID_Rujukan, Tanggal_Pendaftaran, Keluhan dan Status. Dalam tabel ini ID_Pendaftaran merupakan primary key.

10. Tabel Rujukan

Gambar 3.80 Contoh Tabel Rujukan

Gambar 3.80 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel rujukan yaitu ID_Rujukan, ID_Pasien, ID_Pendaftaran_Pasien, ID_RS, ID_Poliklinik, Tanggal_Rujukan dan Status. Dalam tabel ini ID_Pendaftaran merupakan primary key.

11. Tabel Rumah Sakit

Gambar 3.81 Contoh Tabel Rumah Sakit

Gambar 3.81 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel rumah sakit yaitu ID_Rumah_Sakit, Nama_Rumah_Sakit, Alamat, dan Telepon. Dalam tabel ini ID_Rumah_Sakit merupakan primary key.

12. Tabel Poliklinik

ID Poliklinik	Nama Rumah Sakit	Nama Poliklinik
5171001001	RSUP. SANGLAH	Bedah Umum
5171001002	RSUP. SANGLAH	Bedah Saraf
5171001003	RSUP. SANGLAH	Kulit dan Kelamin
5171001004	RSUP. SANGLAH	THT
5171001005	RSUP. SANGLAH	Anak
5171001006	RSUP. SANGLAH	Penyakit Dalam
5171001007	RSUP. SANGLAH	Urologi
5171001008	RSUP. SANGLAH	Mata
5171001009	RSUP. SANGLAH	Dokter Umum

Gambar 3.82 Contoh Tabel Poliklinik

Gambar 3.82 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel poliklinik yaitu ID_Poliklinik, Nama_Rumah_Sakit, dan Nama_Poliklinik. Dalam tabel ini ID_Poliklinik merupakan primary key.

13. Tabel Dokter

Nama Dokter	Nama Dokter	Spesialisasi	Rumah Sakit	Poliklinik	Alamat	Jns. Kelamin	Telepon
51710010010001	Dr. Anak Agung Istri Saraswati	Bedah	RSUP. Sanglah	Bedah	Jln. Pulau Nias No.1	P	08123812345
51710020020001	Dr. Ni Luh Putu Vebriyanti	Gizi	RSUD. Wangaya	Anak	Jln. Diponegoro No.99	P	081338009876
51710010030001	Dr. Ketut Sudarmajaya	Kulit dan kelamin	RSUP. Sanglah	Kulit dan Kelamin	Jln. Diponegoro	L	085678909876
51710010040001	Dr. David	Umum	RSUP. Sanglah	Umum	Jln. Tamrin	L	081222222222
51710020010001	Dr. Arief Parwoto	Bedah	RSUD. Wangaya	Bedah	Jln. Kebo Iwa	L	085666666666
51710030050001	Dr. Ni Putu Dina Wahyuni	THT	RS. Bhakti Rahayu	THT	Jln. Kepundung	P	081237888888
51710030060001	Dr. I Made Birawan	Anak	RS. Bhakti Rahayu	Anak	Jln. Imam Bonjol	L	085678898988
51710010030002	Dr. Ariana	Kulit dan Kelamin	RSUP. Sanglah	Kulit dan Kelamin	Jln. Teuku Umar	L	081234567890
51710010070001	Dr. Endang Soekrawati	Mata	RSUP.Sanglah	Mata	Jln. Hasanudin	P	085237899877

Gambar 3.83 Contoh Tabel Dokter

Gambar 3.83 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel dokter yaitu ID_Dokter, Spesialisasi, Rumah_Sakit, Poliklinik,

Alamat, Jenis_Kelamin, dan Telepon. Dalam tabel ini ID_Dokter merupakan primary key.

14. Tabel Obat

ID Obat	Nama Obat
0101001	Aclam Caplet
0101002	Ambroxol Tablet
0101003	Acetensa Tablet
0102001	Acipar Caplet
0102001	Accolate Tablet
0204001	Amoxicillin Caplet
0204002	Amoxil Vial 1 G
0201001	Fatigon Spirit
0201002	Ostac Capsule

Gambar 3.84 Contoh Tabel Obat

Gambar 3.84 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel Obat yaitu ID_Obat dan Nama_Obat. Dalam tabel ini ID_Obat merupakan primary key.

15. Tabel Resep Obat

ID Resep Obat	ID Pemeriksaan	Tanggal_Resep	Dosis
5202091206800001	5202091206800001	20/08/2012	200 mg

Gambar 3.85 Contoh Tabel Resep Obat

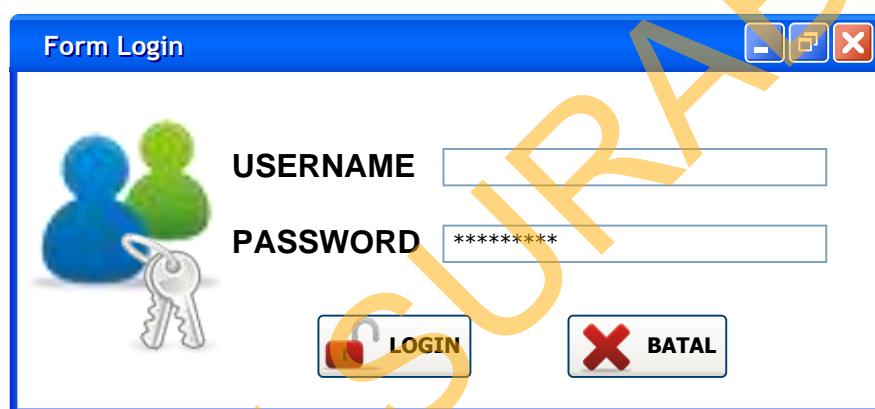
Gambar 3.85 memperlihatkan data yang tersimpan dalam setiap field yang ada pada tabel Obat yaitu ID_Resep_Obat, ID_Pemeriksaan, Tanggal_Resep dan Dosis. Dalam tabel ini ID_Resep_Obat merupakan primary key.

3.6 Desain Interface

Dalam merancang desain interface sistem informasi rekam kesehatan elektronik terpusat ini, tools yang akan digunakan adalah Microsoft Visio 2003.

Dalam desain interface ini, secara umum terdapat master page untuk masing-masing use. Kemudian dalam master page tersebut memiliki content place holder, yang merupakan tempat kontrol-kontrol komponen pendukung aplikasi ini. Desain interface yang dijelaskan dalam tugas akhir ini antara lain sebagai berikut :

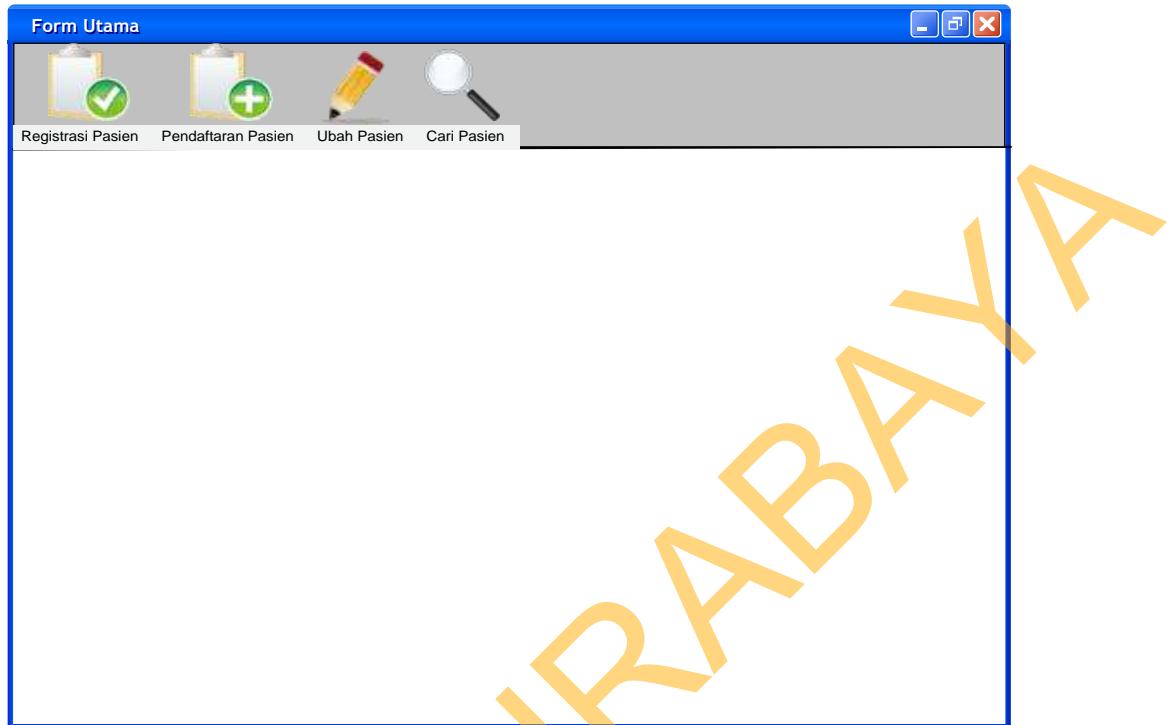
1. Form Login



Gambar 3.86 Form Login

Pada gambar 3.86 menampilkan page login yang akan digunakan oleh masing-masing user untuk dapat berinteraksi dengan sistem. User yang nantinya berinteraksi dengan sistem merupakan orang yang diberi hak akses dengan memiliki username dan password.

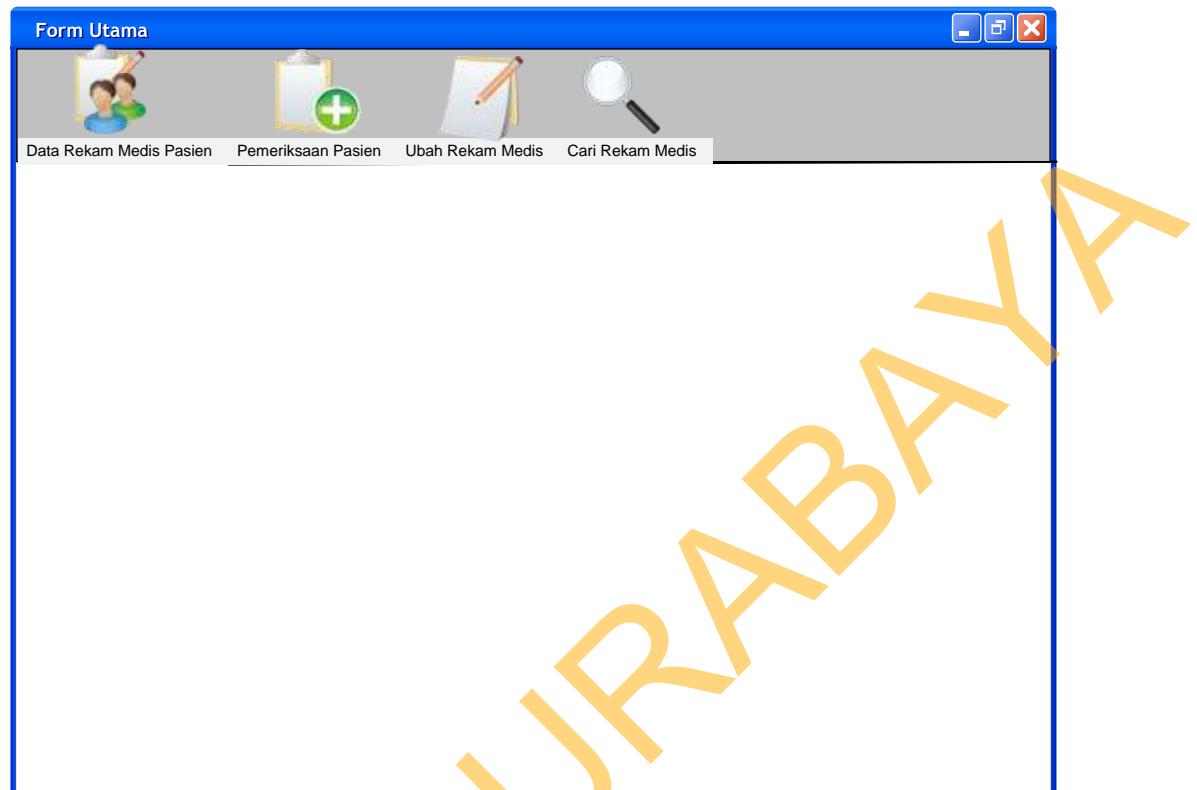
2. Form Utama Untuk Operator Registrasi



Gambar 3.87 Form Utama untuk Operator Registrasi

Pada gambar 3.87 yang menampilkan form utama untuk operator registrasi. Operator registrasi, memiliki hak akses untuk melakukan registrasi pasien, pendaftaran pasien baru, ubah pasien, dan mencari pasien. Untuk itu operator registrasi dapat memilih menu yang telah disediakan sesuai dengan kebutuhan.

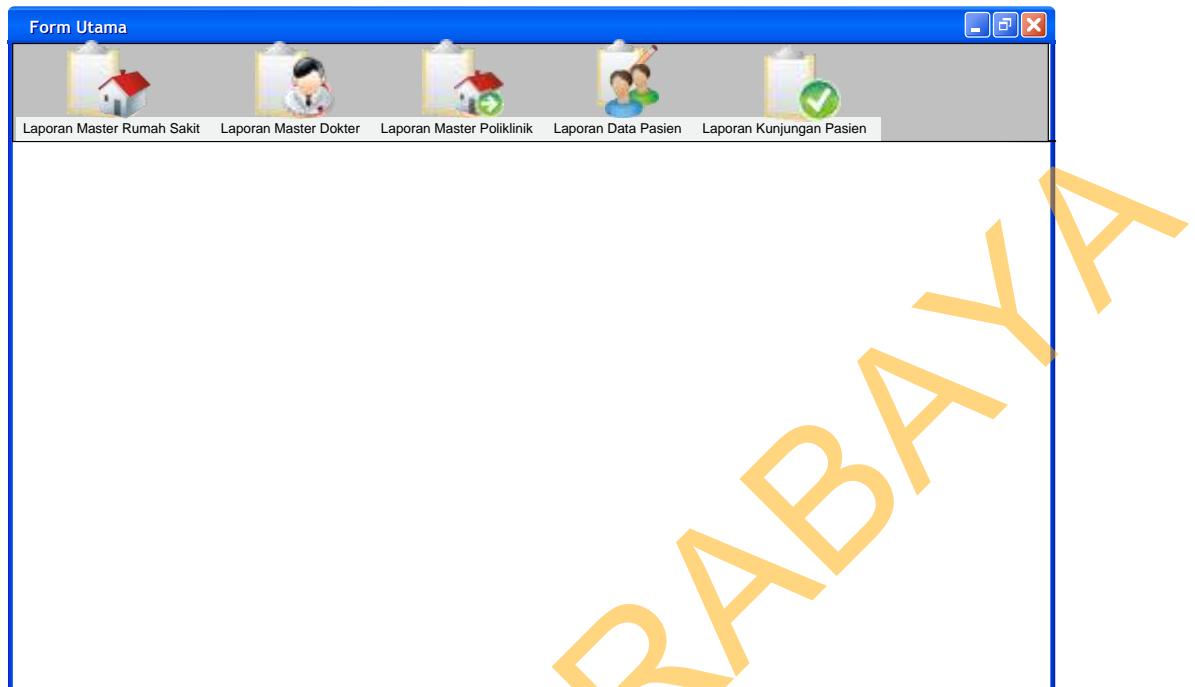
3. Form Utama Untuk Dokter dan Bagian Rekam Medis



Gambar 3.88 Form Utama untuk Dokter dan Bagian Rekam Medis

Pada gambar 3.88 yang menampilkan form utama untuk dokter dan bagian rekam medis. Dokter dan bagian rekam medis, memiliki hak akses untuk melihat rekam kesehatan, menambahkan pemeriksaan pasien, ubah rekam medis, dan mencari rekam medis. Untuk menjalankan tugasnya dokter dan bagian rekam medis dapat memilih menu yang telah disediakan sesuai dengan kebutuhan.

4. Form Utama Untuk Admin



Gambar 3.89 Form Utama untuk Admin

Pada gambar 3.89 yang menampilkan form utama untuk admin. Admin memiliki hak akses untuk menambahkan data user sistem, Mencetak laporan rumah sakit, mencetak laporan dokter, mencetak laporan poliklinik, mencetak laporan data pasien dan mencetak laporan kunjungan pasien. Untuk menjalankan tugasnya admin medis dapat memilih menu yang telah disediakan sesuai dengan kebutuhan.

5. Form Master Desa

ID Desa	Nama Desa	ID Kecamatan
5171032006	Dauh Puri Kangin	517103
5171032003	Dauh Puri Kauh	517103
5171032004	Dauh Puri Klod	517103
5171032011	Padangsambian Kaja	517103
5171032001	Padangsambian Klod	517103
5171032002	Pemecutan Klod	517103
5171032008	Tegal Harum	517103
5171032009	Tegal Kerta	517103

 A 'UBAH' (Change) button with a pencil icon is located at the bottom of the gridview area. The entire form is set against a blue background with a watermark 'STIKOM BANDUNG' diagonally across it."/>

Gambar 3.90 Form Tambah Desa

Pada gambar 3.90 yang menampilkan form desa. Admin memiliki hak akses untuk menambahkan data desa. Data desa yang ditambahkan adalah dengan memasukkan nama desa dan memilih kecamatan. ID_Desa telah ditentukan otomatis oleh sistem. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

6. Form Master Kecamatan

ID Kecamatan	Nama Kecamatan	ID Kabupaten/Kota
517103	Denpasar Barat	5171
517104	Denpasar Utara	5171
517102	Denpasar Timur	5171
517101	Denpasar Selatan	5171

Gambar 3.91 Form Tambah Kecamatan

Pada gambar 3.91 yang menampilkan form kecamatan. Admin memiliki hak akses untuk menambahkan data kecamatan. Data kecamatan yang ditambahkan adalah dengan memasukkan nama kecamatan dan memilih kabupaten/kota. ID_Kecamatan telah ditentukan otomatis oleh sistem. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

7. Form Master Kabupaten

Tambah Kabupaten/Kota

TAMBAH KABUPATEN/KOTA

Nama Kabupaten/Kota	<input type="text" value="Denpasar"/>																														
Nama Provinsi	<input type="text" value="Bali"/>																														
<input style="width: 100px; height: 30px; border: 1px solid black; border-radius: 5px; padding: 5px; margin-right: 10px;" type="button" value="SIMPAN"/> <input style="width: 100px; height: 30px; border: 1px solid black; border-radius: 5px; padding: 5px;" type="button" value="BATAL"/> <input style="width: 100px; height: 30px; border: 1px solid black; border-radius: 5px; padding: 5px;" type="button" value="KELUAR"/>																															
<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>ID Kabupaten/Kota</th> <th>Nama Kabupaten/Kota</th> <th>ID Provinsi</th> </tr> </thead> <tbody> <tr><td>5171</td><td>Denpasar</td><td>51</td></tr> <tr><td>5103</td><td>Badung</td><td>51</td></tr> <tr><td>5102</td><td>Tabanan</td><td>51</td></tr> <tr><td>5101</td><td>Jembarana</td><td>51</td></tr> <tr><td>5105</td><td>Kelungkung</td><td>51</td></tr> <tr><td>5107</td><td>Karangasem</td><td>51</td></tr> <tr><td>5106</td><td>Bangli</td><td>51</td></tr> <tr><td>5104</td><td>Gianyar</td><td>51</td></tr> <tr><td>5108</td><td>Singaraja</td><td>51</td></tr> </tbody> </table>		ID Kabupaten/Kota	Nama Kabupaten/Kota	ID Provinsi	5171	Denpasar	51	5103	Badung	51	5102	Tabanan	51	5101	Jembarana	51	5105	Kelungkung	51	5107	Karangasem	51	5106	Bangli	51	5104	Gianyar	51	5108	Singaraja	51
ID Kabupaten/Kota	Nama Kabupaten/Kota	ID Provinsi																													
5171	Denpasar	51																													
5103	Badung	51																													
5102	Tabanan	51																													
5101	Jembarana	51																													
5105	Kelungkung	51																													
5107	Karangasem	51																													
5106	Bangli	51																													
5104	Gianyar	51																													
5108	Singaraja	51																													
<input style="width: 100px; height: 30px; border: 1px solid black; border-radius: 5px; padding: 5px;" type="button" value="UBAH"/>																															

Gambar 3.92 Form Tambah Kabupaten/Kota

Pada gambar 3.92 yang menampilkan form Kabupaten/Kota. Admin memiliki hak akses untuk menambahkan data kabupaten. Data kabupaten yang ditambahkan adalah dengan memasukkan nama kabupaten dan memilih provinsi. ID_Kabupaten telah ditentukan otomatis oleh sistem. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

8. Form Master Provinsi

ID Provinsi	Nama Provinsi
31	DKI. Jakarta
32	Jawa Barat
33	Jawa Tengah
36	Banten
34	Yogyakarta
35	Jawa Timur
51	Bali
52	Nusa Tenggara Barat
53	Nusa Tenggara Timur

 A large yellow watermark 'STIKOM SURABAYA' is diagonally across the image."/>

Gambar 3.93 Form Tambah Provinsi

Pada gambar 3.93 yang menampilkan form provinsi. Admin memiliki hak akses untuk menambahkan data provinsi. Data provinsi yang ditambahkan adalah dengan memasukkan nama provinsi. ID_Provinsi telah ditentukan otomatis oleh sistem. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

9. Form Master Pekerjaan

Tambah Pekerjaan

TAMBAH PEKERJAAN

ID Pekerjaan Nama Pekerjaan	01 PNS
--	-----------

ID Pekerjaan	Nama Pekerjaan
01	PNS
02	Wiraswasta
03	Dokter
04	Petani
05	Pelajar
06	Mahasiswa
07	Pedagang
08	Guru
09	Sopir

Gambar 3.94 Form Tambah Pekerjaan

Pada gambar 3.94 yang menampilkan form pekerjaan. Admin memiliki hak akses untuk menambahkan data pekerjaan. Data pekerjaan yang ditambahkan adalah dengan memasukkan nama pekerjaan. ID_Pekerjaan telah ditentukan otomatis oleh sistem. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

10. Form Master Agama

ID Agama	Nama Agama
01	Islam
02	Hindu
03	Kristen Katolik
04	Kristen Protestan
05	Buddha
06	Kong Hu Cu

Gambar 3.95 Form Tambah Agama

Pada gambar 3.95 yang menampilkan form agama. Admin memiliki hak akses untuk menambahkan data agama. Data agama yang ditambahkan adalah dengan memasukkan nama agama. ID_Agama telah ditentukan otomatis oleh sistem. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

11. Form Master Rumah Sakit

Tambah Rumah Sakit

TAMBAH RUMAH SAKIT

Nama Rumah Sakit	RSUP. Sanglah
Alamat	Jln. Kesehatan
Kabupaten/Kota	Denpasar
Provinsi	Bali
Telepon	(0361)227911

 **SIMPAN**  **BATAL**  **KELUAR**

ID Rumah Sakit	Nama Rumah Sakit	Alamat	Telepon
5171001	RSUP. Sanglah	Jln. Kesehatan	(0361)227911
5171002	RSUD. Wangaya	Jln. Kartini 133	(0361)222141
5171003	RS. Bhakti Rahayu	Jln. Gatsu II/11	(0361)430270

 **UBAH**

Gambar 3.96 Form Tambah Rumah Sakit

Pada gambar 3.96 yang menampilkan form rumah sakit. Admin memiliki hak akses untuk menambahkan data rumah sakit. Data rumah sakit yang ditambahkan adalah dengan memasukkan nama rumah sakit, alamat dan telepon. ID_rumah_sakit telah ditentukan otomatis sesuai dengan letak rumah sakit. Akan

disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

12. Form Master Poliklinik

Tambah Poliklinik

TAMBAH POLIKLINIK

Nama Polilinik

Rumah Sakit

 **SIMPAN**  **BATAL**  **KELUAR**

ID Poliklinik	Nama Rumah Sakit	Nama Poliklinik
5171001001	RSUP. SANGLAH	Bedah Umum
5171001002	RSUP. SANGLAH	Bedah Saraf
5171001003	RSUP. SANGLAH	Kulit dan Kelamin
5171001004	RSUP. SANGLAH	THT
5171001005	RSUP. SANGLAH	Anak
5171001006	RSUP. SANGLAH	Penyakit Dalam
5171001007	RSUP. SANGLAH	Urologi
5171001008	RSUP. SANGLAH	Mata
5171001009	RSUP. SANGLAH	Dokter Umum

 **UBAH**

Gambar 3.97 Form Tambah Poliklinik

Pada gambar 3.97 yang menampilkan form poliklinik. Admin memiliki hak akses untuk menambahkan data poliklinik. Data poliklinik yang ditambahkan adalah dengan memasukkan nama poliklinik dan dipilih berdasarkan rumah sakitnya. ID_Poliklinik telah ditentukan otomatis oleh sistem. Akan disediakan

beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

13. Form Master Dokter

TAMBAH DOKTER

Nama Dokter	<input type="text"/>
Spesialisasi	<input type="text"/>
Rumah Sakit	RSUP. SANGLAH
Poliklinik	POLIKLINIK BEDAH
Alamat	<input type="text"/>
Jenis Kelamin	<input type="radio"/> Laki-laki <input type="radio"/> Perempuan
Telepon	<input type="text"/>

SIMPAN **CETAK USERNAME** **BATAL** **KELUAR**

Nama Dokter	Nama Dokter	Spesialisasi	Rumah Sakit	Poliklinik	Alamat	Jns. Kelamin	Telepon
51710010010001	Dr. Anak Agung Istri Saraswati	Bedah	RSUP. Sanglah	Bedah	Jln. Pulau Nias No.1	P	08123812345
51710020020001	Dr.Ni Luh Putu Vebriyanti	Gizi	RSUD. Wangaya	Anak	Jln. Diponegoro No.99	P	081338009876
51710010030001	Dr. Ketut Sudarmajaya	Kulit dan kelamin	RSUP. Sanglah	Kulit dan Kelamin	Jln. Diponegoro	L	085678909876
51710010040001	Dr. David	Unum	RSUP. Sanglah	Unum	Jln. Tamrin	L	081222222222
51710020010001	Dr. Arief Parwoto	Bedah	RSUD. Wangaya	Bedah	Jln. Kebo Iwa	L	085666666666
51710030050001	Dr. NI Putu Dina Wahyuni	THT	RS. Bhakti Rahayu	THT	Jln. Kepundung	P	081237888888
51710030060001	Dr. I Made Birawan	Anak	RS. Bhakti Rahayu	Anak	Jln. Imam Bonjol	L	085678898988
51710010030002	Dr. Ariana	Kulit dan Kelamin	RSUP. Sanglah	Kulit dan Kelamin	Jln. Teuku Umar	L	081234567890
51710010070001	Dr. Endang Soekrawati	Mata	RSUP.Sanglah	Mata	Jln. Hasanudin	P	085237899877

UBAH

Gambar 3.98 Form Tambah Dokter

Pada gambar 3.98 yang menampilkan form Dokter. Admin memiliki hak akses untuk menambahkan data dokter. Data dokter yang ditambahkan adalah dengan memasukkan nama dokter, spesialisasi, pilih rumah sakit, pilih poliklinik, alamat dokter, pilih jenis kelamin, dan nomor telepon. ID_Dokter telah ditentukan otomatis oleh sistem. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

14. Form Master Obat

Tambah Obat

TAMBAH OBAT

Nama Obat	<input type="text"/>
Tipe Obat	<input type="text" value="Generik"/>
Jenis Obat	<input type="text" value="Keras"/>

ID Obat	Nama Obat
0101001	Aclam Caplet
0101002	Ambroxol Tablet
0101003	Acetensa Tablet
0102001	Acipar Caplet
0102001	Accolate Tablet
0204001	Amoxicillin Caplet
0204002	Amoxil Vial 1 G
0201001	Fatigon Spirit
0201002	Ostac Capsule

Gambar 3.99 Form Tambah Obat

Pada gambar 3.99 yang menampilkan form obat. Admin memiliki hak akses untuk menambahkan data obat. Data obat yang ditambahkan adalah dengan memasukkan nama obat. ID_Obat telah ditentukan otomatis sesuai dengan jenis obatnya dan tipe obatnya. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Data yang telah disimpan akan ditampilkan dalam gridview.

15. Form Registrasi Pasien Langsung

Gambar 3.100 Form Registrasi Pasien Secara Langsung

Pada gambar 3.100 yang menampilkan form registrasi pasien secara langsung. Operator registrasi memiliki hak akses untuk menambahkan data registrasi pasien. Data registrasi yang ditambahkan sesuai dengan petunjuk pengisian. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Form registrasi ini akan diisi pada saat pasien datang ke loket untuk meminta pelayanan di rumah sakit.

16. Form Registrasi Pasien Dengan Rujukan

REGISTRASI PASIEN

ID Pasien 00 00 00 0000 000000 0000

CARI DAFTAR BARU

Registrasi Pasien

Nama Pasien

Alamat

Jenis Kelamin

Umur

Nama Penanggung Jawab

Tanggal Registrasi 29 08 2012

Rujukan

Ada Rujukan Tidak Ada Rujukan

Baru

Tanggal Rujukan 29 08 2012

Rumah Sakit Perujuk

Poliklinik Perujuk

Poliklinik Tujuan

Status Gawat Darurat

REGISTRASI

Gambar 3.101 Form Registrasi Pasien dengan Rujukan

Pada gambar 3.101 yang menampilkan form registrasi pasien dengan rujukan. Operator registrasi memiliki hak akses untuk menambahkan data registrasi pasien. Data registrasi yang ditambahkan sesuai dengan petunjuk pengisian. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Form registrasi ini akan diisi pada saat pasien datang ke loket dengan membawa rujukan dari rumah sakit lain untuk meminta pelayanan di rumah sakit.

17. Form Pendaftaran Pasien Baru

Gambar 3.102 Form Pendaftaran Pasien Baru

Pada gambar 3.102 yang menampilkan form pendaftaran pasien baru ke dalam sistem. Operator registrasi memiliki hak akses untuk menambahkan data pendaftaran pasien. Data pendaftaran pasien yang ditambahkan sesuai dengan petunjuk pengisian. Akan disediakan beberapa tombol yang dapat digunakan sesuai dengan kebutuhan. Form registrasi ini akan diisi pada saat pasien datang ke loket belum terdaftar pada sistem untuk meminta pelayanan di rumah sakit.

18. Form Cari Pasien

Gambar 3.103 Form Cari Data Pasien

Pada gambar 3.103 yang menampilkan form cari data pasien. Operator registrasi memiliki hak akses untuk mencari data pasien. Data pasien yang dicari dapat berupa kode pasien atau nama pasien kemudian memilih tombol cari untuk menampilkan data pasien yang dicari. Untuk melihat detail penanggung jawab pasien, dapat memilih tombol lihat penanggung jawab dan disediakan beberapa tombol lain yang dapat digunakan sesuai dengan kebutuhan.

19. Form Rekam kesehatan

Gambar 3.104 Form Cari Rekam kesehatan

Pada gambar 3.104 yang menampilkan form cari data rekam kesehatan yang berisi riwayat pemeriksaan pasien. Dokter atau bagian rekam medis memiliki hak akses untuk mencari data rekam kesehatan. Data rekam kesehatan dapat dicari dengan memasukkan kode pasien kemudian memilih tombol cari untuk menampilkan data rekam kesehatan yang dicari. Untuk melihat resep obat, dapat memilih tombol lihat resep obat dan disediakan beberapa tombol lain yang dapat digunakan sesuai dengan kebutuhan.

20. Form Pemeriksaan Pasien

Pemeriksaan Pasien

PEMERIKSAAN PASIEN

Tanggal Pemeriksaan: 29 08 2012

Rumah Sakit: RSUP. Sanglah

Poliklinik: Bedah

Nama Dokter: Dr. David

Nama Pasien: Sdr. Johan

Keluhan:

Diagnosa:

Tindakan Medis:

Alergi Obat:

Nasehat:

Resep Obat:

RESEP OBAT

SIMPAN **BATAL** **KELUAR**

Gambar 3.105 Form Pemeriksaan Pasien

Pada gambar 3.105 yang menampilkan form pemeriksaan. Dokter atau bagian rekam medis memiliki hak akses untuk menambahkan data pemeriksaan pasien. Data pemeriksaan dimasukkan sesuai dengan field yang telah ditentukan pada form. Pada form juga disediakan beberapa tombol lain yang dapat digunakan sesuai dengan kebutuhan user.

21. Form Resep Obat

ID Obat	Nama Obat	Dosis
Enter Text	Enter Text	Enter Text
Enter Text	Enter Text	Enter Text

Gambar 3.106 Form Resep Obat

Pada gambar 3.106 yang menampilkan form resep obat. Dokter atau bagian rekam medis memiliki hak akses untuk menambahkan data resep obat pasien. Data resep obat dimasukkan sesuai dengan field yang telah ditentukan pada form. Beberapa obat yang ditulis diresep akan dipilih pada combo box yang

telah disediakan. Pada form juga disediakan beberapa tombol lain yang dapat digunakan sesuai dengan kebutuhan user.

22. Form Penanggung Jawab

The form is titled 'PENANGGUNG JAWAB'. It includes the following fields:

- Nama Penanggung Jawab (Name): Text input field
- Alamat (Address): Text input field
- Desa (Village): Drop-down menu with options: Serangan, Denpasar Selatan, Denpasar, Bali
- Kecamatan (Sub-district): Drop-down menu with options: Denpasar Selatan, Denpasar
- Kabupaten/Kota (District/City): Drop-down menu with options: Denpasar, Bali
- Provinsi (Province): Drop-down menu with options: Bali
- Pekerjaan (Occupation): Text input field
- Telepon (Telephone): Text input field
- Hubungan Keluarga (Family Relationship): Text input field

At the bottom are two buttons: 'SIMPAN' (Save) with a disk icon and 'BATAL' (Cancel) with a red X icon.

Gambar 3.107 Form Penanggung Jawa

Pada gambar 3.107 yang menampilkan form penanggung jawab terhadap pasien. Operator registrasi memiliki hak akses untuk menambahkan data penanggung jawab terhadap pasien. Data penanggung jawab dimasukkan sesuai dengan field yang telah ditentukan pada form. Pada form juga disediakan beberapa tombol lain yang dapat digunakan sesuai dengan kebutuhan user.

23. Form Cetak Master Rumah Sakit

Gambar 3.108 Form Cetak Data Rumah Sakit

Pada gambar 3.108 yang menampilkan form untuk mencetak laporan data rumah sakit sesuai dengan periode yang diinginkan. Admin sistem memiliki hak akses untuk mencetak laporan data rumah sakit yang terdaftar pada sistem.

24. Form Cetak Master Dokter

Gambar 3.109 Form Cetak Data Dokter

Pada gambar 3.109 yang menampilkan form untuk mencetak laporan data dokter sesuai dengan periode yang diinginkan. Laporan data dokter juga dapat dipilih berdasarkan beberapa katagori yaitu : laporan data dokter keseluruhan sistem, laporan dokter tiap-tiap rumah sakit, dan laporan data dokter tiap-tiap poliklinik. Admin sistem memiliki hak akses untuk mencetak laporan data dokter yang terdaftar pada sistem.

25. Form Cetak Master Poliklinik

Gambar 3.110 Form Cetak Data Poliklinik

Pada gambar 3.110 yang menampilkan form untuk mencetak laporan data poliklinik yang ada di tiap-tiap rumah sakit. Admin sistem memiliki hak akses untuk mencetak laporan data poliklinik masing-masing rumah sakit yang terdaftar pada sistem.

26. Form Cetak Laporan Data Pasien

Gambar 3.111 Form Cetak Data Pasien

Pada gambar 3.111 yang menampilkan form untuk mencetak laporan data pasien. Laporan data pasien juga dapat dipilih berdasarkan beberapa katagori yaitu : laporan data pasien keseluruhan sistem, laporan pasien tiap-tiap rumah sakit, dan laporan data pasien tiap-tiap poliklinik. Admin sistem memiliki hak akses untuk mencetak laporan data pasien yang terdaftar pada sistem.

27. Form Cetak Laporan Data Kunjungan Pasien

Gambar 3.112 Form Cetak Data Kunjungan Pasien

Pada gambar 3.112 yang menampilkan form untuk mencetak laporan data kunjungan pasien. Laporan data kunjungan pasien juga dapat dipilih berdasarkan beberapa katagori yaitu : laporan data kunjungan pasien keseluruhan sistem, laporan data kunjungan pasien tiap-tiap rumah sakit, dan laporan data kunjungan pasien tiap-tiap poliklinik. Admin sistem memiliki hak akses untuk mencetak laporan data kunjungan pasien yang terdaftar pada sistem.

28. Laporan Master Rumah Sakit

Gambar 3.113 Laporan Data Rumah Sakit

29. Laporan Master Dokter Keseluruhan

Gambar 3.114 Laporan Data Dokter Keseluruhan

30. Laporan Master Dokter Tiap Rumah Sakit

Gambar 3.115 Laporan Data Dokter Tiap Rumah Sakit

31. Laporan Master Dokter Tiap Poliklinik Satu Rumah Sakit

Gambar 3.116 Laporan Data Dokter Tiap Poliklinik Satu Rumah Sakit

32. Laporan Master Poliklinik

 **LAPORAN DATA POLIKLINIK**

KOTA MADIYA DENPASAR

Periode : /2012 s/d /2012

RSUP. SANGLAH

ID Poliklinik	Nama Poliklinik

Denpasar,
(Bagian Administrasi)

Gambar 3.117 Laporan Data Poliklinik Tiap Rumah Sakit

33. Laporan Data Pasien Secara Keseluruhan

Gambar 3.118 Laporan Data Pasien Secara Keseluruhan

34. Laporan Data Pasien Tiap Rumah Sakit

Gambar 3.119 Laporan Data Pasien Tiap Rumah Sakit

35. Laporan Data Pasien Tiap Poliklinik

Gambar 3.120 Laporan Data Pasien Tiap Poliklinik

36. Laporan Data Pasien Tiap Poliklinik Satu Rumah Sakit

Gambar 3.121 Laporan Data Pasien Tiap Poliklinik Satu Rumah Sakit

37. Laporan Kunjungan Pasien Secara Keseluruhan

 LAPORAN KUNJUNGAN PASIEN KOTA MADYA DENPASAR					
Tahun : 2012					
RSUP. SANGLAH					
ID Pendaftaran	ID Pasien	Nama Pasien	Keluhan	Tanggal Pendaftaran	Status
Total Kunjungan					0
RSUD. WANGAYA					
ID Pendaftaran	ID Pasien	Nama Pasien	Keluhan	Tanggal Pendaftaran	Status
Total Kunjungan					0
RS. BHAKTI RAHAYU					
ID Pendaftaran	ID Pasien	Nama Pasien	Keluhan	Tanggal Pendaftaran	Status
Total Kunjungan					0
Denpasar,					
(Bagian Administrasi)					

Gambar 3.122 Laporan Data Kunjungan Pasien Keseluruhan

38. Laporan Kunjungan Pasien Tiap Rumah Sakit

Gambar 3.123 Laporan Data Kunjungan Pasien Tiap Rumah Sakit

39. Laporan Kunjungan Pasien Tiap Poliklinik Satu Rumah Sakit

Gambar 3.124 Laporan Data Kunjungan Pasien Tiap Poliklinik Satu Rumah Sakit

40. Laporan Kunjungan Pasien Tiap Poliklinik

LAPORAN KUNJUNGAN PASIEN

KOTA MADYA DENPASAR

Tahun : 2012

POLIKLINIK BEDAH

ID Pendaftaran	Rumah Sakit	ID Pasien	Nama Pasien	Keluhan	Tanggal Pendaftaran	Status

Total Kunjungan

Denpasar,
(Bagian Administrasi)

Gambar 3.125 Laporan Data Kunjungan Pasien Tiap Poliklinik

41. Resep Obat

Gambar 3.126 Resep Obat

42. Kartu Berobat Pasien



KARTU REKAM MEDIS PASIEN ELEKTRONIK TERPUSAT



Gambar 3.127 Kartu Berobat Pasien

43. Form Tambah Data User

Gambar 3.128 Form Tambah User

44. Form Pesan

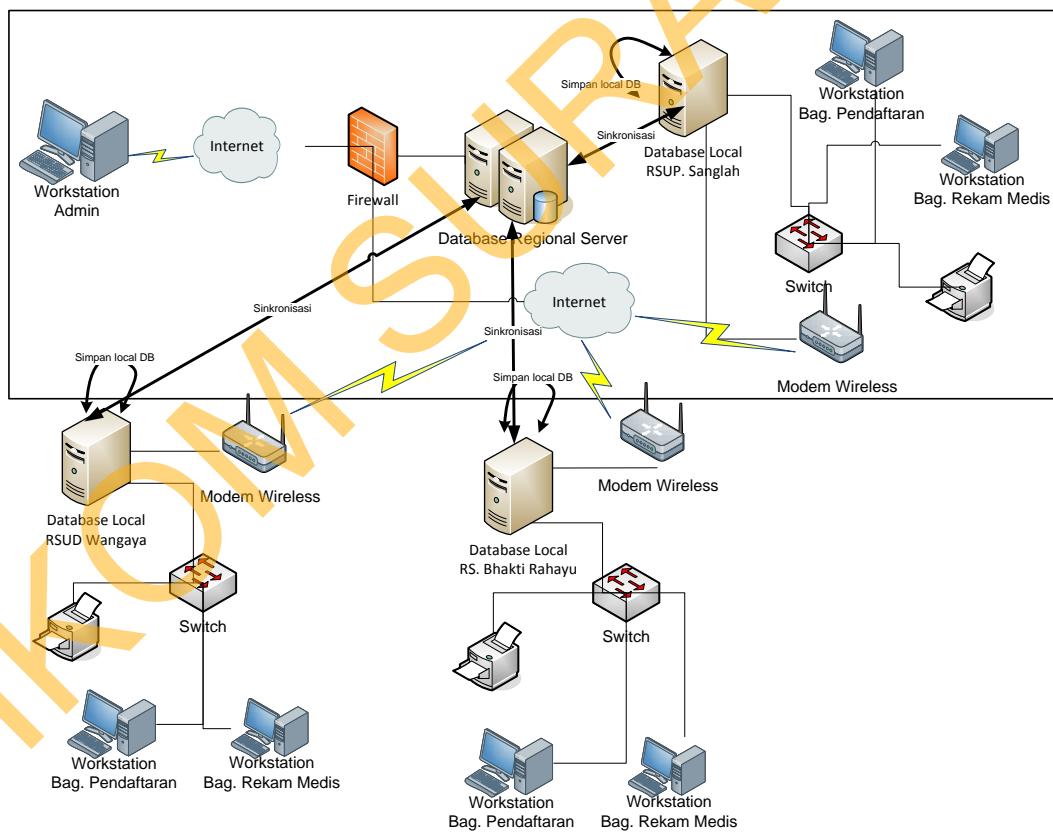


Gambar 3.129 Form Pesan

3.7 Topologi Jaringan, Sinkronisasi Data, Integrity, Confidentiality dan Availability untuk Sistem Rekam Kesehatan Elektronik Terpusat

Aspek keamanan di dalam jaringan yang perlu diperhatikan adalah otentikasi, *integrity*, *confidentiality*, dan otorisasi. Pada rancangan sistem ini akan digambarkan juga rancangan topologi jaringannya. Selain itu untuk unsur-unsur keamanan (safety), kerahasiaan (confidential), sekuritas, privasi dan integritas data yang akan digunakan dalam rancangan sistem rekam kesehatan elektronik ini akan digambarkan dan dijelaskan sebagai berikut :

A. Rancangan Topologi Jaringan dan Sinkronisasi Data



Gambar 3.130 Desain Jaringan Sistem

Ket :



: Sinkronisasi data setiap 2 jam



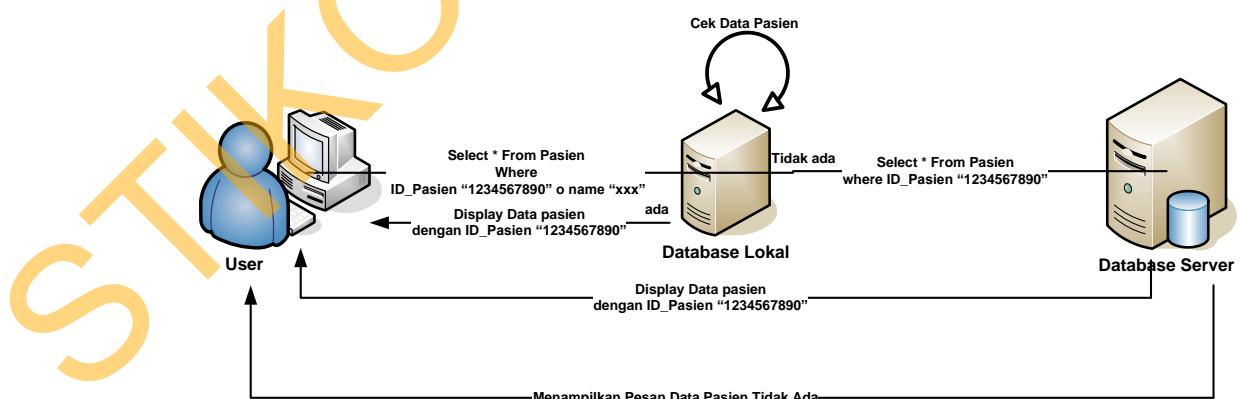
: Menyimpan di database lokal

Dari gambar topologi jaringan diatas digambarkan proses-proses yang terjadi pada sistem rekam kesehatan pasien elektronik terpusat. Dimulai dari data rekam medis yang telah dicatat pada saat pemeriksaan pasien di rumah sakit, data rekam medis pasien akan disimpan pada database local terlebih dahulu. Setiap transaksi pemeriksaan pasien akan disimpan di dalam database local rumah sakit. Dari tabel local rumah sakit, setiap 2 jam akan sistem akan melakukan sinkronisasi data dari database local ke database server melalui koneksi internet.

Apabila terjadi mati listrik ataupun gangguan pada jaringan internet, proses penyimpanan data akan disimpan di dalam database local saja. Setelah koneksi internet dapat digunakan, 2 jam setelah koneksi internet jalan akan terjadi sinkronisasi otomatis oleh sistem.

B. Hubungan Antara Database Lokal dengan Database Server

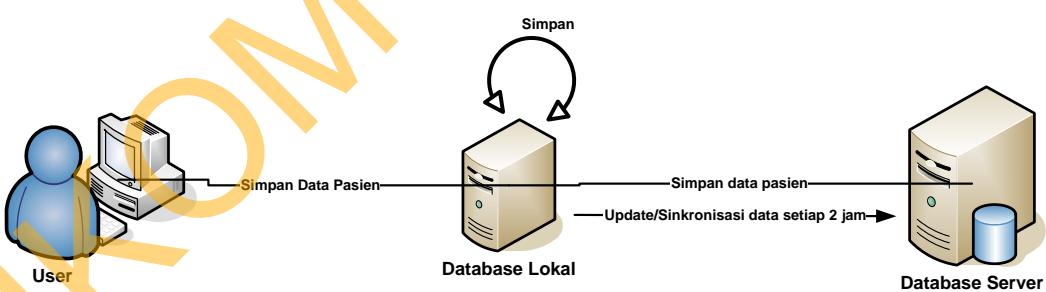
Pada gambar berikut akan digambarkan hubungan antara database local yang ada di masing-masing rumah sakit dengan database yang berada di regional server.



Gambar 3.131 Proses Pencarian Data Pasien

Gambar diatas menggambarkan proses membaca data dari database local dan database server. Ketika user ingin melakukan pencarian data pasien yang melakukan pendaftaran, setelah user menginputkan kode pasien atau nama pasien dan mengklik tombol cari, sistem akan memeriksa data pasien yang dicari tersebut di dalam database local terlebih dahulu. Apabila data pasien ditemukan dalam database local, data pasien akan ditampilkan kepada user. Sedangkan apabila data pasien yang dicari tidak ditemukan di dalam database local, sistem akan melakukan pemeriksaan ke dalam database server. Seperti yang terjadi di database local, apabila data pasien ditemukan pada database server, data akan ditampilkan ke user. Namun jika pasien merupakan pasien baru dan data pasien tidak ditemukan dalam database local maupun database server sistem akan menampilkan pesan bahwa data pasien yang dicari tidak ditemukan.

Adapun hubungan dan proses yang terjadi pada saat penyimpanan transaksi-transaksi yang terjadi pada sistem digambarkan pada gambar berikut.



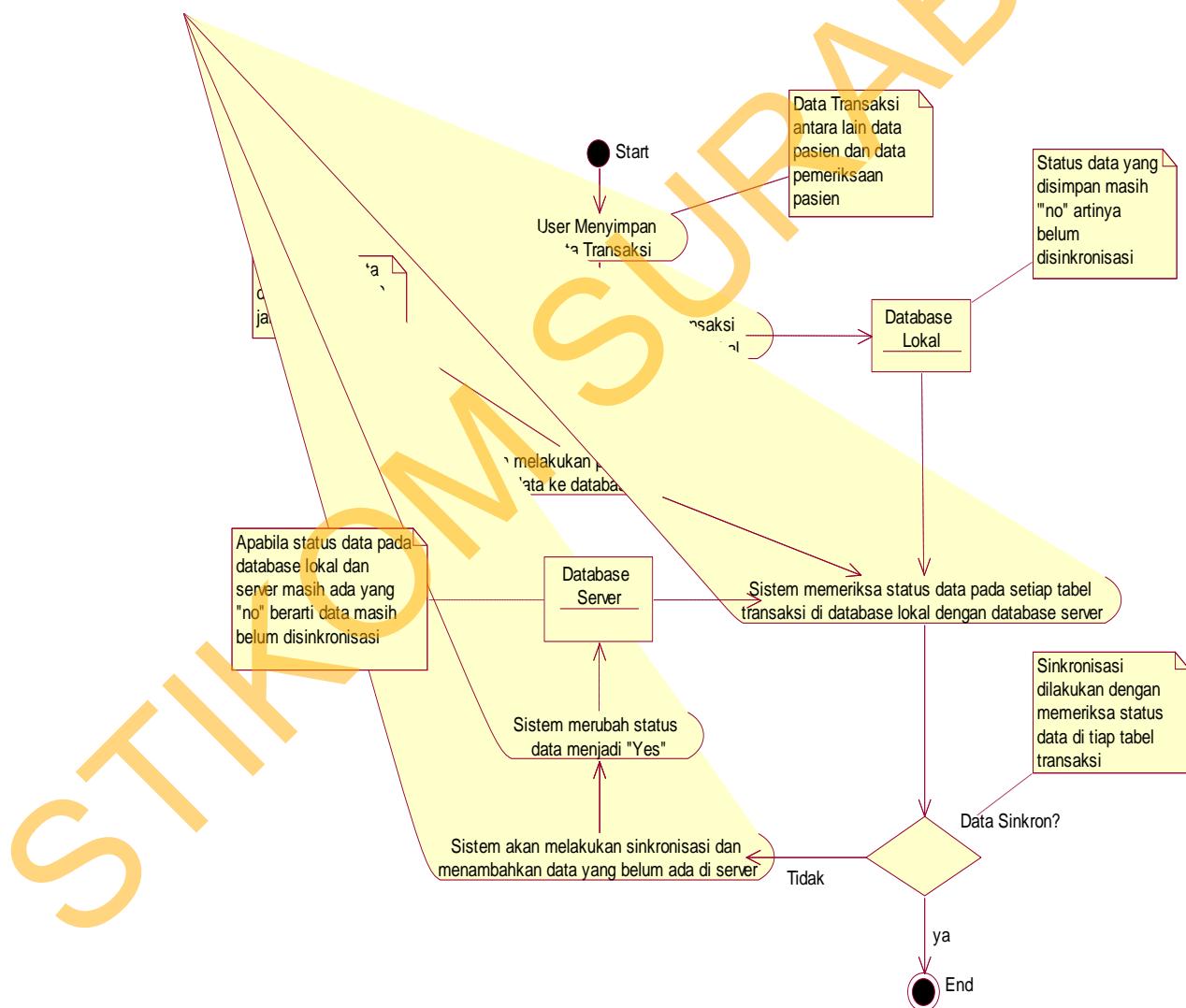
Gambar 3.132 Proses Penyimpanan Data

Proses penyimpanan data ketika user telah melakukan penyimpanan, sistem akan menyimpan data transaksi yang dilakukan oleh user ke dalam database local terlebih dahulu. Dari database local, setiap 2 jam sekali akan dilakukan sinkronisasi data ke dalam database server. Penyimpanan ke database local terlebih dahulu dilakukan untuk mengatasi masalah apabila terjadi mati listrik

atau gangguan koneksi ke database server. Apabila terjadi permasalahan tersebut, dua jam setelah koneksi ke database server dapat digunakan kembali, sistem akan secara otomatis melakukan sinkronisasi ke database server. Jadi database server akan terupdate setiap dua jam.

C. Proses Sinkronisasi Database Lokal ke Database Server

Untuk memastikan data pada database server telah terupdate dengan data yang baru, maka perlu adanya proses sinkronisasi. Proses sinkronisasi pada rancangan sistem ini, akan digambarkan pada gambar diagram berikut.



Gambar 3.133 Proses Sinkronisasi Database local ke Database Server

Pada saat data transaksi disimpan oleh user, data akan disimpan dalam database local sesuai dengan tabel transaksinya dan status data awal adalah “NO” karena data masih disimpan di database lokal. Sebagai contoh untuk transaksi pendaftaran pasien. Data transaksi pendaftaran pasien baru akan disimpan dalam tabel data pasien.

Tabel 3.20 Tabel Data Pasien yang Belum Disinkronisasi

Dari tabel data pasien diatas, dapat dilihat data pasien masih memiliki status data “NO” yang artinya data masih belum disinkronisasi dengan database server. Setelah proses sinkronisasi data, data pasien tersebut akan ditambahkan dan disimpan juga dalam database server. Setelah data server diperbaharui, status data pasien akan berubah menjadi “YES” yang artinya data pasien di database lokal telah sinkron dengan data pasien yang ada pada database server.

Tabel 3.21 Tabel Data Pasien yang Telah Disinkronisasi

D. Integrity

Merupakan aspek yang menjamin bahwa data tidak boleh berubah tanpa ijin pihak yang berwenang (authorized). Pada rancangan sistem ini menggunakan DBMS. Adapun penggunaan DBMS yaitu :

1. Data dapat digunakan bersama
2. Redudansi data bisa dikurangi
3. Ketidakkonsistenan data dapat dihindari
4. Integritas data dapat dipelihara
5. Keamanan terjamin
6. Kebutuhan user yang kompleks dapat teratasi
7. Pelaksanaan standarisasi
8. Produktivitas meningkat
9. Layanan backup dan recovery yang semakin baik

E. Confidential

Aspek *confidentiality* adalah perlindungan terhadap informasi yang diberikan agar tidak jatuh kepada pihak yang tidak berwenang. Aspek keamanan terakhir pada jaringan adalah otorisasi, proses untuk memberikan kewenangan terhadap *principal* yang telah diotentikasi untuk melakukan operasi yang merupakan haknya. Confidentiality merupakan aspek yang menjamin kerahasiaan data atau informasi. Sistem yang digunakan untuk mengimplementasikan e-procurement harus dapat menjamin kerahasiaan data yang dikirim, diterima dan disimpan. Bocornya informasi dapat berakibat batalnya proses pengadaan. Kerahasiaan ini dapat diimplementasikan dengan berbagai cara, seperti misalnya menggunakan teknologi kriptografi dengan melakukan proses enkripsi

(penyandian, pengkodean) pada transmisi data, pengolahan data (aplikasi dan database), dan penyimpanan data (storage). Teknologi kriptografi dapat mempersulit pembacaan data tersebut bagi pihak yang tidak berhak. Seringkali perancang dan implementor dari sistem informasi atau sistem transaksi elektronik lalai dalam menerapkan pengamanan. Umumnya pengamanan ini baru diperhatikan pada tahap akhir saja sehingga pengamanan lebih sulit diintegrasikan dengan sistem yang ada. Penambahan pada tahap akhir ini menyebabkan sistem menjadi tambal sulam. Akibat lain dari hal ini adalah adanya biaya yang lebih mahal daripada jika pengamanan sudah dipikirkan dan diimplementasikan sejak awal. Akses terhadap informasi juga harus dilakukan dengan melalui mekanisme otorisasi (authorization) yang ketat. Tingkat keamanan dari mekanisme otorisasi bergantung kepada tingkat kerahasiaan data yang diinginkan.

Kerberos merupakan protokol jaringan yang menangani masalah otentikasi. Nama Kerberos berasal nama dari mitologi Yunani, yaitu Cerberus. Cerberus adalah makhluk berkepala tiga yang menjaga Underworld dari makhluk hidup yang mencoba untuk memasukinya. Protokol Kerberos mulai didesain pada akhir tahun 1980 di Massachusetts Institute of Technology (MIT) sebagai bagian dari proyek Athena. Kerberos adalah mekanisme otentikasi yang ditujukan untuk *distributed server*. Ia memungkinkan server dan client saling mengotentikasi sebelum melakukan koneksi.

Keuntungan Menggunakan Kerberos abtara lain : Otentikasi *password-based* merupakan proses otentikasi dimana *client* mengirimkan *password* dan kemudian setelah diverifikasi oleh server maka *client* tersebut mendapat otorisasi untuk melakukan operasi-operasi tertentu. Otentikasi *password-based* ini sangat

rawan terhadap serangan pihak ketiga, yaitu ketika seorang penyusup berhasil mendapatkan *password* tersebut saat pengiriman *password* berlangsung. Banyak aplikasi yang menggunakan mekanisme otentikasi yang lemah, bahkan mekanisme otentikasi yang menggunakan asersi, misalnya Barkeley R-command dan protocol IDENT.

Mekanisme otentikasi dengan menggunakan asersi ini sangatlah tidak aman karena pertukaran informasi dan keamanannya hanya berdasarkan kepercayaan. Aspek lain dari keamanan pada jaringan adalah bahwa masalah keamanan bukanlah tugas utama aplikasi pada jaringan. Aplikasi *mail server* yang mempunyai tugas utama untuk mengirimkan *email* kepada pihak yang dituju pada jaringan seharusnya tidak bertugas untuk memverifikasi identitas *user*. Oleh karena itu Kerberos dibutuhkan untuk proses otentikasi. Kerberos mempunyai keuntungan untuk melakukan proses otentikasi dari pusat untuk berbagai aplikasi jaringan. Untuk masing-masing aplikasi yang membutuhkan pelayanan Kerberos maka Kerberos sangat *reliable*, *simple* dan mudah digunakan. Terlebih lagi, Kerberos menghindarkan aplikasi jaringan dari tugas untuk melakukan otentikasi.

Data yang dikirim melalui jaringan dapat dirusak, dilihat, ataupun dimodifikasi isinya. Kerberos menyediakan otentikasi kriptografi melalui kombinasi penggunaan kunci rahasia dan *strong* enkripsi. Kerberos menjamin integritas dan kerahasiaan data. Kunci rahasia adalah password yang diketahui oleh *client* atau *server*. Enkripsi dilakukan dengan algoritma kunci simetri menggunakan DES atau *triple DES*. Sekarang juga dikembangkan untuk diimplementasikan menggunakan AES1. Kerberos adalah *tools* yang menyediakan otentikasi untuk pelayanan interaktif, seperti telnet, ftp, pop dan

sebagainya dimana *user* diminta memasukkan password untuk melakukan login secara *real time*. Kunci simetri mengizinkan otentikasi dapat dilakukan secara *real time* karena karakteristiknya yang cepat. Algoritma kunci simetri menggunakan kunci yang sama untuk melakukan enkripsi dan dekripsi.

F. Availability

Availability maka akan digunakan mekanisme mirroring database jadi bila ada kerusakan 1 server, operisional tetap berjalan karena masih ada 1 *server* lagi (mirror) yg tetap berjalan. Alokasi data *DDBS* dapat dilakukan dalam dua kategori, yakni partisi dan replikasi. Dalam skema partisi, *database* dibagi ke sejumlah partisi yang *disjoin* di mana masing-masing ditempatkan pada *site* berbeda. Perancangan replikasi dibedakan atas *fully replication* atau *fully duplicated* yaitu seluruh *database* ditempatkan di masing-masing *site*, atau *partially replicated* yaitu masing-masing partisi *database* disimpan di beberapa *site* tetapi tidak pada semua *site*. Perancangan dasar alokasi DDB dapat dilakukan dengan cara fragmentasi dan distribusi. Dalam fragmentasi, relasi dibagi ke partisi-partisi yang disebut *fragmen*, setiap *fragmen* disimpan pada *site* berbeda. Fragmentasi dibedakan menjadi dua, yaitu fragmentasi horisontal dan vertikal, dan dimungkinkan secara campuran.

Keuntungan dan kerugian Replikasi adalah terkait dengan :

10. *Availability* : jika satu *site* yang berisi relasi r gagal, relasi r masih bisa diperoleh di *site* yang lain, sehingga sistem tetap dapat melanjutkan proses.
11. Meningkatkan paralel: beberapa *site* dapat memproses *query* pada r secara paralel. Semakin banyak replikasi, semakin besar kemungkinan ditemukan

data pada *site* di mana transaksi dijalankan. Replikasi meminimalkan pergerakan data antar *site*.

12. Meningkatkan *overhead update*: sistem harus memastikan bahwa semua replikasi dari relasi *r* konsisten. Di mana pun *r* di-*update*, maka *update* ini harus disebar ke seluruh *site*. Replikasi meningkatkan kinerja operasi *read* dan *availability* data. Transaksi *update* meningkatkan *overhead*. Pengontrolan konkurensi *update* data yang direplikasi akan semakin kompleks dibandingkan *database* terpusat. Cara sederhana adalah membuat salinan utama dari *r*. Misal: dalam sistem perbankan, rekening dapat dihubungkan dengan *site* di mana rekening tersebut dibuka.

G. Keamanan

Sistem ini menggunakan satu central Login dapat mengamankan seluruh jaringan. Jadi untuk satu *user*, memiliki satu username dan password. Untuk Dokter yang bekerja di lebih dari satu rumah sakit, dokter hanya dapat melakukan akses pada data pasien yang memiliki pemeriksaan dengan nama dokter yang bersangkutan, dengan kata lain dokter yang tidak ada keterkaitan dengan seorang pasien, tidak dapat mengakses data pasien tersebut.

3.8 Kebutuhan Perangkat Keras dan Perangkat Lunak

Untuk dapat menjalankan sistem ini maka diperlukan perangkat keras dan perangkat lunak yang dengan kondisi dan persyaratan tertentu agar sistem dapat berjalan dengan baik. Sistem ini nantinya akan digunakan pada desktop milik pegawai maupun petugas yang bertugas di masing-masing bagian rumah sakit.

Adapun syarat kebutuhan dari hardware (Perangkat Keras) untuk server yang harus dipenuhi adalah:

1. *Processor Intel Pentium 4, sejenis atau diatasnya*
2. *Memory 2 GB atau lebih*
3. *Harddisk 160 GB (workstation) dan 2 TB (Server)*
4. *VGA Card dengan RAM 512Mb support OpenGL*
5. *Printer, Mouse dan Keyboard*

Dan kebutuhan perangkat lunak yang digunakan dalam pembuatan desain sistem informasi rekam kesehatan elektronik terpusat ini adalah:

1. *Operating System Windows 7.*
2. Microsoft visio 2003 untuk merancang desain interface dan desain laporan
3. Untuk perancangan database menggunakan *Data Architect Power Designer 6.0.*
4. Untuk rancangan sistem berorientasi objek dengan *Unified Modeling Language* menggunakan Rational Rose 2003

Perhitungan kapasitas harddisk yang diperlukan berdasarkan perhitungan rata-rata kunjungan pasien 3 rumah sakit selama 5 tahun tahun 2011:

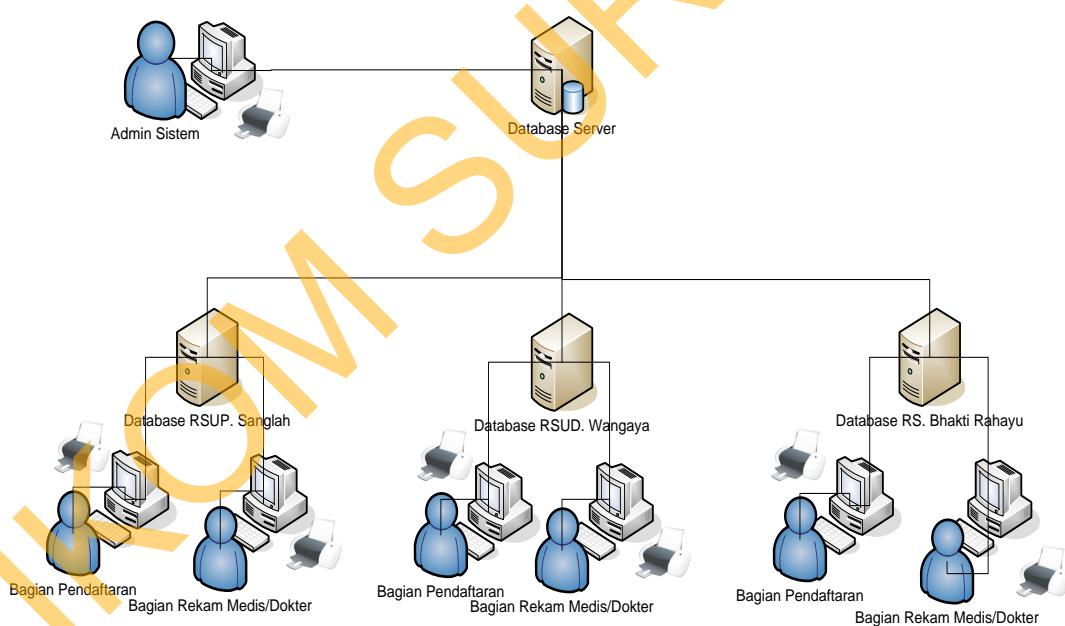
Tabel 3.22 Tabel Kunjungan Pasien di Tiga Rumah Sakit

Nama Rumah Sakit	Rata-Rata Kunjungan Pasien Per Hari	Rata-Rata Kunjungan Pasien Per Tahun	Rata-Rata Kunjungan Pasien 5 Tahun
RSUP. Sanglah	350 Orang	127.750 Orang	638.750 Orang
RSUD. Wangaya	80 Orang	29.200 Orang	146.000 Orang
RS. Bhakti Rahayu	15 Orang	5745 Orang	27.375 Orang
TOTAL	445 Orang	162.425 Orang	812.125 Orang

Menurut perhitungan data rekam kesehatan pasien berukuran ± 2 Megabyte, maka untuk 812.125 Orang selama dibutuhkan $\pm 1.624.250$ Megabyte atau setara dengan ± 1.62425 Terabyte. Untuk itu kapasitas Harddisk untuk server minimal 2 Terabyte agar dapat menampung data rekam kesehatan pasien selama 5 tahun.

3.8.1 Gambaran Umum Sistem

Untuk menggambarkan gambaran umum sistem, user dan tata letak kebutuhan perangkat keras yang akan digunakan pada sistem informasi rekam kesehatan ini digambarkan sebagai berikut :



Gambar 3.134 Gambaran Umum Sistem, User dan Perangkat Keras

3.9 Uji Coba Desain Sistem

Untuk menguji bahwa desain sistem yang telah dibuat dapat digunakan dengan baik sesuai dengan tujuan desain secara umum maka dilakukan pengujian

terhadap desain sistem informasi rekam kesehatan elektronik terpusat. Pengujian desain sistem dilakukan sebagai berikut :

3.9.1 Model Pengujian *Object Oriented Analysis and Desain (OOAD)*

Model analisis dan desain tidak dapat diuji dalam artian yang konvensional karena model ini tidak dapat dieksekusi, maka kajian teknis formal dapat digunakan untuk menguji kebenaran dan konsistensi model analisis dan model desa. Salah satu metode yang ada dan digunakan dalam uji coba ini adalah *scenario-base* testing yaitu identifikasi scenario dari use cases dan menambahkannya dengan diagram interaksi yang menunjukkan objek-objek yang terlibat dalam scenario. Uji coba scenario based testing ini, akan digunakan use case untuk menyimpan pemeriksaan pasien yang mana beberapa hasil pemeriksaan pasien menjadi data rekam kesehatan.

Tabel 3.23 Tabel Use Case, Skenario, dan Objek dari Use Case Menyimpan Hasil Pemeriksaan Pasien

Use Case	Skenario	Objek
Menyimpan Hasil Pemeriksaan Pasien	<ol style="list-style-type: none"> 1. User (Dokter atau Bagian Rekam Medis) memasukkan kode pasien pada menu cari rem medis pasien 2. User mengklik tombol cari 3. Sistem akan menampilkan data rekam kesehatan yang berisi riwayat data pemeriksaan pasien 4. User memilih tambah pemeriksaan baru 5. User input hasil pemeriksaan pasien 6. User input resep obat 7. User klik simpan untuk menyimpan hasil pemeriksaan 8. Sistem akan menyimpan pemeriksaan pasien ke dalam 	<ol style="list-style-type: none"> 1. Form Utama Untuk Dokter atau Bagian Rekam Medis 2. Form Cari Rekam kesehatan 3. Form Pemeriksaan Pasien 4. Form Resep Obat 5. Tabel Pemeriksaan 6. Tabel Rekam kesehatan 7. Form Pesan

	tabel pemeriksaan pasien dan tabel rekam kesehatan 9. Sistem menampilkan pesan data telah disimpan	
--	---	--

Tabel 3.23 menunjukkan scenario dari use case menyimpan hasil pemeriksaan pasien. Dari scenario yang ada akan dipilih kata benda, kata benda yang ada nantinya mungkin akan menjadi aktor, objek, class dan attribut dari class. Dalam tahapan pembuatan diagram inteaksi (*sequence diagram*) dibutuhkan objek-objek yang akan berinterasi.

3.9.2 Uji Coba Desain Sistem dengan Kuisioner

Untuk menguji bahwa desain sistem informasi berorientasi objek untuk sistem informasi rekam kesehatan elektronik terpusat ini dapat menampilkan pandangan dari sudut yang berbeda terhadap sistem yang dikembangkan dan berguna untuk memahami persoalan, menyiapkan dokumentasi, merancang program dan database maupun mengkomunikasikan dengan orang yang terlibat dalam proyek, terutama pengguna sistem dan juga programmer, maka penulis melakukan uji coba terhadap beberapa pengguna tugas akhir dan juga beberapa programmer. Hasil uji coba tersebut didapat dengan memberikan kuisioner terhadap 18 orang dari 3 bagian salah satu rumah sakit, yaitu 3 orang bagian pendaftaran pasien, 5 orang bagian rekam medis dan 10 orang dokter.

Ujicoba terakhir dilakukan dengan meminta masing-masing bagian memeriksa desain sistem yang telah dirancang untuk masing-masing bagian dan mengisi angket yang telah disediakan dengan jumlah pertanyaan sebanyak 5

pertanyaan. Angket dapat dilihat pada lampiran. Berikut ini adalah hasil dari angket tersebut :

Tabel 3.24 Rekap Hasil Angket Bagian Pendaftaran

Pertanyaan	5 (Sangat Baik)	4 (Baik)	3 (Cukup Baik)	2 (Kurang)	1 (Sangat Kurang)	Total	Skor
1	3					15	100%
2		3				12	80%
3		3				12	80%
4	3					15	100%
5		3				12	80%

Tabel 3.25 Rekap Hasil Angket Bagian Rekam Medis

Pertanyaan	5 (Sangat Baik)	4 (Baik)	3 (Cukup Baik)	2 (Kurang)	1 (Sangat Kurang)	Total	Skor
1	5					25	100%
2		5				20	80%
3		3	2			18	72%
4		5				20	80%
5		5				20	80%

Tabel 3.26 Rekap Hasil Angket Dokter

Pertanyaan	5 (Sangat Baik)	4 (Baik)	3 (Cukup Baik)	2 (Kurang)	1 (Sangat Kurang)	Total	Skor
1		10				40	80%
2		10				40	80%
3		10				40	80%
4	6	4				46	92%
5	4	6				44	88%

Menurut Hague (1995), rumus untuk mengitung nilai kolom ‘Total’ dan ‘Skor’ adalah sebagai berikut :

Jumlah orang disini berarti beberapa responden yang memilih nilai tertentu dari pertanyaan yang diajukan. Sedangkan bobot nilai adalah rentang nilai antara 5 – 1, dimana 5 adalah sangat baik dan 1 adalah sangat kurang. Hasil dari

jumlah orang dikali dengan bobot nilai akan menghasilkan nilai total, dimana total tersebut akan digunakan untuk menghitung skor akhir.

$$\text{Skor} = \frac{\sum (\text{jumlah orang} * \text{bobot nilai})}{(\text{bobot nilai tertinggi} * \text{total responden})} \dots \dots \dots (3.2)$$

Skor digunakan untuk mengetahui sejauh mana sebuah pertanyaan yang diajukan menghasilkan nilai balik antara sangat lemah sampai sangat kuat. Contoh, untuk pertanyaan nomor 1 untuk bagian rekam medis yaitu “Apakah rancangan sistem sudah dapat menyimpan semua data rekam kesehatan?”, jika skor akhir dari pertanyaan tersebut adalah 100% termasuk kedalam interpretasi sangat baik (sesuai dengan tabel kriteria interpretasi skor pada Tabel 3.24), yang berarti bahwa rata-rata responden beranggapan bahwa “Apakah rancangan sistem sudah dapat menyimpan semua data rekam kesehatan? terbilang sesuai dan dapat menyimpan rekam kesehatan secara terpusat, berarti dinilai dengan skor akhir yang dihasilkan dari hasil rekap angket.

Tabel 3.27 Kriteria Interpretasi Skor

Jangkaun Skor	Interpretasi
Antara 0% - 20%	Tidak Baik
Antara 21% - 40%	Kurang Baik
Antara 41% - 60%	Sedang
Antara 61% - 80%	Baik
Antara 81% - 100%	Sangat Baik

Tabel 3.24 menjelaskan interpretasi dari tiap nilai pada kolom ‘Skor’ pada Tabel 3.24. Berikut adalah kesimpulan dari kuesioner ini:

Tabel 3.28 Kesimpulan Angket Bagian Pendaftaran Pasien

No	Pertanyaan	Skor	Interpretasi
1.	Apakah rancangan sistem sudah dapat menyimpan semua data pasien?	100%	Sangat Baik
2.	Apakah rancangan sistem dapat menyimpan pendaftaran dan registrasi pasien pada saat pasien datang?	80%	Baik
3.	Apakah rancangan sistem sudah dapat mencari dan menampilkan data pasien yang akan melakukan registrasi?	80%	Baik
4.	Apakah rancangan sistem sudah dapat menampilkan dan mencetak laporan data pasien?	100%	Sangat Baik
5.	Apakah sistem sudah dapat menampilkan dan mencetak laporan kunjungan pasien?	80%	Baik

Tabel 3.29 Kesimpulan Angket Bagian Rekam Medis

No	Pertanyaan	Skor	Interpretasi
1.	Apakah rancangan sistem sudah dapat menyimpan semua data rekam kesehatan?	100%	Sangat Baik
2.	Apakah rancangan sistem telah mampu menjalankan fungsinya menambahkan hasil pemeriksaan ke dalam rekam kesehatan?	80%	Baik
3.	Apakah rancangan sistem sudah dapat mencari dan menampilkan data rekam kesehatan yang akan melakukan kunjungan?	72%	Baik
4.	Apakah rancangan sistem sudah dapat menambahkan resep obat yang diberikan pada saat pemeriksaan pasien?	80%	Baik
5.	Apakah rancangan sistem sudah dapat menampilkan rekam kesehatan?	80%	Baik

Tabel 3.30 Kesimpulan Angket Dokter

No	Pertanyaan	Skor	Interpretasi
1.	Apakah rancangan sistem sudah dapat mencari dan menampilkan data rekam kesehatan yang akan melakukan kunjungan?	80%	Baik
2.	Apakah rancangan sistem sudah menjalankan fungsinya dalam mempilkhan data rekam medis	80%	Baik
3.	Apakah rancangan sistem sudah dapat menambahkan resep obat yang diberikan pada saat pemeriksaan pasien?	80%	Baik
4.	Apakah rancangan sistem telah dapat menambahkan nama dokter pemeriksa pada lembar pemeriksaan pasien?	92%	Sangat Baik
5.	Apakah rancangan sistem sudah dapat menampilkan rekam kesehatan?	88%	Sangat Baik

Dari rata – rata hasil angket tersebut dapat disimpulkan bahwa desain sistem informasi rekam kesehatan elektronik terpusat layak digunakan dengan hasil baik.